

Bc. Michal Tvarožek

**PERSONALIZOVANÁ NAVIGÁCIA
V INFORMAČNOM PRIESTORE
REPREZENTOVANOM ONTOLÓGIU**

Diplomová práca

ZADANIE DIPLOMOVEJ PRÁCE

Meno študenta: **Bc. Michal Tvarožek**
Študijný odbor: SOFTVÉROVÉ INŽINIERSTVO
Študijný program: Softvérové inžinierstvo
Názov projektu: **Personalizovaná navigácia v informačnom priestore
reprezentovanom ontológiou**

Zadanie:

Efektívna prezentácia informácií, najmä navigácia v rozsiahlych informačných priestoroch predstavuje dnes jeden z aktuálnych problémov. Problémy vznikajú najmä v prípade rozsiahlych informačných priestorov, ktoré sa menia a obsahujú údaje reprezentované v rôznych formách. Tu môže pomôcť opis obsahu informačného priestoru s využitím jazykov vytvorených pre web so sémantikou. Analyzujte možnosti prezentovania informácií a navigácie s využitím opisu domény ontológiou. Navrhňte metódu personalizovanej navigácie v priestore opísanom ontológiou domény. Využite pri tom model používateľa na vhodné označenie odkazov na prezentovaný text a usporiadanie odkazov tak, aby to zodpovedalo požiadavkám, charakteristike a preferenciám používateľa. Ďalej skúmajte možnosti vhodnej prezentácie výsledkov v prostredí webu. Navrhnuté riešenie overte vytvorením softvérového prototypu systému na personalizovanú navigáciu. Riešenie experimentálne overte pre informačný priestor pracovných príležitostí.

Odporúčaná literatúra:

L. Zhang, Y. Yu, J. Zhou, CX. Lin, Y. Yang: An Enhanced Model for Searching in Semantic Portals. In Proc. of WWW 2005, ACM Press, 2005, pp. 453-462.

P. Castells, M. Fernández, D. Vallet, P. Mylonas, Y. Avrithis. Self-Tuning Personalized Information Retrieval in an Ontology-Based Framework. In Proc. of 1st IFIP WG 2.12 & WG 12.4 Int. Workshop on Web Semantics - SWWS 2005, Meersman, R.; Tari, Z.; Herrero, P. (Eds.), Springer LNCS 3762, 2005, pp. 977-986.

Práca musí obsahovať:

- Anotáciu v slovenskom a anglickom jazyku
- Analýzu problému
- Opis riešenia
- Zhodnotenie
- Technickú dokumentáciu
- Zoznam použitej literatúry
- Elektronické médium obsahujúce vytvorený produkt spolu s dokumentáciou

Miesto vypracovania: Ústav informatiky a softvérového inžinierstva, FIIT STU, Bratislava
Vedúci projektu: prof. Ing. Mária Bieliková PhD.

Termín odovzdania práce v zimnom semestri: dňa 12. decembra 2006

Bratislava, dňa 25. septembra 2006



prof. Ing. Pavol Návrat, PhD.
riaditeľ ÚISI

ANOTÁCIA

Slovenská technická univerzita v Bratislave

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

Študijný odbor: SOFTVÉROVÉ INŽINIERSTVO

Autor: Bc. Michal Tvarožek

Diplomová práca: Personalizovaná navigácia v informačnom priestore reprezentovanom ontológiou

Vedúci diplomového projektu: prof. Ing. Mária Bieliková, PhD.

december, 2006

Efektívna práca s veľkým otvoreným informačným priestorom prináša so sebou viacero problémov v podobe veľkej časovej náročnosti vyhľadania výsledkov a náročnosti formulácie dopytov so sémantikou. K problémom navigácie patria vysoká miera rekurzcie, stratenie sa používateľov a neprívetivosť webových sídel. Práca obsahuje analýzu aktuálnych prístupov k riešeniu uvedených problémov, ku ktorým patria rozdielne spôsoby navigácie v informačnom priestore, adaptívne (webové) systémy a fazetové prehliadače. V práci opisujeme návrh metódy personalizovanej navigácie v informačnom priestore opísanom ontológiou s využitím fazetového prehliadača a adaptívnej navigácie. Navrhnutá metóda využíva sémantiku údajov, uložených v doménovej ontológii a ontológii používateľa, na prispôbovanie navigácie vo fazetovom prehliadači. Hlavné prínosy metódy sú rozšírenie fazetového prehliadača o pokročilý režim tvorby dopytov, dynamické generovanie faziet z doménovej ontológie. Ďalším prínosom je adaptácia a anotácia faziet a výsledkov vyhľadávania s využitím porovnania podobnosti konceptov v ontológii domény a používateľa. Overenie vybraných častí navrhovanej metódy sme realizovali ich implementáciou v podobe softvérových nástrojov *Factic* a *SemanticLog* a vykonaním experimentov v doméne pracovných príležitostí a v doméne vedeckých publikácií v kontexte väčších projektov riešených na FIIT STU.

ANNOTATION

Slovak University of Technology Bratislava

FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Degree Course: SOFTWARE ENGINEERING

Author: Bc. Michal Tvarožek

Thesis: Personalized navigation in an information space represented by an ontology

Supervisor: prof. Ing. Mária Bieliková, PhD.

2006, December

The need for effective access to large open information spaces introduced several new challenges such as the long time required to find relevant information and the difficulty of creation of search queries with semantics. Additional navigation problems – high navigation recursion rate, the “lost in hyperspace” syndrome and the user-unfriendliness of web sites further hamper user experience. The author analyzes current approaches that are proposed as possible solutions for the aforementioned problems – various navigation methods, adaptive web systems and faceted browsers. Based on these approaches, the author proposes a novel method for adaptive navigation in information spaces represented by domain and user ontologies by means of a faceted browser. The proposed method uses the semantics of the stored data to adapt navigation in a faceted browser enhanced with advanced query construction, dynamic facet generation and adaptation and annotation of facets and search results based on ontological concept comparison. The evaluation was performed by implementing selected aspects of the proposed method as two software tools – *Factic* and *SemanticLog* and by their successive validation in the domain of online job offers and research articles in the context of two larger projects conducted at FIIT STU.

*Čestne prehlasujem, že som diplomovú prácu vypracoval samostatne,
s použitím uvedenej literatúry.*

Bratislava, december 2006

*Chcem sa poďakovať mojej diplomovej vedúcej profesorky Márii Bielíkovej za nápady,
podnety, ochotu, vynaložené úsilie a ústretovosť počas riešenia diplomovej práce.
Tiež sa chcem poďakovať mojim rodičom a bratovi za podporu a trpezlivosť.*

Ďakujem.

Obsah

1	ÚVOD.....	1
2	NAVIGÁCIA V INFORMAČNOM PRIESTORE WEBU SO SÉMANTIKOU	5
2.1	VIZUALIZÁCIA NAVIGÁCIE	7
2.2	LINEÁRNA NAVIGÁCIA	8
2.3	HIERARCHICKÁ NAVIGÁCIA	9
2.4	FAZETOVÁ NAVIGÁCIA.....	9
2.4.1	<i>Fazetová klasifikácia</i>	<i>10</i>
2.4.2	<i>Charakteristiky fazetových prehliadačov.....</i>	<i>11</i>
2.4.3	<i>Navigácia vo fazetovom prehliadači.....</i>	<i>12</i>
3	ADAPTÍVNA NAVIGÁCIA	15
3.1	TECHNIKY ADAPTÍVNEJ NAVIGÁCIE	15
3.2	REALIZÁCIA ADAPTÍVNEJ NAVIGÁCIE.....	17
3.3	NAVIGÁCIA V EXISTUJÚCICH ADAPTÍVNYCH SYSTÉMOCH	18
4	ZHODNOTENIE SÚČASNÉHO STAVU A CIELE.....	21
5	METÓDA PERSONALIZOVANEJ NAVIGÁCIE SO SÉMANTIKOU	23
5.1	PREHEAD METÓDY	23
5.1.1	<i>Reprezentácia údajov ontológiou</i>	<i>25</i>
5.1.2	<i>Fazetový prehliadač ako prostriedok navigácie</i>	<i>25</i>
5.1.3	<i>Prispôbovanie vo fazetovom prehliadači</i>	<i>26</i>
5.1.4	<i>Prispôbovanie vo výslednom dokumente.....</i>	<i>27</i>
5.2	REPREZENTÁCIA DOMÉNY A MODELU POUŽÍVATEĽA	27
5.2.1	<i>Doménová ontológia.....</i>	<i>27</i>
5.2.2	<i>Ontológia používateľa</i>	<i>28</i>
5.2.3	<i>Ontológia udalostí.....</i>	<i>30</i>
5.3	ROZŠÍRENÝ FAZETOVÝ SÉMANTICKÝ PREHLIADAČ	30
5.3.1	<i>Rozšírenia fazetového prehliadača.....</i>	<i>31</i>
5.3.2	<i>Zaznamenávanie akcií používateľa so sémantikou</i>	<i>33</i>
5.3.3	<i>Prispôbovania navigácie vo fazetovom prehliadači.....</i>	<i>33</i>
5.3.4	<i>Prispôbovanie práce s výsledným dokumentom.....</i>	<i>37</i>
5.4	NÁVRH GRAFICKÉHO POUŽÍVATEĽSKÉHO ROZHRAŇA	38
6	SOFTVÉROVÉ NÁSTROJE REALIZUJÚCE METÓDU	41
6.1	KONTEXT A ARCHITEKTÚRA RIEŠENIA	41
6.2	FACTIC – NÁVRH A IMPLEMENTÁCIA	43
6.3	SEMANTICLOG – NÁVRH A IMPLEMENTÁCIA	46
7	EXPERIMENTÁLNE OVERENIE METÓDY	49
7.1	ZÁZNAM AKCIÍ POUŽÍVATEĽA SO SÉMANTIKOU	49
7.2	EFEKTÍVNOSŤ PRÁCE S VEĽKÝM INFORMAČNÝM PRIESTOROM.....	49
7.3	PRISPÔBOVANIE SA CHARAKTERISTIKÁM POUŽÍVATEĽA	52
8	ZHODNOTENIE	55
PRÍLOHY		
PRÍLOHA A	PRINCÍPY WEBU SO SÉMANTIKOU	
PRÍLOHA B	ONTOLOGICKÉ MODELÝ PROJEKTOV NAZOU A MAPEKUS	
PRÍLOHA C	PUBLIKÁCIE VZŤAHUJÚCE SA K VÝSLEDKOM DIPLOMOVEJ PRÁCE	
PRÍLOHA D	TECHNICKÁ DOKUMENTÁCIA	
PRÍLOHA E	OBSAH ELEKTRONICKÉHO MÉDIA	
POUŽITÁ LITERATÚRA		

1 Úvod

Problematika vizualizácie informačných priestorov a navigácie v nich je v súčasnosti mimoriadne aktuálna a venujú sa jej tímy výskumníkov na mnohých pracoviskách. Jeden z hlavných smerov výskumu, ktorý je v súčasnosti už pomerne dobre rozpracovaný, predstavujú metódy pre prácu s uzavretými informačnými priestormi, napr. kurzami pre e-vzdelávanie. Na druhú stranu, metódam určeným pre otvorené informačné priestory, napr. web bola doteraz venovaná neporovnateľne menšia pozornosť.

Práve problematika otvorených informačných priestorov a najmä webu v súčasnosti predstavuje veľmi dynamickú oblasť. Množstvo dostupných informácií, webových stránok i celých webových sídel narastá exponenciálne rýchlo, pričom úmerne k tomu narastá aj náročnosť práce s dostupnými informáciami. Podľa jednej štúdie web obsahoval v roku 2001 približne 550 miliárd dokumentov, väčšinou vo svojej „neviditeľnej“ časti (Wikipedia, 2006b). Podľa inej štúdie bolo k januáru 2005 na webe dostupných približne 11,5 miliardy verejne indexovateľných stránok (Gulli, Signorini, 2005). Dôsledkom narastajúceho množstva dostupných informácií je skutočnosť, že súčasné (nedokonalé) technické a softvérové prostriedky neumožňujú dostatočne efektívne vyhľadávať a pracovať s dostupnými informáciami. Z uvedeného vyplýva viacero závažných problémov, ktoré súvisia nielen s webom ako takým, ale prejavujú sa viac-menej aj na jednotlivých webových sídlach.

Pravdepodobne najzávažnejším problémom je skutočnosť, že *vyhľadanie relevantných informácií trvá čoraz dlhšie* a používatelia následne trávajú hľadaním relevantných informácií veľmi dlhý čas. Súčasne sa stávajú od týchto informácií „závislí“ a stále viac sa spoliehajú na konkrétnu informačnú bázu na úkor práce s inými zdrojmi. Samotný proces vyhľadania výsledkov pomocou niektorého z vyhľadávacích strojov trvá len zlomky sekúnd, avšak používatelia musia stráviť nad výsledkami dlhé minúty, resp. ich strávia hľadaním informácií na niektorej z nájdených stránok. Tento fakt potvrdzujú aj výsledky štúdií vykonaných v rokoch 1996 a 1999, ktoré určili mieru rekurencie (návratu k predchádzajúcim stránkam, napr. pomocou tlačidla „späť“) pri navigácii na úrovni asi 60%, z celkového počtu navštívených stránok (Levene, Wheeldon, 2003).

S vyhľadávaním informácií súvisí skutočnosť, že súčasný web bol vytvorený tak, aby bol použiteľný, resp. čitateľný človekom. Tomuto účelu bol prispôsobený aj jazyk HTML, ktorý určuje spôsob vizualizácie webových stránok, ale neopisuje ich samotný informačný obsah. *Strojové spracovanie obsahu webových stránok je v súčasnosti značne obmedzené*, ak vôbec možné. Existujú prístupy na obaľovanie stránok a získavanie informácií v nich uložených, ale tieto majú len obmedzené použitie a vyžadujú si značnú údržbu kvôli častým zmenám štruktúry webových stránok. Práve obtiažnosť strojového spracovania webových stránok značne obmedzuje súčasné možnosti vyhľadávania na webe.

Štruktúra jednotlivých webových sídel, ako aj samotný obsah stránok vedie v súčasnosti často k „*strateniu sa*“ *používateľa*, pretože tento stratí pojem o štruktúre hyperpriestoru, v ktorom sa práve pohybuje. K tomuto javu dochádza v už existujúcich uzavretých systémoch, ktoré sú svojou povahou jednoduchšie a „menšie“ ako web samotný. Práve kvôli veľkosti

a otvorenosti webu je problém „stratenia sa“ používateľa ďaleko závažnejší ako v uzavretých systémoch a jeho riešenie si vyžaduje pozornosť.

Súvisiacou, aj keď mierne odlišnou, je problematika „neprívetivosti“ *súčasného webu*. Samotný web obsahuje veľké množstvo stránok, ktoré ponúkajú rozličné informácie (napr. spravodajstvo, výučbové kurzy), resp. funkcionality (napr. internetové bankovníctvo), pričom každá stránka má spravidla svoj vlastný spôsob použitia a vizualizácie. Jednotliví používatelia webu sa však od seba odlišujú, majú iné požiadavky na web a webové aplikácie, iné očakávania a iné zaužívané pracovné postupy. Aj keď sa v súčasnosti stránky v lepšom prípade vytvárajú tak, aby vyhovovali čo možno najväčšiemu počtu používateľov, zďaleka nemôžu pokryť potreby všetkých používateľov. Následne predstavujú len istý kompromis medzi súčasnými možnosťami tvorby webových stránok a webových systémov a použiteľnosťou pre isté skupiny používateľov. Čoraz častejšie sa objavujú snahy o prispôbovanie sa používateľom, resp. skupinám používateľov, ale tieto nie sú zatiaľ príliš rozšírené a ani príliš úspešné v otvorenom priestore webu.

Uvedené problémy a ich závažnosť sú známe dlhšiu dobu a na ich riešení sa v súčasnosti podieľa pomerne veľká komunita výskumníkov. Boli navrhnuté viaceré prístupy k ich riešeniu a práca sa v súčasnosti sústreďuje na ich dopracovanie, resp. zlepšovanie navrhnutých metód a ich „pretavenie“ do praxe. Toto však nevylučuje objavenie nových prístupov a metód riešenia uvedených problémov.

Jedno z pracovísk, ktoré pracuje na riešení spomenutých problémov je aj Ústav informatiky a softvérového inžinierstva na Fakulte informatiky a informačných technológií STU v rámci projektov NAZOU¹ a MAPEKUS². Projekt NAZOU – *Nástroje pre získavanie, organizovanie a udržiavanie znalostí v prostredí heterogénnych informačných zdrojov* (Návrat et al., 2006) sa zaoberá celým procesom práce s informáciami s overením v doméne pracovných príležitostí. Rieši získanie zdrojových informácií z prostredia webu, ich strojové spracovanie a konverziu do ontologického modelu domény, získavanie znalostí o používateľoch a ich uloženie do ontologického modelu používateľa ako aj personalizovanú prezentáciu informácií na základe modelu používateľa.

Projekt MAPEKUS – *Modelovanie a získavanie, spracovanie a využívanie znalostí o konaní používateľa v hyperpriestore Internetu* (Bieliková a kol., 2005a) sa sústreďuje na pokročilé metódy modelovania charakteristík používateľa a následné prispôbovanie prezentácie a odporúčanie obsahu s overením v doméne vedeckých publikácií.

Diplomovú prácu som vypracoval v kontexte spomenutých projektov, na ktorých riešení som sa podieľal najmä v podobe práce na metódach a nástrojoch pre personalizáciu navigácie a získavanie znalostí o používateľovi, ďalej práce na tvorbe ontologických modeloch oboch domén a modelu používateľa.

Hlavné ciele práce sú:

- *Identifikovať problémy navigácie* v rozsiahlych otvorených informačných priestoroch.
- *Identifikovať možné riešenia* nájdených problémov navigácie.

¹ Projekt NAZOU: nazou.fiit.stuba.sk (10.12.2006)

² Projekt MAPEKUS: mapekus.fiit.stuba.sk (10.12.2006)

- *Navrhnuť metódy na riešenie identifikovaných problémov navigácie.*
- *Overiť navrhnuté metódy implementáciou prototypu softvérového nástroja a vykonaním experimentov.*

V práci navrhujeme riešenie problémov navigácie pomocou kombinácie fazetového prehliadača, technológií webu so sémantikou (ontológií) a personalizácie na základe modelu používateľa. Uvedeným prístupom je venovaná nemalá pozornosť v rámci oboch projektov.

Samotné ontológie predstavujú pomerne nový prístup k reprezentácii a zápisu informácií, ktorý má niekoľko výhod oproti „klasickému“ ukladaniu údajov do relačných databáz. Výhodou použitia ontológií je jednotný prístup tak k uloženým dátam, ako aj k samotnej štruktúre dát. Ontológie súčasne umožňujú definovať význam konceptov a vzťahov medzi nimi. Ak sa naplní vízia webu so sémantikou a ontológie budú zdieľané viacerými (mnohými) subjektmi, umožnia definovať jednotnú sémantiku konceptov, ktorej budú rozumieť všetky zúčastnené strany ako aj stroje.

Nezanedbateľnou výhodou ontológií je tiež ich znovupoužitie, ktoré je jednoduchšie a priamočiarejšie ako pri dátových modeloch relačných databáz. Tvorba dopytov pomocou ontologických dopytovacích jazykov je prirodzenejšia a úplne nezávislá od skutočného spôsobu ich uloženia v databáze. V neposlednom rade využitie ontológií a vhodných databázových strojov umožňuje použiť odvodzovanie (angl. reasoning), pomocou ktorého je možné automatizovane odvodiť nové znalosti z už existujúcich znalostí. Táto vlastnosť sa napr. prejavuje v potrebe značne kratších dopytov do databáz.

Použitie ontológií však nie je bezproblémové. Asi najväčšou prekážkou ich použitia sú jednak nedokonalosť a nezrelosť dostupných nástrojov (editorov, databázových strojov) ako aj nevyhnutnosť tvorby vlastných doménových ontológií a ich napĺňanie informáciami, keďže web v súčasnosti priamo neobsahuje potrebné informácie v ontologickej podobe.

Druhým dôležitým aspektom je personalizácia založená na automatizovanom získavaní charakteristík používateľa a následné prispôbovanie navigácie a obsahu informačného priestoru odhadovaným potrebám a požiadavkám používateľa. Problematika prispôbovania sa nástrojov potrebám používateľa skrýva v sebe aj ďalšie aspekty ako sú ochrana súkromia, filtrovanie a (ne)dostupnosť informácií, resp. obavy z cenzúry informácií.

K prispôbovaniu je nevyhnutné zbierať informácie o používateľovi, pričom samotné prispôbovanie má predpoklady byť úspešnejším a použiteľnejším v prípade, že o používateľovi je známych mnoho informácií s veľkou presnosťou. Ich získavanie prebieha na pozadí a v ideálnom prípade bez zásahu či dokonca bez vedomia používateľa (po jeho predchádzajúcom súhlase). Zneužitie, resp. zverejnenie získaných informácií môže nielen narušiť súkromie používateľa ale aj spôsobiť závažné škody (napr. stratu zamestnania).

Na zabezpečenie súkromia používateľov je nielen potrebné primeraným spôsobom chrániť adaptívne systémy a ich databázy s údajmi o používateľoch ale aj umožniť samotným používateľom riadiť množstvo a povahu údajov, ktoré bude systém o používateľovi zbierať.

Potenciálne ďaleko nebezpečnejšie však môže byť prispôbovanie obsahu, resp. filtrovanie informácií na základe modelu používateľa, ktoré adaptívny systém vykonáva za účelom zjednodušenia práce používateľa. Jeho cieľom je poskytnúť používateľovi práve tie

informácie, ktoré potrebuje a súčasne pred ním skryť tie, ktoré nepotrebuje alebo sú prebytočné. Systém sa však môže „pomýliť“ a znepriístupniť informácie, ktoré sú pre používateľa v skutočnosti relevantné ale podľa systému zbytočné. V prípade, že by niekto úmyselne do systému zaviedol „chybu“ napr. tak, že by o každom používateľovi systém predpokladal nezáujem o šport, tak by boli informácie o športe efektívne „cenzurované“.

V praxi bude potrebné nájsť vhodnú mieru filtrovania a skrývania informácií a tiež umožniť používateľovi sprístupnenie všetkých informácií, resp. vypnúť prispôsobovanie. Aj napriek tomu však zostáva otvorenou otázka, či takéto zasahovanie do prezentovaných informácií je prípustné, vhodné či želateľné. Ako autor metódy, ktorá využíva uvedené postupy k zefektívneniu práce používateľa so systémom som presvedčený, že potenciálne prínosy prevyšujú hroziace riziká. Aj napriek tomu si však myslím, že skutočné praktické nasadenie prispôsobovania do praxe si bude vyžadovať širšiu diskusiu a najmä veľmi dôkladnú realizáciu z pohľadu zabezpečenia a konfigurovateľnosti používateľom. Používateľ si musí byť vedomý potenciálnych rizík a úskalí a musí mať konečné slovo v tom, či daný systém bude používať a v akom rozsahu to bude robiť.

Práca je rozdelená do ôsmich častí. Kapitola 2 opisuje základné charakteristiky informačných priestorov z hľadiska navigácie a najčastejšie používané metódy navigácie v nich. Ďalej opisuje fazetový prehliadač ako jeden z možných prístupov pre navigáciu vo veľkých informačných priestoroch pomocou fazetovej klasifikácie, ktorá značne zjednodušuje navigáciu vo veľkých otvorených informačných priestoroch.

V kapitole 3 bližšie analyzujeme metódy adaptívnej navigácie a jej základné prístupy. Venujeme sa spôsobom realizácie adaptívnych systémov a tiež spôsobom modelovania používateľa pre potreby personalizácie. V krátkosti opisujeme navigáciu vo vybraných existujúcich adaptívnych systémoch. Kapitola 4 obsahuje stručné zhrnutie súčasného stavu v oblasti navigácie a definovanie cieľov práce. Navrhnutú metódu adaptívnej navigácie s využitím sémantiky opíšeme v kapitole 5. Na začiatku uvedieme prehľad navrhutej metódy, ďalej opíšeme spôsob reprezentácie doménových údajov pomocou doménovej ontológie a údajov o používateľovi pomocou ontológie používateľa. Následne opíšeme podrobný návrh metódy – adaptívneho fazetového prehliadača.

V kapitole 6 sa venujeme overeniu návrhu v podobe implementácie softvérových nástrojov pre prispôsobovanie navigácie pomocou fazetového sémantického prehliadača *Factic* a nástroja na zaznamenávanie údajov o používateľovi so sémantikou *SemanticLog*. Následne v kapitole 7 opíšeme spôsob overenia navrhutej metódy v dvoch doménach – v doméne pracovných príležitostí a v doméne vedeckých publikácií. Vysvetlíme postup vykonania experimentov spolu s ich výsledkami.

V kapitole 8 zhodnotíme výsledky práce, prínos navrhutej metódy a možné smery ďalšej práce. Práca obsahuje prílohy – opis základných princípov webu so sémantikou, opis použitých ontológií, technickú dokumentáciu k riešeniu a obsah priloženého elektronického média. Ako prílohu uvádzame aj výsledky práce vo forme vedeckých príspevkov prijatých na tri medzinárodné konferencie Adaptive Hypermedia 2006, Dublin (Doctoral Consortium), DATAKON 2006, Brno a Sofsem 2007, Harachov.

2 Navigácia v informačnom priestore webu so sémantikou

Vo všeobecnosti navigácia predstavuje pohyb v informačnom priestore spojený s orientovaním sa v ňom. Ak sa obmedzíme na prostredie webu, tak navigáciu možno definovať ako *aktivitu nasledovania odkazov a prehliadania webových stránok*. Cieľom „dobrého“ navigačného systému je zabrániť „strateniu sa“ používateľov a informovať používateľov o ich aktuálnej polohe v informačnom priestore a možnostiach ďalšej navigácie. V súčasnosti je navigácia nevyhnutná na uspokojenie potrieb používateľa, pretože za bežných okolností nie je možné tieto uspokojiť na jednej stránke, resp. na prvej stránke, ktorú navštívi. Všeobecne možno potreby používateľa rozdeliť takto (Levene, Wheeldon, 2003):

- *Informačná potreba* vzniká, keď používateľ hľadá nejaké informácie, napr. o nejakej knihe alebo o najnovšom filme.
- *Transakčná potreba* vzniká, ak používateľ potrebuje vykonať nejakú operáciu, napr. cez internetové bankovníctvo vykonať prevod peňazí na účet.
- *Navigačná potreba* vzniká, ak používateľ hľadá vhodnú počiatočnú stránku, z ktorej začne postupne prezerat' ďalšie stránky a študovať informácie na nich.

Na uspokojenie uvedených potrieb používateľa možno použiť rôzne spôsoby navigácie v informačnom priestore s rozličnou mierou efektivity. Typy navigácie v rámci jedného webového sídla možno rozdeliť na (AGIMO, 2004):

- *Globálnu navigáciu*, ktorá umožňuje jednoduchý prístup k hlavným častiam sídla a je prítomná spravidla na každej stránke.
- *Lokálnu navigáciu*, ktorá umožňuje používateľom pohybovať sa po časti sídla, v ktorej sa práve nachádzajú.
- *Kontextovú navigáciu*, ktorá spája súvisiace stránky a najčastejšie býva riešená pomocou odkazov texte alebo zoznamov odkazov typu „pozri tiež“.
- *Podpornú navigáciu*, ktorá predstavuje doplnkové navigačné nástroje ako mapy sídel, indexy a návody, resp. sprievodcov.

Obrázok 1 znázorňuje vybrané prostriedky navigácie, pomocou ktorých možno uvedené typy navigácie prakticky realizovať (Timberlake, 2000):

- *Navigačné lišty* (angl. navigation bar), ktoré obsahujú zoznam odkazov, najčastejšie používané ako menu vľavo/hore/vpravo (Obrázok 1, A).
- *Odkazy* (angl. embedded links) predstavujúce základný spôsob navigácie pomocou vkladania odkazov priamo do prezentovaných informácií (Obrázok 1, D). Odkazy môžu byť textové alebo môžu byť realizované pomocou obrázkov.
- *Záložky* (angl. tab navigation), pomocou ktorých možno obsah efektívne rozdeliť na niekoľko hlavných kategórií (Obrázok 1, C).
- *Mapu sídla* (angl. site map), ktorá obsahuje zoznam všetkých stránok webového sídla.

- *Stopy* (angl. bread-crumb trail), ktoré zjednodušujú organizáciu veľkého množstva informácií a zobrazujú aktuálnu polohu v informačnom priestore spolu s cestou akou sa do nej používateľ dostal (Obrázok 1, B).



Obrázok 1: Často používané prostriedky navigácie predstavujú navigačná lišta – menu (A), stopy (B), záložky (C) a odkazy v texte (D).

Používateľ môže v praxi použiť rôzne spôsoby navigácie v informačnom priestore v závislosti od aktuálneho typu navigácie a dostupných navigačných prostriedkov. V istom zmysle možno navigáciu používateľa chápať aj ako spôsob vyhľadávania informácií na webovom sídle, kedy sa pomocou navigácie medzi stránkami používateľ pokúša nájsť cieľovú stránku a uspokojiť tak svoje potreby.

Rozlišujeme tieto spôsoby navigácie:

- *Lineárna navigácia* – postupné prezeranie zoznamu stránok, napr. výsledkov vyhľadávania internetového vyhľadávača.
- *Hierarchická navigácia* – navigácia v hierarchickej stromovej štruktúre, napr. navigácia na veľkých sídlach pomocou hlavného menu alebo v katalógoch internetových obchodov.
- *Fazetová navigácia* – navigácia pomocou fazetovej klasifikácie, napr. hľadanie vhodnej pracovnej ponuky alebo produktu v internetovom obchode.

2.1 Vizualizácia navigácie

V priestore webu sa používa na zobrazovanie stránok viacero spôsobov vizualizácie, pričom pre každý spôsob navigácie môže byť vhodný iný spôsob vizualizácie. Existujúce prístupy k navigácii z hľadiska vizualizácie sú:

- *Klasická („textová“) navigácia*, ktorá je používaná najčastejšie a predstavuje navigáciu v rozhraní tvorenom textom a samostatnými obrázkami pomocou textových alebo obrázkových odkazov (Obrázok 2).



Obrázok 2: Ukážka klasickej navigácie s využitím vizualizácie textu.
Odkazy sú realizované pomocou textu, resp. ako obrázky.

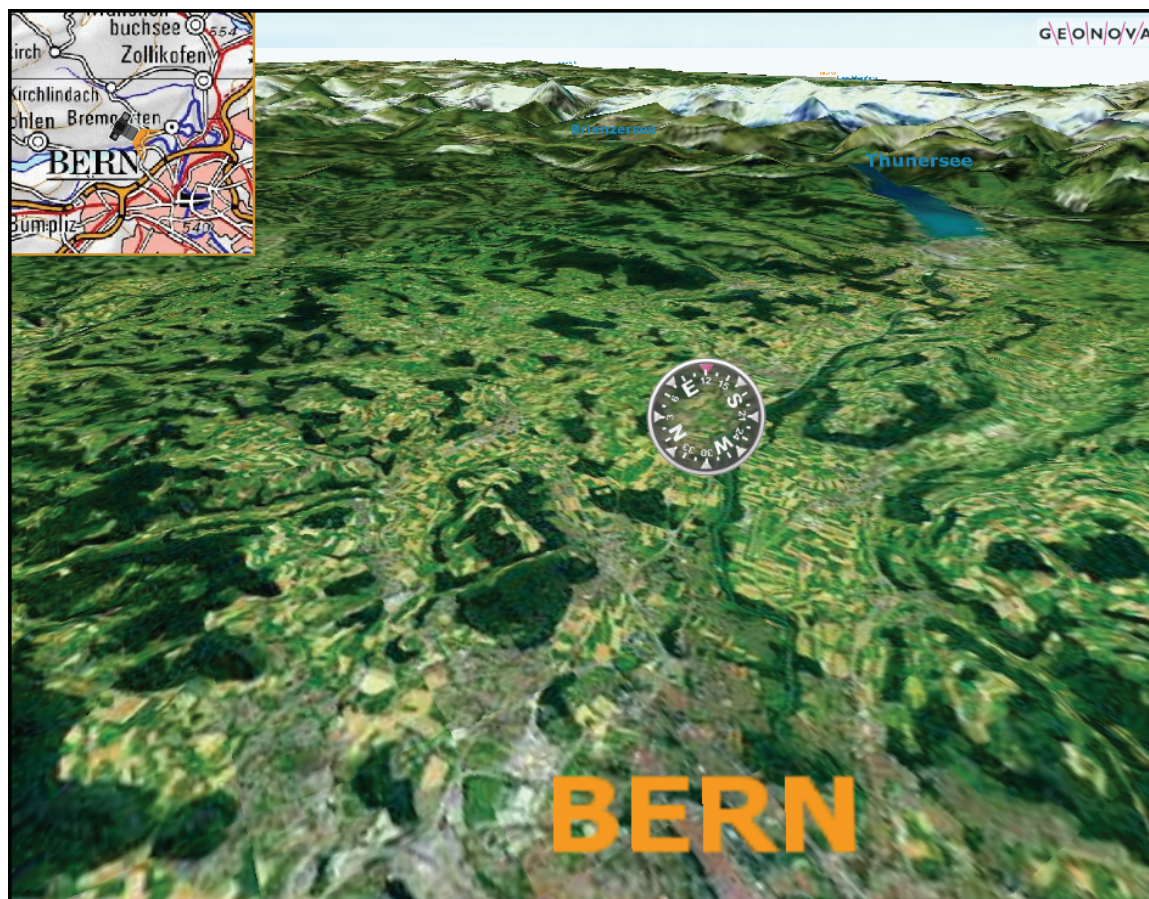
- *Vizuálna navigácia* využíva interaktívnu grafickú vizualizáciu odkazov, ktoré môžu napr. znázorňovať štruktúru informačného priestoru za účelom zlepšenia predstavy používateľa o informačnom priestore (Obrázok 3). Možno použiť napr. vizualizáciu grafov (napr. AllIKnow³) alebo hierarchických zhlukov dát. Vizuálna navigácia môže byť realizovaná pomocou Java appletu, priamo pomocou obrázkov, prípadne pomocou SVG.



Obrázok 3: Ukážka vizuálnej navigácie s využitím vizualizácie grafu v projekte AllIKnow.
Odkazy sú realizované ako vrcholy grafu.

³ AllIKnow, <http://alliknow.net/>, (16.11.2006)

- *Priestorová navigácia*, ktorá je najčastejšie realizovaná pomocou zásuvných modulov, pričom umožňuje používateľovi voľne sa pohybovať v dvojrozmernom, resp. trojrozmernom priestore a prípadne interagovať s objektmi v ňom. Najčastejšie použitie priestorovej vizualizácie predstavujú virtuálne mapy napr. Google Maps⁴ (2D), resp. Geonova⁵ (3D). Použitie priestorovej navigácie zatiaľ nie je na webe veľmi rozšírené.



Obrázok 4: Ukážka priestorovej navigácie v systéme GVista spoločnosti Geonova.

2.2 Lineárna navigácia

Lineárna navigácia predstavuje priamočiary spôsob prezerania webových stránok, ktorý sa používa najčastejšie pri lokálnej navigácii v neštruktúrovaných informačných priestoroch, resp. pri prezeraní zoznamov (napr. výsledkov vyhľadávania). Pri lineárnej navigácii používateľ postupne prezerá stránky jednu za druhou a zisťuje, či uspokojí jeho aktuálnu potrebu. Nevýhodou lineárnej navigácie je najmä skutočnosť, že je pomerne časovo náročná a používateľ má len málo informácií o informačnom obsahu ďalších stránok. Napr. pri lineárnom prezeraní výsledkov vyhľadávania si používateľ zvyčajne prezrie len niekoľko prvých výsledkov a ostatné vynechá. Výhodou je univerzálna a jednoduchá možnosť použitia

⁴ Google Maps, <http://maps.google.com/> (10.12.2006)

⁵ Geonova, <http://www.geonova.ch/gvista/pages/ch/GVista.htm> (16.11.2006)

Lineárna navigácia môže byť postačujúca ak predpokladáme napr. pri vyhľadávačoch, že prvé odkazy sú najviac relevantné.

2.3 Hierarchická navigácia

Hierarchická navigácia pre svoje použitie vyžaduje existenciu hierarchickej klasifikácie v informačnom priestore. Často sa používa pri globálnej navigácii po webových sídlach s definovanou štruktúrou, kde používateľ na základe klasifikácie „vyhľadáva“ potrebné informácie. Hlavné menu (navigačná lišta) webových sídel často zodpovedá hierarchickej navigačnej štruktúre sídla (klasifikácii informácií na stránkach).

Výhodou hierarchickej navigácie je, že sprostredkuje používateľovi informáciu o približnom obsahu nasledujúcich stránok a následne umožňuje používateľovi nájsť potrebné informácie rýchlejšie. Napr. výberom odkazu „Notebooky“ hierarchickej navigácii po internetovom obchode používateľ môže predpokladať, že na ďalších stránkach budú informácie o notebookoch. Nevýhodami sú potreba hierarchickej klasifikácie a ňou vopred definované smery navigácie, kedy navigácia je možná len po cestách v danej hierarchii.

2.4 Fazetová navigácia

Fazetovú navigáciu využíva fazetový prehliadač (angl. faceted browser), ktorý predstavuje vhodné riešenie na prehliadanie veľkých štruktúrovaných informačných priestorov. Umožňuje používateľovi jednoduchým spôsobom špecifikovať informácie, ktoré sa majú prezentovať pomocou definovania ohraničení nad zobrazovanými informáciami. V súčasnosti sú fazetové prehliadače použité v rôznych aplikáciách, napr. internetových obchodoch alebo rozhraniach pre vyhľadávanie informácií.

Základnou myšlienkou fazetového prehliadača je použitie fazetovej klasifikácie, ktorá opisuje významné aspekty inštancií zobrazovaného informačného priestoru a pomocou ich hodnôt umožňuje definovať ohraničenia určujúce vlastnosti prezentovaných inštancií z informačného priestoru. Používateľ má možnosť efektívne zužovať množinu prezentovaných inštancií podľa zvolených kritérií pomocou voľby jednej alebo viacerých reštrikcií a tak zmenšovať celkovú veľkosť prezentovaného informačného priestoru. Jednotlivé reštrikcie môžu byť navyše kombinované do zložených reštrikcií, čo umožňuje používateľovi vykonávať presnejšie dopyty.

Po ohraničení informačného priestoru fazetový prehliadač umožňuje ďalšiu prácu nad výslednou množinou inštancií (informácií) pomocou ich usporiadania na základe zvolených kritérií. Funkcionalitu fazetového prehliadača možno rozdeliť na nasledovné časti:

- *Ohraničovanie informačného priestoru* – zníženie veľkosti informačného priestoru pomocou definovania ohraničení nad vlastnosťami inštancií (informácií). V každej fazete je možné vybrať najviac jedno ohraničenie, na kombináciu ohraničené vo viacerých fazetách sa používa logická funkcia AND.

- *Usporiadanie* informácií alebo inštancií podľa zvoleného kritéria alebo podľa viacerých kritérií súčasne, resp. po sebe, napr. podľa názvu.
- *Pohľady* na inštanacie informačného priestoru, napr. sumárny pohľad len s hlavnými atribútmi, detailný pohľad zo všetkými atribútmi, pohľady zamerané na vybrané charakteristiky inštancií, možnosť zvoliť si počet súčasne zobrazených inštancií (napr. 10, 20 alebo 50).
- *Porovnanie* inštancií; napr. porovnanie dvoch alebo viacerých inštancií pracovných ponúk v podobe prehľadnej tabuľky obsahujúcej ich relevantné vlastnosti ako plat, pracovnú pozíciu a miesto vykonávania práce.
- *Vyhľadávanie nad ohraničenou množinou* inštancií pomocou externého nástroja.

2.4.1 Fazetová klasifikácia

Fazetový prehliadač je založený na využití *fazetovej klasifikácie* prebratej z knihovníctva, ktorú Wynar opisuje takto (Wynar, Taylor, 1992):

„Fazetová klasifikácia sa odlišuje od tradičnej klasifikácie tým, že nepriradzuje pevné umiestnenie subjektom v konkrétnom poradí, ale využíva jasne definované, vzájomne sa vylučujúce a kolektívne vyčerpávajúce aspekty, vlastnosti alebo charakteristiky triedy alebo konkrétneho subjektu. Tieto aspekty, vlastnosti alebo charakteristiky sa nazývajú *fazety* danej triedy alebo subjektu. Termín *fazeta* bol zavedený s týmto významom do teórie klasifikácie Indickým knihovníkom a klasifikátorom S. R. Ranganathanom, pričom bol prvýkrát použitý v *Colon Classification*, ktorú vytvoril začiatkom tridsiatych rokov minulého storočia.“

Znaky fazetovej klasifikácie možno nájsť vo viacerých systémoch mimo oblasť knihovníctva avšak je nepravdepodobné, že tvorcovia týchto systémov pri ich návrhu poznali formálne techniky fazetovej klasifikácie v knihovníctve. Práve pri využití vlastností fazetovej klasifikácie v iných oblastiach nie je nevyhnutné dodržať niektoré jej vlastnosti ako vyčerpávanosť faziet (The Knowledge Management Connection, 2006).

Medzi hlavné výhody fazetovej klasifikácie v porovnaní s tradičnou klasifikáciou patria (He et al., 2003):

- Zameranie sa na význačné charakteristiky obsahu, vhodné pre podrobné rýchlo sa meniace úložiská dát.
- Jednoduchá rozširiteľnosť o nové fazety.
- Flexibilita voči zmenám vo všeobecnosti.
- Jednoduchšia konštrukcia fazetových hierarchií.
- Vysoká efektivita pri použití v systémoch pre ukladanie a vyhľadávanie informácií.
- Podpora viacrozmerných dopytov pomocou kombinácie jednotlivých faziet vhodné pre prostredie webu.

Medzi nevýhody fazetovej klasifikácie patrí skutočnosť, že neumožňuje rýchle pochopenie rozsahu a obsahu domény a rýchly prístup k populárnym témam. Ďalším z problémov je fakt,

že v súčasnosti neexistujú formálne štandardy pre tvorbu fazetových slovníkov (The Knowledge Management Connection, 2006).

2.4.2 Charakteristiky fazetových prehliadačov

Podľa spôsobu akým fazetové prehliadače využívajú fazetovú klasifikáciu je možné ich rozdeliť do viacerých skupín. Medzi hlavné charakteristiky fazetových prehliadačov patria (Adkisson, 2005):

- *Pokročilé vyhľadávanie vs. fazetová navigácia*, ktoré hovoria o spôsobe akým používatelia pristupujú k fazetovej klasifikácii. *Pokročilé vyhľadávanie* umožňuje používateľom pomocou formulára definovať dopyt - ohraničenia pre jednotlivé fazety a vykonať zvolený dopyt. Naproti tomu, *fazetová navigácia* umožňuje používateľom postupne konštruovať výsledný dopyt pomocou nasledovania postupnosti odkazov.
- *Rovnocenné fazety vs. hlavné a vedľajšie fazety*, ktoré opisujú preferovanie vybraných faziet pred ostatnými. *Rovnocenné fazety* majú rovnakú polohu a štýl prezentácie. *Hlavné a vedľajšie fazety* uprednostňujú zobrazenie hlavných faziet napr. ich viditeľnejším umiestnením, väčšou veľkosťou na obrazovke, zobrazením ich hodnôt a skrytím hodnôt vedľajších faziet.
- *Jednoúrovňové vs. viacúrovňové fazety*, ktoré opisujú hĺbku hierarchie hodnôt použitej fazetovej klasifikácie pre jednotlivé fazety. *Jednoúrovňové* poskytujú len jednoduchý zoznam hodnôt pre každú fazetu, zatiaľ čo *viacúrovňové* ponúkajú progresívny zoznam hodnôt faziet pre viaceré úrovne hierarchií z fazetovej klasifikácie.

Podľa štúdie z roku 2003 (Adkisson, 2005), približne 69% zo skúmaných webových obchodov využívalo fazetovú klasifikáciu v nejakej podobe. Z nich 77% využívalo fazetovú navigáciu, 6% pokročilé vyhľadávanie a 17% využívalo oba spôsoby. Väčšina (65%) webových sídel uprednostnila využitie rovnocenných faziet a jednoúrovňové fazety (67%).

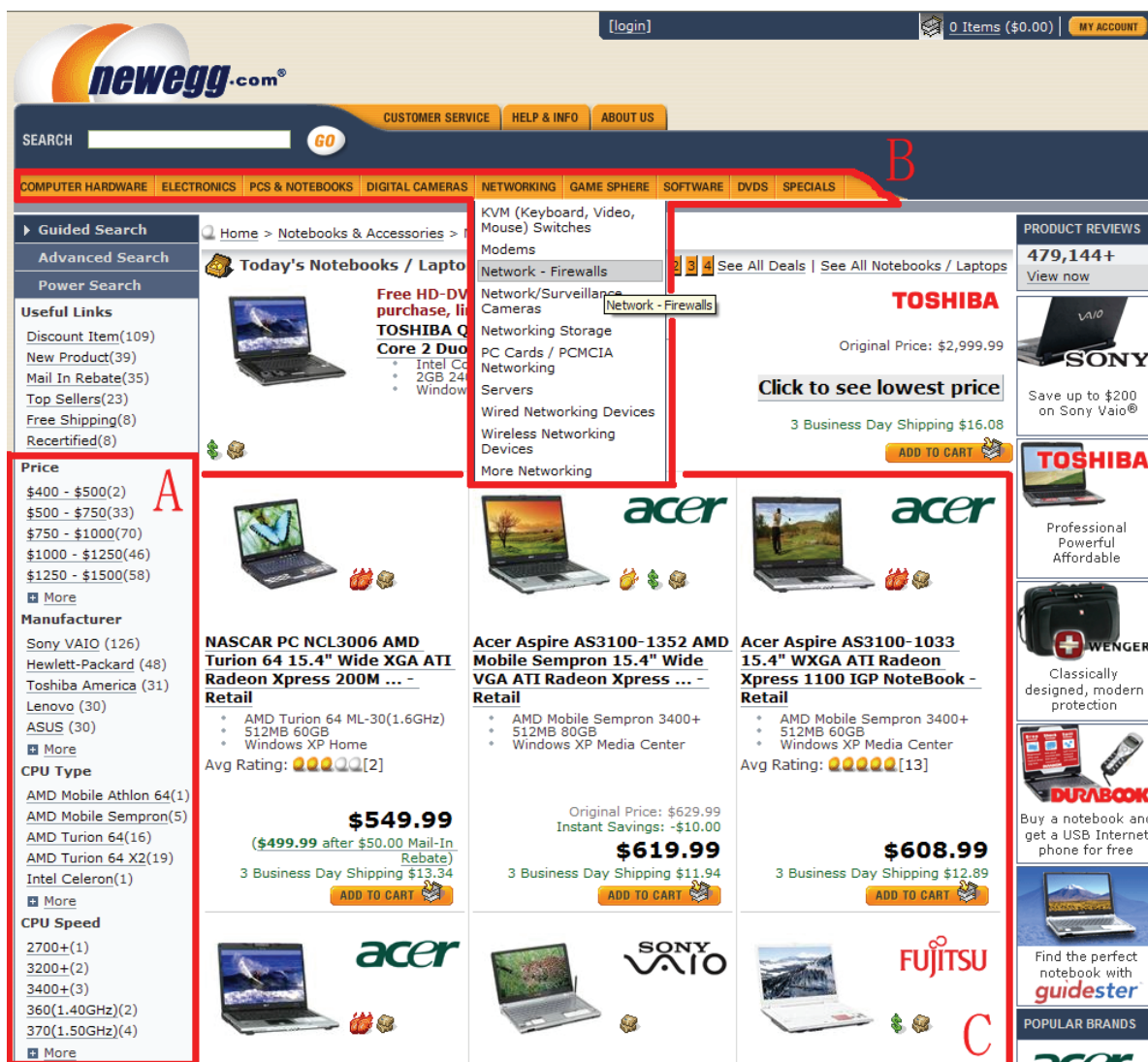
Medzi dôležitú charakteristiku fazetových prehliadačov patria aj jednotlivé typy faziet:

- *Hierarchické klasifikácie*, predstavujúce viacúrovňovú stromovú hierarchiu tried inštancií s nominálnymi hodnotami (napr. taxonómie živočíchov, rastlín, miesto vykonávania práce, profesia).
- *Číselníky*, predstavujúce jednoúrovňovú hierarchiu nominálnych hodnôt (napr. typ pracovného pomeru, farba výrobku).
- *Intervaly*, reprezentujúce veľkosť niektorej vlastnosti inštancií v podobe intervalov numerických hodnôt – celých alebo reálnych čísel (napr. plat, dátum, hmotnosť, cena).

Využitie hierarchií a číselníkov je podobné, pričom číselník možno z konceptuálneho hľadiska pokladať za podmnožinu hierarchií. Intervaly možno využiť priamo, definovaním fazety pomocou vopred určených intervalov alebo vo forme intervalov zadaných používateľom. Obrázok 5 obsahuje ukážku rozhrania fazetového prehliadača.

Ohraničenie informačného priestoru možno dosiahnuť kombináciou ohraničení z jednej alebo viacerých faziet súčasne, pričom možno kombinovať aj ohraničenia na rôznych

úrovniah hierarchie. Pokročilou možnosťou ohraničenia informačného priestoru je využitie inej funkcie na kombináciu hodnôt vo fazetách ako je AND, napr. funkcie OR a možnosť zvolit' aj viaceré položky v jednej fazete na rovnakej úrovni hierarchie súčasne.



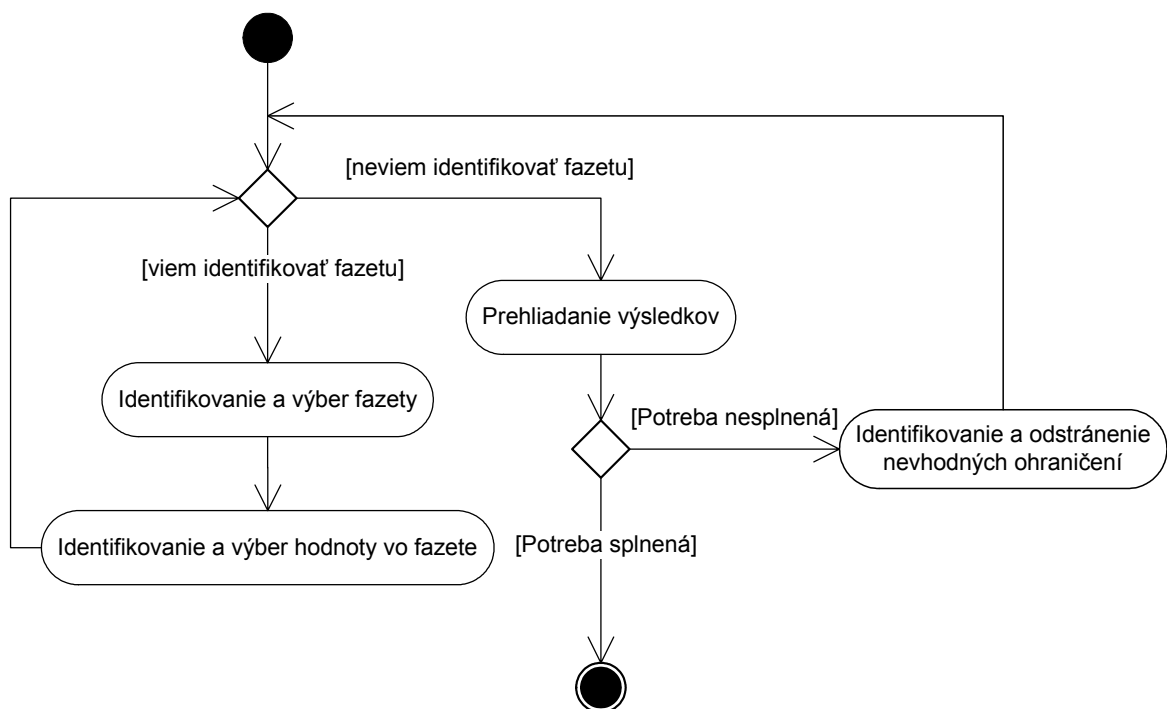
Obrázok 5: Ukážka rozhrania fazetového prehliadača s jednoúrovňovými rovnocennými fazetami – intervalmi a číselníkmi (A). Výber typu inštancií je riešený globálnou hierarchickou navigáciou (B), výsledky vyhľadávania sú zobrazené v matici (C).

2.4.3 Navigácia vo fazetovom prehliadači

Fazetový prehliadač umožňuje základné vyhľadávanie pomocou ohraňovania informačného priestoru. Spôsob navigácie v informačnom priestore doménovej ontológie pomocou fazetového prehliadača je nasledovný (pozri Obrázok 6):

- 1) Navigácia začína bez definovaných ohraňení, kedy je viditeľný celý informačný priestor.

- 2) Pokiaľ používateľ vie definovať požadované vlastnosti hľadaných inšancií pokračuje definovaním ohraňení v kroku 3, inak pokračuje krokom 5.
- 3) Používateľ identifikuje a vyberie fazetu (vlastnosť ponuky), nad ktorou chce definovať ohraňenie (napr. miesto vykonávania práce).
- 4) Používateľ identifikuje a vyberie hodnotu vo zvolenej fazete, na ktorú chce ohraňiť hodnoty zodpovedajúcej vlastnosti prehliadaných inšancií (napr. New York). Po zvolení ohraňenia sa spravidla zmenší veľkosť viditeľného informačného priestoru, ktorý bude obsahovať len inšancie spĺňajúce zvolené ohraňenie. Pokračuje krokom 2.
- 5) Používateľ sekvenčne prehliada výsledné inšancie (napr. pracovné ponuky) a podľa potreby zmení kritérium ich usporiadania.
- 6) Ak používateľ našiel hľadané inšancie a jeho potreba je uspokojená, ukončí prácu s prehliadačom. Ak však inšancie vo zvolenej časti informačného priestoru neuspokojili jeho potreby, pokračuje odstránením niektorého z definovaných ohraňení vo fazetách – identifikuje „nevhodné“ ohraňenie, odstráni ho z množiny použitých ohraňení a pokračuje krokom 2.



Obrázok 6: Postup navigácie vo fazetovom prehliadači.

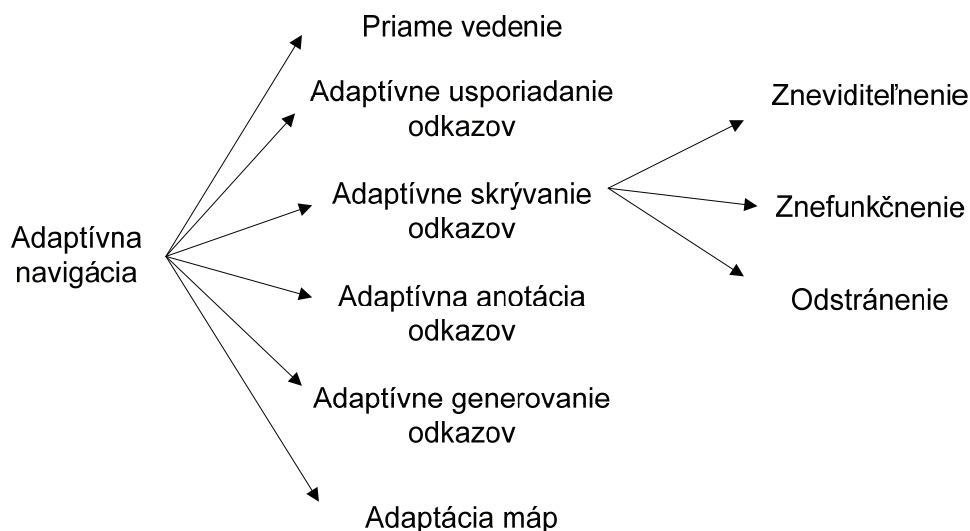
3 Adaptívna navigácia

Adaptívna navigácia predstavuje prístup k riešeniu navigácie založený na prispôbovaní sa potrebám a požiadavkám používateľa. Adaptívne webové systémy využívajú techniky adaptívnej navigácie na dosiahnutie konkrétnych cieľov adaptácie. Vychádzajú pritom z predpokladu, že jedna prezentácia informačného priestoru a navigácie v ňom nemôže vyhovovať všetkým používateľom.

Ciele adaptívnej navigácie sú odvodené od cieľov navigácie vo všeobecnosti (zamedziť strateniu sa, informovať o aktuálnej polohe a možnostiach ďalšej navigácie) so zameraním sa na možnosti zefektívnenia a zjednodušenia práce jednotlivých používateľov so systémom. Adaptívna navigácia kladie dôraz na prispôbovanie existujúcich a prípadné generovanie nových navigačných prvkov stránok tak, aby čo možno najviac vyhovovali konkrétnemu používateľovi, resp. skupine používateľov.

3.1 Techniky adaptívnej navigácie

Techniky adaptívnej navigácie možno rozdeliť do viacerých skupín podľa ich funkcie a určenia. Obrázok 7 obsahuje prehľad techník používaných v adaptívnej navigácii:



Obrázok 7: Techniky adaptívnej navigácie, prevzaté z (Brusilovsky, 2001).

Priame vedenie je technika, ktorá používateľovi odporúča nasledujúcu stránku, ktorú by mal navštíviť, zvyčajne v podobe odkazu „ďalej“ (angl. next). Často je využívaná najmä vo výučbových systémoch, v ktorých sa ďalšia stránka určuje na základe znalostí o poznatkoch používateľa o danom koncepte. Odporúčaním sa všeobecne zaoberajú systémy na odporúčanie (angl. recommender systems), využívajúce napr. kolaboratívne filtrovanie.

Adaptívne usporiadanie odkazov sa zaoberá zmenou poradia odkazov na stránke. Doteraz bolo využité najmä v uzavretých informačných priestoroch avšak nebolo pozitívne

prijaté používateľmi. Jeho neúspech vyplýval zo skutočnosti, že sa poradie odkazov menilo a používatelia hľadali odkazy na pôvodných miestach, kde sa však tieto už nenachádzali práve preto, že informačný priestor bol relatívne malý.

Usporiadanie odkazov spolu s *adaptívnym generovaním odkazov* však môže byť použité v priestore otvorených informačných priestorov, ktoré sa už zo svojej podstaty často menia. Práve vo veľkom informačnom priestore predstavuje usporiadanie odkazov jedinou možnosť ako nájsť požadované informácie, pričom ho z úspechom využívajú aj všetky vyhľadávače.

Z tohto dôvodu by v otvorených priestoroch zmena poradia odkazov, resp. pridanie nových odkazov nemuselo na používateľa pôsobiť rušivo. Ďalšou príbuznou technológiou je aj *adaptívne skrytie odkazov*, ktoré sa ďalej delí na:

- *Zneviditeľnenie odkazov*, kedy odkaz vyzerá ako obyčajný text, ale je možné ho nasledovať.
- *Znefunkčnenie odkazov*, kedy odkaz vyzerá ako obyčajný text a nie je možné ho nasledovať; používané hlavne na kontextové odkazy.
- *Odstránenie odkazov*, kedy je celý odkaz odstránený zo stránky; používané najmä na bezkontextové odkazy.

Odlišnou technológiou je *adaptácia máp*, ktorá sa zaoberá prispôbovaním lokálnej, resp. globálnej mapy hyperpriestoru, v ktorom sa používateľ nachádza. Poslednou a zároveň asi najpoužívanejšou technológiou je *adaptívna anotácia odkazov*, ktorá ku vybraným odkazom na stránke pridáva dodatočné informácie, ktorých cieľom je bližšie opísať, kam daný odkaz vedie, resp. ako vhodné je pre používateľa nasledovanie daného odkazu, najčastejšie farebným zvýraznením alebo rôznymi značkami.

V praxi sa adaptívna navigácia používa na dosiahnutie konkrétnych cieľov adaptácie dôležitých pre konkrétnu aplikáciu. Podľa (Brusilovsky, 1996) rozlišujeme tieto ciele adaptácie:

- *Lokálne vedenie*, ktoré pomáha používateľovi pri výbere nasledovného kroku navigácie pomocou usporiadania odkazov, resp. odporúčania odkazov na základe preferencií a znalostí používateľa.
- *Globálne vedenie*, je podobné lokálnemu vedeniu a vedie používateľa k splneniu jeho globálneho informačného cieľa pomocou odporúčania cesty hyperpriestorom tak, aby táto bola čo najkratšia. V prípade výučbového systému by používateľa previedlo len tými konceptmi, ktoré sú nevyhnutné na zvládnutie zvoleného cieľového konceptu s minimálnymi „odbočkami“.
- *Podpora lokálnej orientácie*, ktorej cieľom je poskytnúť používateľovi informáciu o jeho polohe v hyperpriestore a jeho najbližšom okolí. Najčastejšie sa používa anotácia odkazov na okolité stránky, ktorá poskytuje používateľovi informáciu o ich vhodnosti a zneviditeľnenie menej dôležitých odkazov, ktoré znižuje zahltenie používateľa odkazmi.
- *Podpora globálnej orientácie*, ktorej cieľom je pomôcť používateľovi pochopiť štruktúru celého hyperpriestoru ako aj jeho absolútnu polohu v ňom, pričom využíva predovšetkým anotáciu odkazov.

3.2 Realizácia adaptívnej navigácie

Nutným predpokladom pre prispôsobovanie je existencia modelu používateľa, ktorý obsahuje relevantné informácie špecifické pre konkrétneho používateľa. Model používateľa je v ideálnom prípade previazaný s modelom domény, čo umožňuje v modeli používateľa priamo zaznamenávať charakteristiky používateľa vzťahujúce sa k doménovým konceptom. Samotné prispôsobovanie využíva údaje z modelu používateľa a pomocou definovaného mechanizmu prispôsobovania využíva techniky adaptívnej navigácie na prispôbenie existujúceho navigačného modelu za účelom dosiahnutia konkrétnych cieľov navigácie. Znalosti o prispôsobovaní môžu byť „natvrdo“ vložené do zdrojového kódu adaptívneho systému. V pokročilých systémoch je model adaptácie definovaný samostatne, najčastejšie s využitím pravidiel, pričom konkrétne pravidlá prispôsobovania sú definované napr. v podobe konfiguračných súborov. Definovanie formálneho modelu adaptácie a spôsobu zápisu pravidiel prispôsobovania je nutné pre tvorbu generických adaptívnych systémov (De Bra et al., 2000).

Model používateľa

Model používateľa obsahuje relevantné údaje o používateľovi na základe ktorých je možné prispôbovať správanie sa systému. Z hľadiska typov modelu používateľa rozlišujeme *prekryvný model*, kde každý používateľ má vytvorenú vlastnú kópiu všetkých relevantných údajov a *stereotypný model*, ktorý zdieľajú viacerí používatelia. Prekryvný model je vhodný pre prispôsobovanie sa konkrétnemu používateľovi, sú s ním však spojené problémy pri jeho prvotnej inicializácii (angl. cold start problem). Problém inicializácie modelu pre používateľa odpadá pri stereotypnom modeli, ktorý používateľa zaradí do jednej z (vopred) pripravených skupín – stereotypov. Stereotypný model ale systému neumožňuje prispôbovať sa konkrétnemu používateľovi, ale len skupine používateľov.

Samotný model používateľa možno rozdeliť na *doménovo nezávislú časť* obsahujúcu všeobecné charakteristiky používateľa (napr. vek, pohlavie) a *doménovo závislú časť* obsahujúcu vlastnosti používateľa súvisiace s konkrétnou aplikačnou doménou (napr. očakávaný plat pre doménou ponúk pracovných príležitostí alebo oblasti záujmu v prípade vedeckých publikácií). Model používateľa je možné reprezentovať viacerými spôsobmi, pričom v kontexte webu so sémantikou je vhodnou reprezentáciou ontológia používateľa.

Naplnenie modelu používateľa charakteristikami používateľa zabezpečuje modelovanie používateľa, pričom podľa (Brusilovsky, 1996) je model používateľa možné vytvoriť na základe rôznych typov informácií:

- *Znalostí používateľa* – toho, čo vie (napr. výborne ovláda matematickú analýzu, ale neovláda teóriu elektromagnetického poľa).
- *Cieľov používateľa* – toho, čo chce (práve) dosiahnuť (napr. vykonať prevod z účtu, pochopiť princípy elektromagnetizmu, nájsť si dobré pracovné miesto).

- *Zázemia používateľa* – toho, čím je (napr. profesia, pracovné skúsenosti, názory).
- *Skúsenosti používateľa* – toho, ako dobre pozná štruktúru hyperpriestoru a ako dobre sa v ňom vie pohybovať (napr. žiak základnej školy, alebo vysokoškolský profesor).
- *Preferencií používateľa* – toho, čo sa mu páči alebo nepáči (napr. má rád veľryby a zelenú farbu, nemá rád reklamu).

Nevyhnutnosť aktuálneho modelu používateľa pre efektívne prispôsobovanie vyžaduje jeho priebežnú aktualizáciu tak, aby vždy obsahoval skutočné charakteristiky používateľa. Z tohto dôvodu je dôležitá spolupráca softvérových nástrojov pre navigáciu a nástrojov pre modelovanie používateľa. Potrebné údaje je možné získať pomocou nástrojov na zachytenie preferencií používateľa buď plne automaticky (Barla, 2006a) alebo za účasti používateľa.

Používateľsky najjednoduchším spôsobom je plne automatický systém, ktorý všetky relevantné údaje zistí bez zásahu používateľa. V praxi však nie je možné získať všetky potrebné údaje automaticky, čo je spôsobené nedokonalosťou nástrojov ako aj skutočnosťou, že zistené údaje nie sú úplne spoľahlivé (Brusilovsky, 1996).

Najjednoduchšou a najčastejšie používanou metódou získania údajov od používateľa je vyplnenie dotazníka. Nevýhodou tejto metódy je nevyhnutnosť zapojenia používateľa a zdĺhavosť v prípade veľkého počtu potrebných údajov. Pridruženým problémom je skutočnosť, že sám používateľ nemusí byť schopný (ochotný) objektívne vyplniť všetky požadované údaje.

Výhodou metód automatického získavania charakteristík používateľa je ich používateľská nenáročnosť a vyššia záruka získania relevantných charakteristík v porovnaní s dotazníkovým prístupom (používateľa len zriedkakedy podrobne a objektívne vyplní všetky údaje). V (Andrejko et al., 2006a) autori opisujú automatizovanú metódu modelovania používateľa s využitím zaznamenávania akcií používateľa a ich následnej analýze s vyhodnotením implicitnej spätnej väzby a sledovania konzistencie správania sa.

3.3 Navigácia v existujúcich adaptívnych systémoch

Existuje viacero typov adaptívnych webových systémov. Medzi ich hlavné charakteristiky patrí ich účel, ďalej veľkosť informačného priestoru, s ktorým pracujú a skutočnosť, či je tento priestor otvorený alebo uzavretý. Niektoré systémy slúžia primárne na vyhľadávanie informácií (WebWatcher), iné na prácu s informáciami (Hynecosum) alebo na štúdium (ISIS-Tutor, ELM-ART, InterBook) (Brusilovsky, 2004). Asi najčastejšie sa vyskytujúci systémami sú výučbové systémy pracujúce s uzavretým informačným priestorom. Viaceré z metód adaptívnych webových systémov možno využiť aj v iných aplikačných oblastiach napr. v trojrozmerných virtuálnych prostrediach pre múzeá (Hughes et al., 2002).

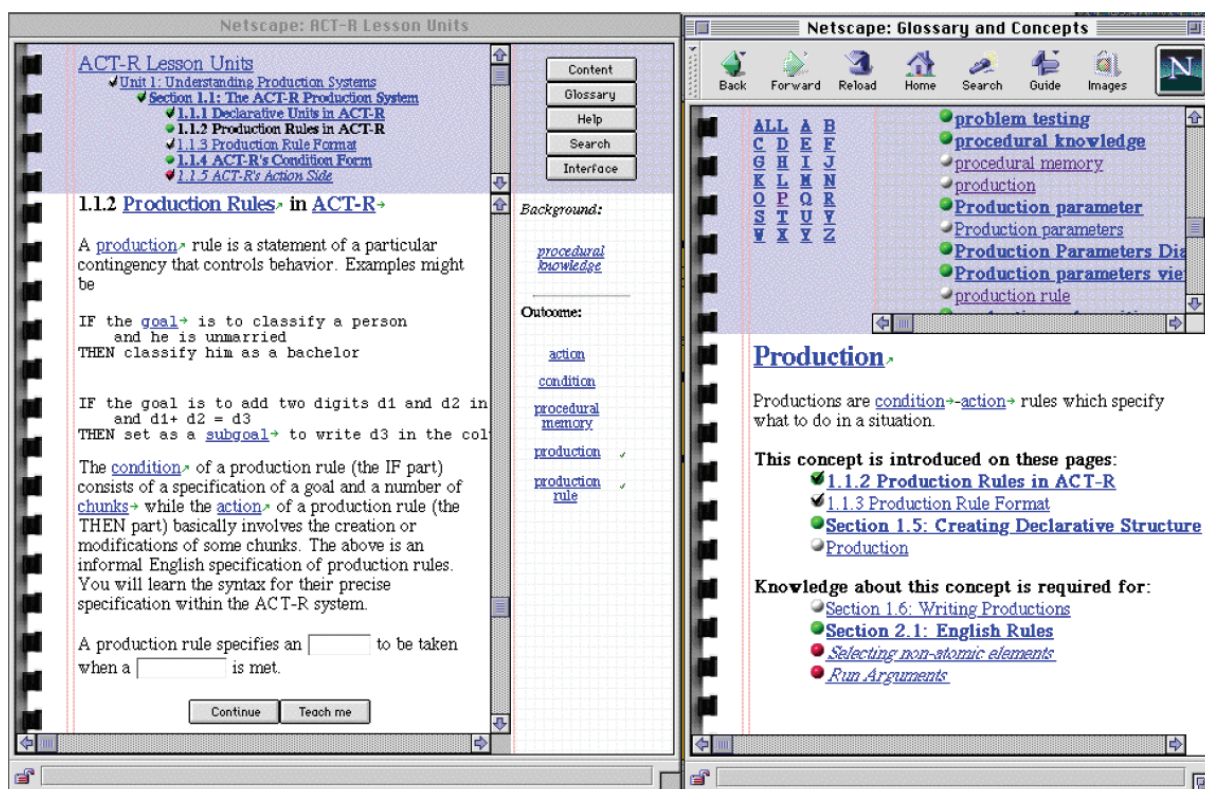
Pre výskumné i výučbové účely sa často používa generický adaptívny systém AHA! (De Bra et al., 2003a). Tento je založený na referenčnom modeli AHAM (De Bra et al., 1999), ktorý definuje pre adaptívny systém tri základné modely – model domény, model adaptácie a model používateľa. Na základe pravidiel z modelu adaptácie AHA! upravuje hodnoty atribútov vybraných konceptov v prekryvnom modeli používateľa. AHA! definuje rekurzívny

spôsob aktualizácie modelu používateľa pomocou pravidiel adaptácie s podporou pre cykly a nemonotónnu aktualizáciu (De Bra et al., 2000).

Systém AHA! využíva *adaptívne skrývanie a anotáciu odkazov a adaptívnu prezentáciu konceptov*, resp. ich fragmentov, čím podporuje *lokálnu orientáciu a vedenie*. Odkazy označuje troma farbami s významom zlý, neutrálny, dobrý na základe (ne)navštívenia dokumentu a množiny predpokladov. *Adaptívnu prezentáciu* fragmentov AHA! realizuje pomocou vyhodnotenia ich vhodnosti.

V oblasti výučbových systémov sa na FIIT STU používa systém ALEA (Kostelník, Bieliková, 2003). Model domény v systéme ALEA predstavuje graf, ktorého vrcholy sú koncepty a hrany vzťahy medzi nimi. Samotné koncepty neobsahujú obsah, ktorý je prezentovaný používateľovi, ale len mapovanie medzi konceptmi a samotným obsahom, ktorý je uložený zvlášť v báze informačných fragmentov (napr. texty, obrázky). ALEA vyhodnocuje poradie prezentácie konceptov a im prislúchajúcich informačných fragmentov ako aj ich prezentáciu. Systém umožňuje navigáciu v priestore konceptov, pričom využíva *anotáciu odkazov* rôznymi farbami a *odporúčanie konceptov* na základe modelu používateľa, čím realizuje *lokálne a globálne vedenie* v priestore konceptov a *podporu lokálnej orientácie*.

Systém InterBook (Eklund et al., 1999) predstavuje adaptívnu elektronickú knihu. Obrázok 8 obsahuje ukážku *adaptívnej anotácie odkazov*, realizovanej na základe doménových znalostí, histórie navigácie a predpokladov nutných pre navštívenie konceptu.



Obrázok 8: Ukážka adaptívnej anotácie v systéme InterBook v podobe farebných odrážok zodpovedajúcich svetlám na semafore; prevzaté z (De Bra et al., 2003b).

V ľavom hornom rohu sa nachádza časť obsahujúca obsah kurzu s adaptívnou anotáciou jednotlivých konceptov pomocou farieb semaforu (zelenej a červenej) a typu písma (hrubé, kurzíva). Vyznačené sú tak odporúčané ako aj už preštudované koncepty. V pravej časti sa nachádzajú odporúčania na ďalšie štúdium, ktoré využívajú *dynamické generovanie a adaptívne usporiadanie odkazov*. Systém tak realizuje *lokálne vedenie* používateľa k stanovenému cieľu s *podporou lokálnej a globálnej orientácie pomocou anotácie odkazov*.

Opísané adaptívne systémy využívajú rôzne techniky na dosiahnutie vybraných cieľov adaptácie so zameraním sa na podporu *lokálnej orientácie a lokálneho vedenia* a čiastočnú podporu *globálnej orientácie a vedenia*. Z hľadiska techník adaptácie navigácie sa najčastejšie používa *anotácia odkazov* spojená s *odporúčaním*. Používa sa tiež *adaptívne generovanie a usporiadanie odkazov*, ako aj techniky *adaptívnej prezentácie*. Dôležité je modelovanie domény a pravidiel prispôsobovania, pomocou modelu domény a adaptácie (model AHAM).

4 Zhodnotenie súčasného stavu a ciele

V predchádzajúcich kapitolách sme analyzovali súčasný stav a identifikovali súčasné problémy navigácie vo veľkých informačných priestoroch. Medzi najvýznamnejšie z nich patrí práca s informáciami vo veľkých otvorených informačných priestoroch, náročnosť vyhľadania informácií, „stratenie sa“ používateľov a neprívetivosť webových sídel. Analyzovali sme existujúce prístupy a aktuálny stav v oblasti navigácie vo webových informačných systémoch. Zamerali sme sa na hľadanie a analýzu možných smerov riešenia v kontexte projektov NAZOU a MAPEKUS, pričom za hlavné smery riešenia sme prijali využitie adaptívnej navigácie v prostredí fazetového prehliadača a použitie ontológií na reprezentáciu dát. (Príloha A opisuje základné princípy webu so sémantikou.)

Súčasný stav v oblasti navigácie možno hodnotiť z dvoch pohľadov. Adaptívne prispôbovanie sa systémom používateľovi je v otvorenom priestore webu zatiaľ zriedkavé, častejšie sa však vyskytujú adaptabilné systémy, ktoré umožňujú používateľovi manuálne si prispôbiť niektoré charakteristiky systému. Veľmi často sa vyskytujú systémy na odporúčanie obsahu (napr. internetové obchody), avšak tieto väčšinou nepracujú na základe poznania charakteristík používateľa ale na základe sociálnych alebo štatistických modelov správania sa skupín používateľov.

Navigačná štruktúra webových sídel vo veľkej miere využíva globálnu hierarchickú navigáciu v rámci sídel s definovanou hierarchickou klasifikáciou informácií. Celková kvalita a prehľadnosť navigačnej štruktúry je značne rozdielna, pričom existujú tak prepracované webové sídla s veľmi dobre riešenou viacúrovňovou hierarchickou navigáciou ako aj sídla s minimálnym využitím pokročilých navigačných prvkov.

Veľmi častý typ webových sídel predstavujú internetové obchody alebo iné portály pracujúce s veľkým informačným priestorom. Tieto využívajú často jednorozmernú hierarchickú klasifikáciu informačného obsahu, čo v praxi sťažuje prácu s informačným obsahom. Prepracovanejšie riešenia používajú tak hierarchickú navigáciu ako aj fazetovú navigáciu pre tie časti informačného priestoru, kde to dáva zmysel (napr. www.newegg.com).

Cieľom analýzy bolo najmä hľadať a nájsť možné riešenia identifikovaných problémov navigácie a následne z nich vybrať a bližšie analyzovať, tie ktoré použijeme na riešenie uvedených problémov.

Cieľom práce je rozšíriť a vylepšiť existujúce prístupy k navigácii:

- Navrhnuť metódu personalizovanej navigácie v otvorenom informačnom priestore reprezentovanom doménovou ontológiou na základe modelu používateľa, s podporou pre zaznamenávanie akcií používateľa so sémantikou za účelom modelovania používateľa pre prispôbovanie.
- Overiť navrhnutú metódu implementovaním prototypu jej vybraných častí a vykonaním experimentov v doméne pracovných príležitostí a vedeckých publikácií.

Na základe výsledkov analýzy sme identifikovali tieto smery riešenia:

- Ontologická reprezentáciu modelov domény a používateľa s použitím modelov z projektov NAZOU a MAPEKUS.
- Fazetová klasifikácia a fazetový prehliadač ako vhodný spôsob navigácie a vyhľadávania vo veľkom informačnom priestore.
- Prispôsobovanie s využitím adaptívnej navigácie, ktorá zefektívňuje prácu používateľa s fazetovým prehliadačom pri práci s veľkým informačným priestorom. Pomocou prispôsobovania je tiež možné efektívne kompenzovať viaceré nedostatky fazetovej klasifikácie ako sú problém pochopenia rozsahu a ťažký prístup k populárnym témam, napr. využitím anotácie odkazov a odporúčania (pozri časť 2.4.1).

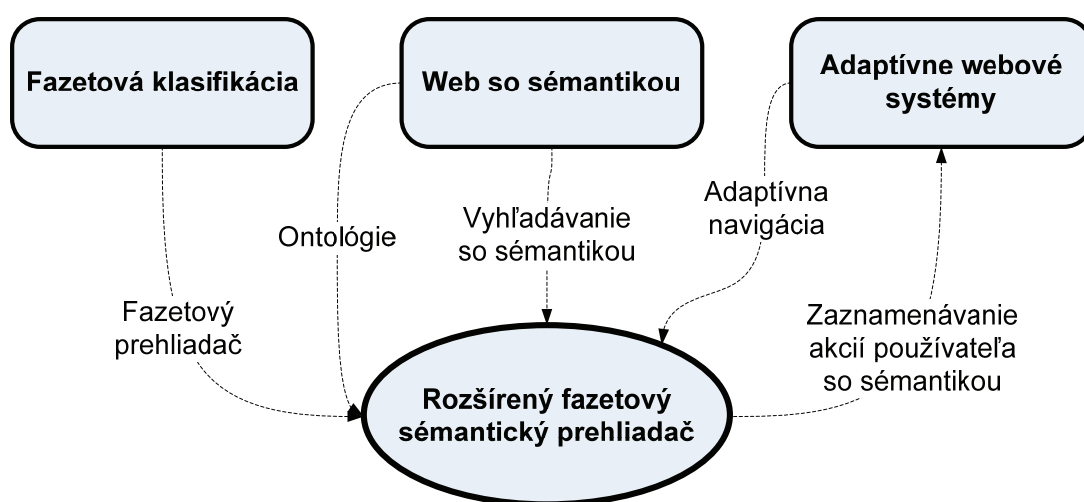
5 Metóda personalizovanej navigácie so sémantikou

V tejto kapitole opíšeme metódu navigácie so sémantikou a bližšie rozpracujeme jej návrh do podoby konkrétneho softvérového nástroja – *adaptívneho fazetového sémantického prehliadača*. Zdôrazníme úlohu jednotlivých technológií v navrhnutom riešení a vyjadríme sa k spôsobu ich spolupráce a vzájomného dopĺňania sa. Opíšeme spôsob reprezentácie údajov pomocou doménovej ontológie a ontológie používateľa, ako aj základné vlastnosti navrhnutého prehliadača, jeho funkcionality a vzťah k napĺňaniu modelu používateľa.

5.1 Prehľad metódy

Navrhnutá metóda personalizovanej navigácie v informačnom priestore reprezentovanom ontológiou je založená na troch hlavných princípoch (Obrázok 9):

- Reprezentácii údajov pomocou ontológií (model domény a model používateľa).
- Použití fazetového prehliadača na navigáciu a vyhľadávanie v informačnom priestore.
- Použití techník adaptívnej navigácie na základe modelu používateľa.



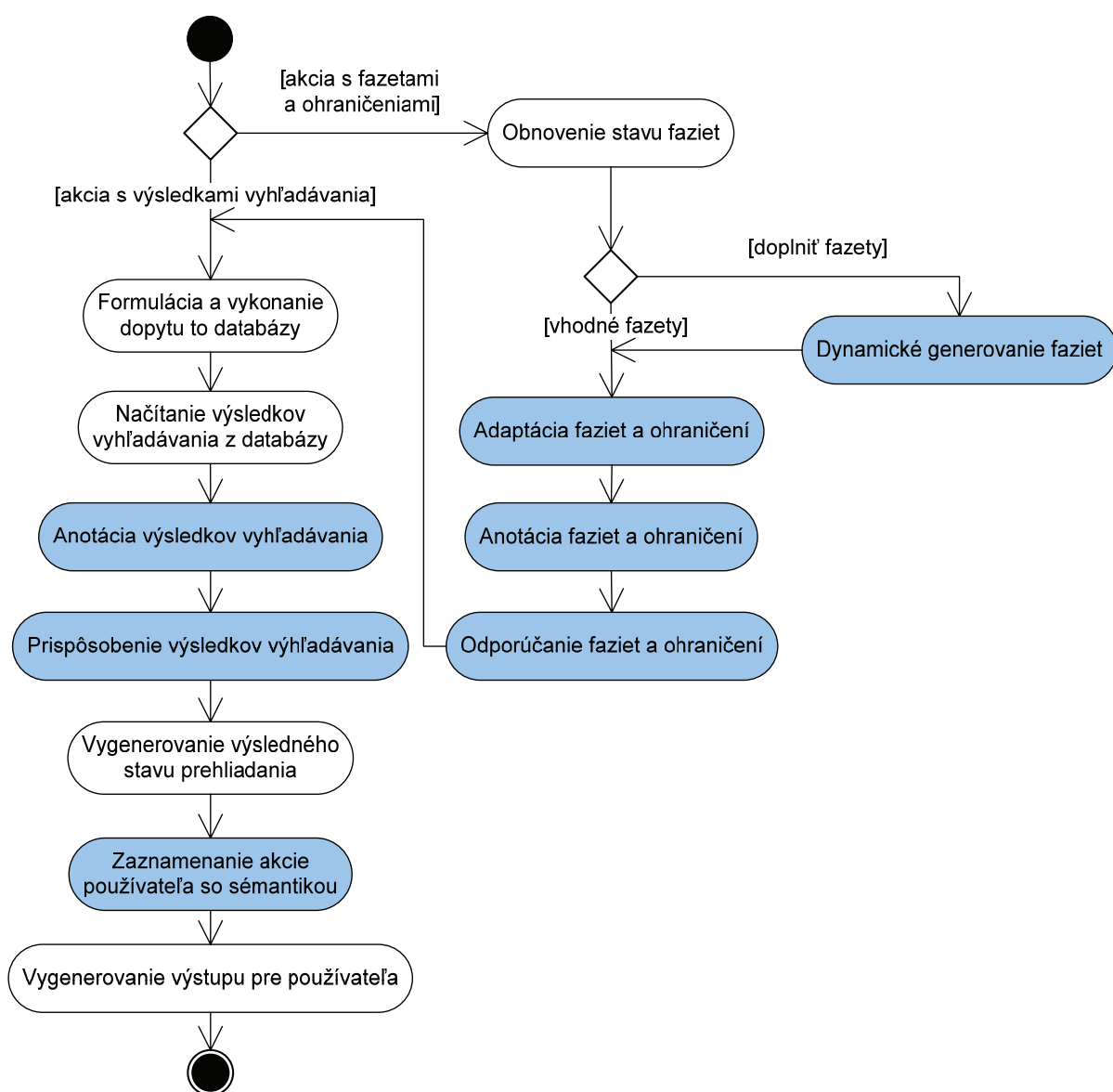
Obrázok 9: Princípy metódy navigácie pomocou rozšíreného fazetového sémantického prehliadača.

Rozšírený fazetový sémantický prehliadač realizuje navigáciu v ontológii pomocou fazetovej navigácie, ktorú prispôbuje aktuálnym požiadavkám používateľa na základe jeho (doménových) charakteristík uložených v modeli používateľa. Vizuálne definovanie ohraničení pomocou navigácie v ontológii umožňuje používateľovi vytvoriť dopyt do databázy a predstavuje tak obdobu vyhľadávania so sémantikou. Samotné ontológie predstavujú explicitný formálny spôsob zápisu sémantiky vo webe so sémantikou (Príloha A).

Obrázok 10 znázorňuje spracovanie požiadavky používateľa pomocou metódy personalizovanej navigácie so sémantikou realizovanej rozšíreným fazetovým sémantickým prehliadačom. Postup spracovania požiadavky pomocou klasického fazetového prehliadača

znázorňujú činnosti vyznačené bielou farbou. Navrhnutá metóda navigácie rozširuje spracovanie klasického fazetového prehliadača o činnosti vyznačené modrou farbou.

Spracovanie faziet sme rozšírili o doplnenie množiny dostupných faziet pomocou dynamického generovania faziet a príslušných ohraničení, ich následnú adaptáciu, anotáciu a odporúčanie (Obrázok 10, vpravo). Spracovanie výsledkov vyhľadávania sme rozšírili o ich anotáciu s využitím modelu používateľa a prispôbovanie pomocou voľby pohľadu, resp. zvýraznenia vybraných informácií (Obrázok 10, vľavo). Na podporu modelovania používateľa sme navrhli zaznamenávanie akcií používateľa so sémantikou pre účely ďalšieho spracovania nástrojmi na odvodenie charakteristík používateľa (Obrázok 10, dole).



Obrázok 10: Spracovanie požiadavky používateľa pomocou rozšíreného fazetového sémantického prehliadača.

Postup navigácie používateľa v rozšírenom fazetovom sémantickom prehliadači je totožný s postupom navigácie v klasickom fazetovom prehliadači (Obrázok 6, s. 13). Rozdiely sú

najmä v spôsobe realizácie jednotlivých krokov, avšak celková povaha navigácie sa nemení. Napr. pri výbere fazety ju môže byť najprv nutné aktivovať, alebo používateľ môže pri výbere fazety najprv zjemniť niektorú z existujúcich faziet a následne si vybrať niektorú z dynamicky vygenerovaných faziet. Pri výbere ohraničení si používateľ môže vybrať aj na základe odporúčania alebo informácií poskytnutých anotáciou odkazov.

5.1.1 Reprezentácia údajov ontológiou

Navrhnutá metóda používa ontológie ako spôsob reprezentácie a uloženia údajov o konkrétnej aplikačnej doméne ako aj o používateľovi. Budeme hovoriť o doménovej ontológii, ktorá obsahuje opis konceptov zo zvolenej domény, opis ich vzťahov medzi nimi ako aj inštancie konceptov samotných. Podobne budeme hovoriť o ontológii používateľa, ktorá obsahuje opis charakteristík používateľa relevantných pre zvolenú aplikačnú doménu, pričom niektoré koncepty budú aj doménovo závislé. Ontológiu používateľa možno vytvoriť jej odvodením od doménovej ontológie a doplnením o všeobecné doménovo nezávislé charakteristiky.

Samotná ontológia predstavuje štruktúrovanú bázu informácií, pričom obsahuje aj ich opis v podobe konceptov a vzťahov medzi nimi (metadáta), čo značne rozširuje možnosti jej automatizovaného spracovania. Výhodou použitia ontológií je možnosť práce so štruktúrou informácií (metadátami) rovnakým spôsobom ako s informáciami samotnými, čo zvyšuje flexibilitu voči zmenám v prostredí otvorených informačných priestorov ako aj možnosti strojového spracovania informácií s definovanou sémantikou.

Definovanú sémantiku dát využívame aj na zefektívnenie modelovania používateľa na základe zaznamenávania akcií používateľa so sémantikou. Adaptívny sémantický fazetový prehliadač zaznamenáva udalosti vyvolané používateľom počas jeho práce so systémom spolu s ich významom a zmenami stavu, ktoré vyvolali. Na zaznamenanie ich sémantiky a následné odvodenie charakteristík používateľa využíva ontológiu udalostí odvodenú od navigačného modelu prehliadača a koncepty z doménovej ontológie.

5.1.2 Fazetový prehliadač ako prostriedok navigácie

Fazetový prehliadač je určený na navigáciu a vyhľadávanie informácií vo veľkých, komplexne štruktúrovaných informačných priestoroch. Pomocou fazetovej navigácie môže používateľ rýchlym a relatívne intuitívnym spôsobom definovať tú časť priestoru, ktorá ho práve zaujíma. Informácie v ohraničenej časti informačného priestoru si môže používateľ priamo prezerať. Rovnako môže ohraničenie informačného priestoru ďalej zjemňovať a ešte viac zúžiť množstvo dostupných informácií.

Fazetový sémantický prehliadač – fazetový prehliadač, ktorý pracuje nad údajmi s definovanou sémantikou, umožňuje používateľovi tvorbu a vykonanie dopytov so sémantikou. Napr. voľba ohraničenia „práca v Bratislave“ neznamena, že vyhovujúca ponuka obsahuje text „Bratislava“, ako pri fulltextovom vyhľadávaní, ale že má v ontológii

definovanú vlastnosť „miesto vykonávania práce“ v konkrétnom regióne s názvom „Bratislava“.

Definovanie ohraničení pomocou fazetového prehliadača možno považovať za (vizuálne) formulovanie a vykonanie dopytov so sémantikou nad базou informácií obsiahnutých v doménovej ontológii. Formulovanie dopytu so sémantikou používateľsky prívetivým spôsobom je významným prínosom pri vyhľadávaní, keďže práve formulácia vhodného dopytu je často používateľsky náročná.

Pre použitie fazetového prehliadača je nevyhnutné definovať fazetovú klasifikáciu, ktorá definuje význačné vlastnosti doménových konceptov – tried v ontológii, ktoré budú následne použité vo fazetách na navigáciu v priestore inštancií zvolených tried. Ontológia predstavuje výborný zdroj dát pre fazetový prehliadač, vďaka svojej prirodzenej štruktúrovanosti, definovanej sémantike a možnosti priamej konštrukcie fazetovej klasifikácie na základe metadát.

5.1.3 Prispôbovanie vo fazetovom prehliadači

Fazetový prehliadač umožňuje navigáciu používateľa v informačnom priestore. Používateľ pomocou navigácie definuje ohraničenia nad množinou faziet a prehliada výsledky vyhľadávania spĺňajúce zvolenú množinu ohraničení.

Rozšírený fazetový sémantický prehliadač zlepšuje efektívnosť práce používateľa pri práci s veľkým otvoreným informačným priestorom opísaným zložitou ontológiou. Pomocou adaptívneho výberu a usporiadania faziet a ohraničení zjednodušuje navigáciu používateľa, kde napr. namiesto piatich obrazoviek faziet a ohraničení sa zobrazí len jedna obrazovka, na ktorej budú zobrazené len relevantné fazety. Zvyšné fazety bude možné zobrazit' na vyžiadanie používateľa. Dynamické generovanie faziet na základe doménovej ontológie a modelu používateľa umožní používateľovi využiť aj také fazety a ohraničenia, ktoré pôvodne neboli definované, resp. ktoré „vznikli“ neskoršou zmenou informačného priestoru.

Nevýhody fazetovej klasifikácie, ktorá nie je ideálna pre rýchle pochopenie rozsahu údajov a efektívny prístup k populárnym témam riešime pomocou generovania, odporúčania a anotácie odkazov na relevantné informácie na základe modelu používateľa.

Na rýchlu a jednoduchú interpretáciu výsledkov vyhľadávania používame techniky adaptívnej navigácie a čiastočne aj adaptívnej prezentácie. Vhodným pridaním anotácie k výsledkom vyhľadávania zlepšujeme prehľad používateľa o zobrazených inštanciách informačného priestoru, zmenou vzhľadu prezentovaných informácií zvýrazňujeme najrelevantnejšie informácie.

Hlavné princípy prispôbovania navigácie vo fazetovom prehliadači sú:

- 1) Generovanie množiny faziet na základe doménovej ontológie a modelu používateľa.
- 2) Výber a usporiadanie vhodných faziet z definovanej množiny faziet.
- 3) Výber a usporiadanie prezentovaných hodnôt (ohraničení) vo fazetách.
- 4) Anotácia vybraných faziet a prezentovaných hodnôt faziet.
- 5) Anotácia výsledkov vyhľadávania.

5.1.4 Prispôsobovanie vo výslednom dokumente

Vyhľadávanie pomocou fazetového prehliadača nájde len stránky, resp. inštancie konceptov informačného priestoru, ktoré vyhovujú zadanému dopytu. Tieto však sami o sebe môžu obsahovať veľa informácií. Interpretácia výsledkov samotných vyžaduje efektívnu navigáciu vo výslednom dokumente a pracuje ako „vyhľadávanie vo vyhľadávaní“, keď na výslednej stránke napr. identifikuje a zvýrazní tie časti, ktoré súvisia so zadaným dopytom.

Jednoduchým príkladom je nástroj Magpie, implementovaný ako rozšírenie prehliadača Internet Explorer, ktorý stránkam priradzuje obmedzenú sémantiku a zvýrazňuje vybrané časti dokumentu (Domingue, Dzbor, Motta, 2004). Pokročilou technikou na efektívnu interpretáciu výsledkov vo fazetovom prehliadači je pridanie sady „alternatívnych faziet“, ktoré namiesto definovania ohraničení nad inštanciami určujú o aké informácie v rámci cieľovej inštancie má používateľ záujem. Tieto sú zobrazené pri prezeraní detailných informácií o inštanciách a umožňujú rýchlo a jednoducho identifikovať informácie v cieľovom dokumente, napr. ich zvýraznením.

Hlavné princípy prispôsobovania vo výslednom dokumente:

- Určenie pohľadu, resp. relevantných vlastností inštancií „alternatívnymi fazetami“.
- Zobrazenie vybraných informácií z inštancií na základe zvoleného pohľadu.
- Zvýraznenie vybraných informácií na základe daných relevantných vlastností inštancií.

5.2 Reprezentácia domény a modelu používateľa

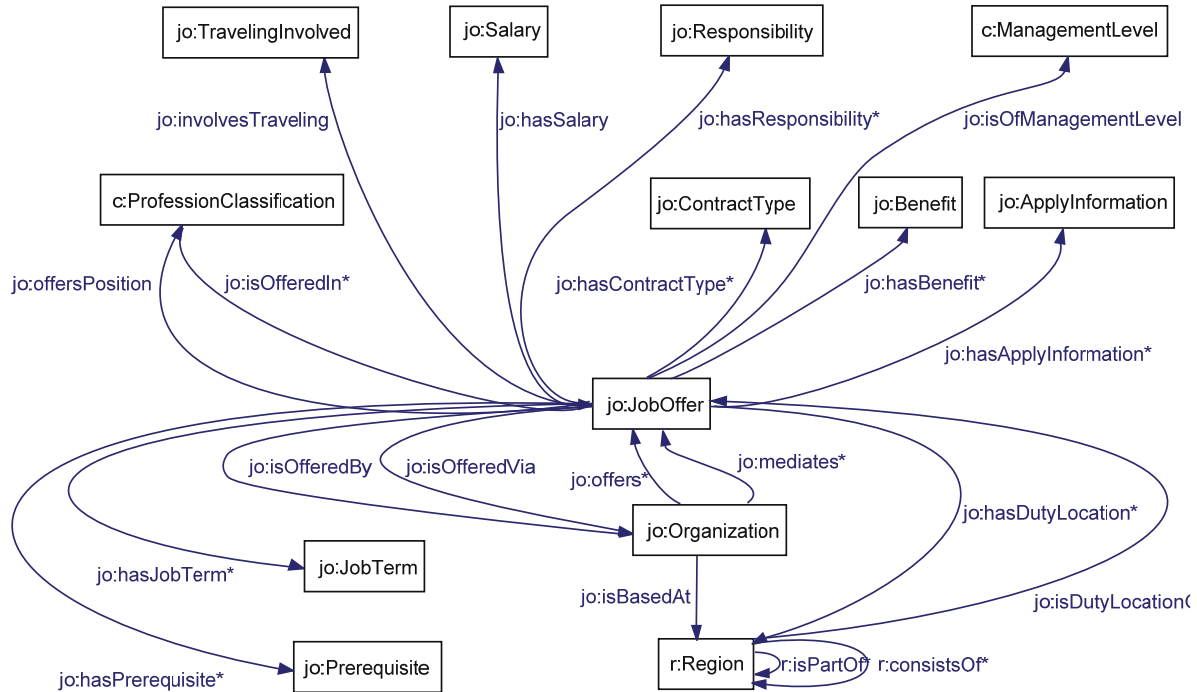
Metóda personalizovanej navigácie je založená na reprezentácii údajov pomocou ontológií a pracuje s opisom domény pomocou doménovej ontológie, opisom modelu používateľa pomocou ontológie používateľa a opisom záznamov o akciách používateľa pomocou ontológie udalostí. Na overenie metódy využívame ontológie z už spomínaných projektov NAZOU a MAPEKUS. Príloha B obsahuje podrobný opis použitých ontológií. Príklady budeme uvádzať buď z domény pracovných príležitostí alebo z domény vedeckých publikácií.

5.2.1 Doménová ontológia

Opis konceptov a vzťahov medzi nimi pre vybranú aplikačnú oblasť definuje štruktúru doménovej ontológie. Súčasne so štruktúrou doménová ontológia obsahuje aj inštancie jednotlivých konceptov s ich atribútmi, ktoré spolu tvoria dáta naplnené v doménovej ontológii. V doméne pracovných ponúk predstavuje štruktúru ontológie opis pracovnej ponuky a jej vlastností, opis požiadaviek na uchádzačov, opis klasifikácií priemyselných odvetví, vedných odborov, profesií atď. Samotné inštancie pracovných ponúk, ktoré opisujú konkrétne voľné pracovné miesta, predstavujú dáta naplnené v ontológii.

Obrázok 11 znázorňuje prehľad objektových vlastností hlavnej triedy ontológie pracovných ponúk *jo:JobOffer*. Ontológia definuje vlastnosť pre miesto vykonania práce –

jo:hasDutyLocation, plat – *jo:hasSalary*, profesiu – *jo:offersPosition*, požiadavky na uchádzača – *jo:has Prerequisite* a ďalšie. Trieda *r:Region* opisuje regióny, napr. mestá – *r:SK_Bratislava*, trieda *c:ProfessionClassification* opisuje hierarchiu profesií. Celkovo ontológia obsahuje opis pracovných ponúk, regiónov, klasifikácií profesií, odvetví a pod.



Obrázok 11: Prehľad objektových vlastností triedy *jo:JobOffer* z ontológie pracovných ponúk.

Nevyhnutným predpokladom pre použitie metódy je možnosť definovať fazetovú klasifikáciu na základe doménovej ontológie. Fazety predstavujú ohraničenia, ktoré musia spĺňať výsledné inštancie informačného priestoru a možno ich definovať na základe štruktúry doménovej ontológie. Napr. pre triedu *jo:JobOffer* môže byť fazeta definovaná na základe jej vlastnosti *jo:hasDutyLocation*.

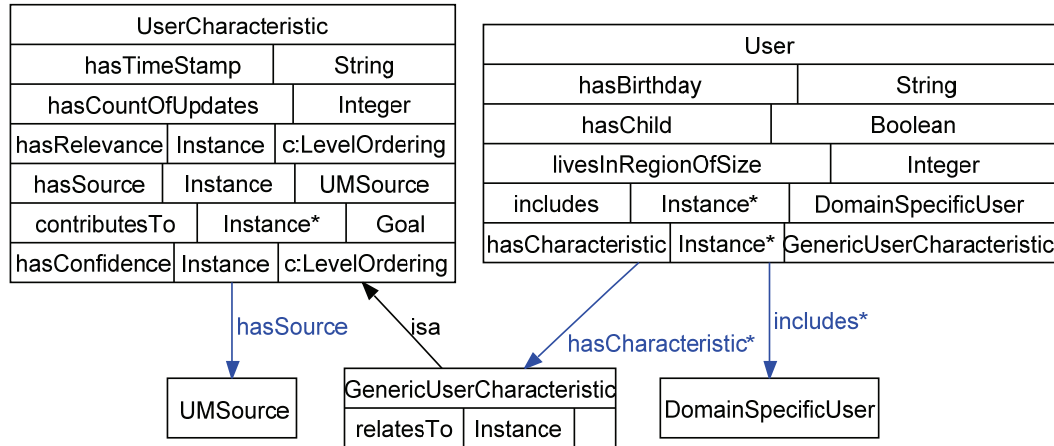
Doménová ontológia definuje koncepty relevantné pre danú aplikačnú oblasť a priradzuje im jednoznačné identifikátory URI. Tieto reprezentujú daný koncept a je možné ich použiť v ďalšom spracovaní tak pri zaznamenávaní akcií používateľa ako aj pri modelovaní používateľa a následnej personalizácii.

5.2.2 Ontológia používateľa

Charakteristiky používateľa nevyhnutné pre prispôbovanie opisuje ontológia používateľa. Vo svojej doménovo nezávislej časti (Obrázok 12) definuje používateľa – *User* a jeho základné vlastnosti, ktoré nie sú viazané na konkrétnu doménu. Ide napr. o meno, pohlavie, vek, záľuby, všeobecné preferencie alebo znalosti. V našom modeli ukladáme dátum narodenia – *hasBirthday*, skutočnosť, či má deti – *hasChild* a veľkosť regiónu, kde býva – *livesInRegionOfSize*.

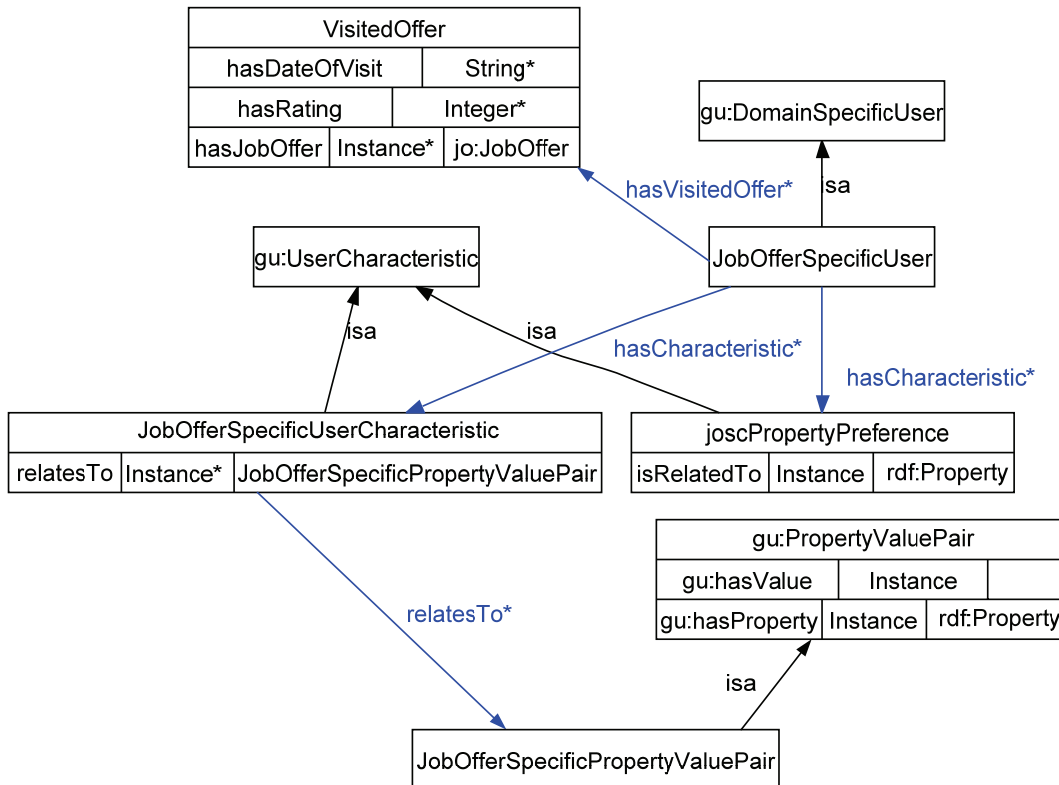
Pre každú charakteristiku používateľa model používateľa definuje:

- spoľahlivosť – *hasConfidence*,
- relevantnosť – *hasRelevance*,
- hodnotu – *relatesTo*,
- cieľ – *contributesTo*.



Obrázok 12: Prehľad doménovo nezávislých charakteristík používateľa.

Doménovo závislá časť ontológie používateľa je odvodená od doménovej ontológie a definuje charakteristiky a používateľa súvisiace s konceptmi z doménovej ontológie (Obrázok 13).



Obrázok 13: Prehľad doménových charakteristík používateľa v doméne pracovných ponúk.

Doménovo špecifický používateľ – *JobOfferSpecificUser* definuje vlastnosti pre dodatočné doménové charakteristiky:

- preferenciu atribútov ponuky – *joscPropertyPreference*,
- preferenciu konceptu z ontológie pre daný atribút – *JobOfferSpecificUserCharacteristic*,
- ohodnotenie navštívenej pracovnej ponuky – *VisitedOffer*.

Pre praktické použitie pri navigácii v informačnom priestore je dôležitejšia doménovo závislá časť ontológie používateľa, ktorá sa priamo odkazuje na koncepty doménovej ontológie pomocou URI. Definuje ciele jednotlivých používateľov v kontexte danej domény (napr. nájsť si prácu) ako aj ďalšie charakteristiky opisujúce relevanciu doménových konceptov k splneniu daného cieľa a spoľahlivosť týchto charakteristík. Príkladom charakteristiky je: vlastnosť *jo:hasSalary* je pre používateľa veľmi dôležitá (*hasRelevance* ~ 1,0) s strednou spoľahlivosťou (*hasConfidence* ~ 0,5).

5.2.3 Ontológia udalostí

Automatické odvodenie charakteristík používateľa si vyžaduje zaznamenávanie udalostí, ktoré nastávajú pri práci používateľa s adaptívnym fazetovým sémantickým prehliadačom spolu s ich sémantikou. Na reprezentáciu záznamov o akciách používateľa slúži ontológia udalostí, ktorá definuje typy udalostí, ktoré môžu pri práci používateľa so systémom nastať.

Štruktúra ontológie udalostí je daná doménou zaznamenávania udalostí a definuje používateľa, sedenie používateľa (angl. session), samotnú udalosť, jej atribúty. Ďalej definuje koncepty pre stav systému (zobrazenia) počas práce používateľa spolu s konkrétnymi zobrazenými položkami a ich vlastnosťami. Udalosti vyskytujúce sa v systéme a navigačný model adaptívneho fazetového sémantického prehliadača, resp. iných interakčných nástrojov, definujú typy udalostí, im zodpovedajúce typy atribútov ako aj interné stavy systému.

Dáta v ontológii udalostí predstavujú konkrétne záznamy o aktivite používateľa a spájajú vzniknuté udalosti s doménovými konceptmi pomocou URI. Príklad záznamu akcie používateľa je: používateľ s URI *User_1234* vykonal akciu *sl:Event_SelectRestriction* (voľba ohraničenia vo fazete) vo fazete *jo:hasDutyLocation* (miesto vykonávania práce) s cieľovým regiónom *r:SK_Bratislava*.

5.3 Rozšírený fazetový sémantický prehliadač

Navrhnutú metódu navigácie so sémantikou realizuje rozšírený fazetový sémantický prehliadač. Z hľadiska charakteristík fazetových prehliadačov je zaradený medzi prehliadače s podporou fazetovej navigácie a hlavných a vedľajších viacúrovňových faziet, pričom je možné ho doplniť aj o podporu pokročilého fazetového vyhľadávania.

5.3.1 Rozšírenia fazetového prehliadača

Základnú funkcionálnosť fazetového prehliadača (pozri časť 2.4) sme rozšírili o viaceré vlastnosti na zefektívnenie práce používateľa s informačným priestorom:

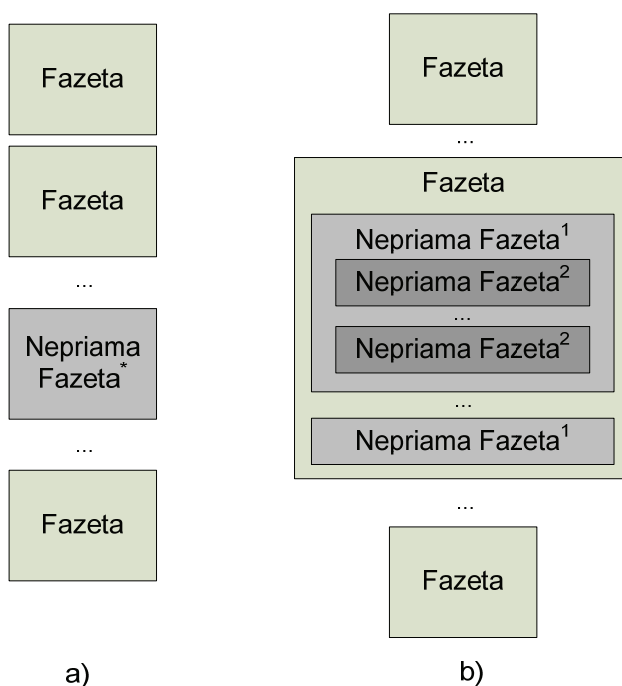
- *definovanie nepriamych ohraničení,*
- *dynamické generovanie faziet,*
- *pokročilý režim dopytovania.*

Definovanie nepriamych faziet

„Klasický“ fazetový prehliadač definuje fazety a následne ohraničenia nad inštanciami informačného priestoru pomocou priamych vlastností cieľových inštancií (napr. pracovných ponúk). V našej metóde umožňujeme definíciu faziet aj na základe nepriamych vlastností cieľových inštancií, teda vlastností inštancií, ktoré sú nimi v nejakom vzťahu.

Príklad ohraničenia pomocou nepriamej vlastnosti je: zobraz všetky ponuky práce, ktoré ponúka medzinárodná spoločnosť s aspoň 1000 zamestnancami. V uvedenom príklade cieľové inštalácie sú pracovné ponuky a ich priama vlastnosť je spoločnosť, ktorá ich ponúka. Použité nepriame ohraničenia – „medzinárodná“ a „s aspoň 1000 zamestnancami“ sa vzťahujú nie na ponuku práce, ale na spoločnosť, ktorá prácu ponúka.

Nepriame fazety môžeme neobmedzene vnárať, pokiaľ to dovoľuje doménová ontológia, čo umožňuje vizuálnu tvorbu aj veľmi komplexných dopytov. Navrhli sme dva spôsoby realizácie – priame zobrazenie nepriamych faziet spolu s priamymi fazetami (Obrázok 14a) alebo zobrazenie nepriamych faziet v nadradenej fazete (Obrázok 14b).



Obrázok 14: Realizácia nepriamych faziet (a) spolu s priamymi fazetami alebo (b) oddelene (rekurzívne) v rámci nadradenej fazety.

Dynamické generovanie faziet

Fazetový prehliadač v uzavretom informačnom priestore pracuje s vopred staticky definovanými fazetami. Pre prácu v otvorenom, dynamicky sa meniacom informačnom priestore s podporou adaptácie sme navrhli dynamické generovanie faziet z doménovej ontológie s využitím modelu používateľa.

Pomocou dynamického generovania faziet sprístupňujeme používateľovi fazety, ktoré práve potrebuje, aj keď neboli vopred staticky definované. Uvedená metóda umožňuje využitie aj tých vlastností inštancií z doménovej ontológie, ktoré pribudli až po nasadení systému alebo ich tvorcovia systému nepokladali za dôležité. Veľký význam generovanie faziet nadobúda v spojení s nepriamymi fazetami, pretože v praxi väčšinou nie je možné vopred staticky definovať všetky možné kombinácie priamych a nepriamych faziet.

Proces generovania faziet využíva metadáta o štruktúre doménovej ontológie a model používateľa, pričom pozostáva z týchto krokov:

- 1) Identifikácia nadradených faziet, resp. nepoužitých vlastností cieľových inštancií pre ktoré „má zmysel“ generovať fazety.
- 2) Identifikovanie oboru hodnôt nadradených faziet, resp. vlastností – určenie zodpovedajúcich tried, pričom pre literály sa fazety negenerujú.
- 3) Vygenerovanie množiny nepriamych faziet na základe vlastností tried a ich typov.
- 4) Výber spôsobu použitia vygenerovaných faziet na základe modelu používateľa.

Za vlastnosti vhodné pre dynamické generovanie pokladáme tie, ktoré majú vysokú relevanciu v modeli používateľa. Po vygenerovaní faziet určíme ich použitie v rozhraní fazetového prehliadača, kde môžu byť umiestnené rôznym spôsobom – na rovnakej úrovni ako nadradená fazeta alebo vnútri nadradenej fazety. Čím vyššia je relevancia vlastnosti, na základe ktorej bola nepriama fazeta vytvorená, tým vyššie v hierarchii nepriamych faziet by mala byť. Nepriame fazety s vysokou relevanciou umiestnime na najvyššiu úroveň, nepriame fazety s nižšou relevanciou umiestnime priamo do nadradenej fazety.

Dynamické generovanie faziet je schopné počas práce používateľa so systémom doplniť rozhranie prehliadača o nové fazety „je medzinárodná“, „počet zamestnancov“ a „umiestnenie v rebríčku TOP spoločností“, ak pre používateľa je dôležitá spoločnosť ponúkajúca prácu a príslušné údaje sa nachádzajú v doménovej ontológii. Súčasne umiestni fazetu „je medzinárodná“ na najvyššiu úroveň (používateľ preferuje medzinárodné spoločnosti) a ostatné fazety do nadradenej fazety pre „spoločnosť ponúkajúca prácu“.

Pokročilý režim dopytovania

Pokročilý režim dopytovania rozširuje základný režim o možnosť výberu viacerých ohraničení v jednej fazety s významom zjednotenia – OR (napr. New York alebo New Jersey), výber ohraničení na rôznych úrovniach a tiež použitie iných funkcií ako AND na kombináciu ohraničení vo viacerých fazetách. Pre podporu tvorby komplexnejších dopytov s vyššou vyjadrovacou silou sme navrhli použitie funkcie negácie NOT, resp. funkcií OR a zátvoriek.

Pokročilý režim dopytovania umožní používateľovi formulovať dopyt v tvare: zobraz všetky ponuky práce pre IT špecialistov v Bratislave, ale nie ponuky pre programátorov pokiaľ ponúkajú plat pod 50 tisíc mesačne.

Na zadanie konkrétnych funkcií pre dané ohraničenie sme navrhli použitie sady doplnkových odkazov pre jednotlivé funkcie OR, NOT a zátvorky v podobe ikoniek pri ohraničeniach vo fazetách, resp. pri aktuálne vybraných ohraničeniach.

5.3.2 Zaznamenávanie akcií používateľa so sémantikou

Pre potreby modelovania používateľa sme navrhli spôsob zaznamenávania akcií používateľa so sémantikou pomocou ontológie udalostí. Rozšírený fazetový sémantický prehliadač počas práce používateľa so systémom zaznamenáva jednotlivé udalosti, ktoré vyvolal používateľ.

Každý záznam obsahuje informáciu o používateľovi, ktorý udalosť vyvolal a previazanie na konkrétne sedenie používateľa, ktorý môže byť prihlásený viackrát súčasne. Pre každú udalosť sa ukladajú atribúty, ktoré bližšie opisujú kontext udalosti, resp. jej cieľ. Napr. pre výber ohraničenia vo fazete sa zaznamená fazeta *jo:hasDutyLocation* ale aj výsledné ohraničenie *r:SK_Bratislava*.

Súčasne s parametrami udalosti sa zaznamená aj stav zobrazenia pred a po nastaní udalosti, ktorý zachytáva dôvody, ktoré používateľa viedli k vykonaniu danej akcie. Záznam stavu zobrazenia obsahuje informácie o zobrazených konceptoch – URI zobrazených faziet, inštancií a atribútov. Tento záznam slúži pre odvodenie charakteristík používateľa pomocou analýzy implicitnej spätnej väzby (Barla, 2006a), pričom predpokladáme, že používateľ sa rozhoduje na základe informácií zobrazených v rozhraní prehliadača.

5.3.3 Prispôsobovania navigácie vo fazetovom prehliadači

Prispôsobovanie navigácie realizujeme pomocou adaptácie faziet a ich hodnôt – ohraničení na základe modelu používateľa, ktorý opisuje relevantné charakteristiky používateľa (pozri prílohu C). Samotné prispôsobovanie realizujeme pre konkrétny cieľ – nájsť vhodnú ponuku práce, s ktorým môže byť v modeli používateľa asociovaná jedna alebo viacero charakteristík. Pre každú charakteristiku model používateľa definuje *relevance* – hodnotu relevancie k splneniu cieľa a *confidence* – mieru spoľahlivosti, s ktorou bola daná charakteristika odvodená. Obe hodnoty – relevancia a spoľahlivosť sú z intervalu $<0, 1>$.

Navrhli sme viaceré spôsoby adaptácie navigácie vo fazetovom prehliadači:

- Výber a usporiadanie vhodných faziet z definovanej množiny faziet.
- Výber a usporiadanie prezentovaných hodnôt vo fazetách.
- Anotácia vybraných faziet a prezentovaných hodnôt faziet.
- Anotácia výsledkov vyhľadávania.

Výber a usporiadanie relevantných faziet z definovanej množiny faziet

Pre výber a usporiadanie faziet pre konkrétneho používateľa využívame údaje o relevancii jednotlivých faziet z modelu používateľa. Dodatočne tiež využívame údaje o relevancii faziet z modelov ostatných používateľov na prispôbenie pomocou techník kolaboratívneho filtrovania. Pre každého používateľa U a každú fazetu X definujeme tieto typy relevancie:

- *Lokálna relevancia* $L_U(X)$ je daná relevanciou $relevance_U(X)$ danej fazety z modelu používateľa. (1).

$$L_U(X) = relevance_U(X) \quad (1)$$

- *Križová relevancia* $C_U(X)$ je priemer zo súčiny hodnoty vzdialenosti používateľa U od ostatných používateľov V a hodnoty lokálnej relevancie fazety X pre používateľa V (2). Vzdialenosť určíme z miery podobnosti konceptov U, V – modelov používateľa pomocou externého nástroja pre porovnávanie konceptov. Alternatívne za vzdialenosť možno považovať sumu rozdielov lokálnych relevancií pre všetky fazety X daných používateľov (3). Ak sa niektorá nenachádza v modeli používateľa použije sa namiesto nej predvolená hodnota 0.

$$C_U(X) = \frac{\sum_V^{users} (dist(U, V) * L_V(X))}{|users|}, u \neq v \quad (2)$$

$$dist(U, V) = \sum_X^{facets} (L_U(X) - L_V(X))^2 \quad (3)$$

- *Globálna relevancia* $G(X)$ predstavuje celkovú relevanciu fazety v systéme v závislosti od jej používania jednotlivými používateľmi. Globálna relevancia fazety sa inicializuje v čase jej pridania do systému na predvolenú hodnotu 1,0. Po určení lokálnej relevancie fazety X pre daný minimálny počet používateľov určujeme globálnu relevanciu fazety ako priemer jej lokálnych relevancií pre každého používateľa V (4).

$$G(X) = \frac{\sum_V^{users} L_V(X)}{|users|} \quad (4)$$

- *Dočasná relevancia* $T_U(X)$ je odvodená od použitia fazety X počas aktuálneho sedenia používateľa. Určíme ju ako podiel kliknutí používateľa do fazety X a celkového počtu kliknutí používateľa do ľubovoľnej fazety (5).

$$T_U(X) = \frac{Clicks(X)}{TotalClicks} \quad (5)$$

- *Statická relevancia* $S_U(X)$ je odvodená z ontológie používateľa na základe profilov používateľov určených externými nástrojmi. Určíme ju na základe lokálnej, križovej a globálnej relevancie (6).

$$S_U(X) = \frac{L_U(X) * confidence_U(X) + (C_U(X) + G_U(X)) * (1 - confidence_U(X))}{2} \quad (6)$$

- *Dynamická relevancia* $D_U(X)$ zohľadňuje statickú relevanciu $S_U(X)$ fazety určenú z modelov používateľa a dočasnú relevanciu $T_U(X)$ danú použitím faziet v rámci aktívneho sedenia (7).

$$D_U(X) = S_U(X) + T_U(X) \quad (7)$$

Pre prezentáciu používateľovi U vyberieme k faziet s najvyššou dynamickou relevanciou $D_U(X)$, pričom relevanciu poslednej použitej fazety umelo zvýšime o predvolenú hodnotu 0,5. Určenie vhodného k bolo predmetom experimentovania, pričom z experimentov sme určili hodnotu $k = 1$ pre adaptívne fazety bez odporúčania a $k = 2$, resp. 1 až 3 pre adaptívne fazety s odporúčaním.

Výber a usporiadanie prezentovaných hodnôt vo fazetách

Praktické skúsenosti ukazujú, že výber a usporiadanie hodnôt faziet má veľký vplyv na použiteľnosť fazetového prehliadača. Používateľ by mal mať možnosť výberu deterministického usporiadania hodnôt vo fazetách, aby mohol efektívne vyhľadávať želané hodnoty. Ak napr. hľadáme štát New York vo fazete usporiadanej abecedne, vieme ako a kde ho mám hľadať (približne „v strede“ fazety). Ak však usporiadame položky napr. podľa počtu inštancií spĺňajúcich dané ohraňenie, musíme lineárne prezrieť celú fazetu, keďže vopred nevieme, koľko inštancií vyhovuje.

Z uvedeného dôvodu sme navrhli predvolené usporiadanie všetkých hodnôt faziet v ich prirodzenom (deterministickom) poradí. Numerické hodnoty usporiadame podľa veľkosti, dátumy podľa časovej následnosti, pre texty podľa abecedy, resp. lexikograficky. Predpokladáme aj možnosť používateľského nastavenia usporiadania (vzostupne/zostupne), pričom v takom prípade sa použije posledné nastavenie pre danú fazetu, resp. ak ešte nebola použitá daným používateľom najčastejšie nastavenie ostatných používateľov (predvolene vzostupne).

Navrhli sme tieto spôsoby usporiadania hodnôt vo fazetách:

- *prirodzené usporiadanie*,
- usporiadanie podľa *počtu inštancií*,
- usporiadanie podľa *relevancie* hodnoty H určenej z modelu používateľa (8).

$$relevancia(H) = relevance_U(H) * confidence_U(H) \quad (8)$$

Z usporiadanej množiny ohraničení následne prezentujeme vybranú časť v závislosti od typu usporiadania a relevancie (poradia) fazety. Pre prirodzené usporiadanie zobrazujeme všetky hodnoty, pre usporiadanie podľa počtu inštancií, resp. relevancie zobrazujeme prvých maximálne k hodnôt. Hodnota k závisí od poradia fazety, pričom pre prvú až tretiu fazetu je (predvolene) $k = 10$, pre štvrtú a vyššiu fazety $k = 5$. Alternatívne môže hodnota k byť adaptívna pre rôznych používateľov v závislosti od ich spôsobu práce a úspešnosti jej odhadu. Odporúčanie hodnôt realizujeme duplicitným výberom maximálne troch najrelevantnejších hodnôt na začiatku fazety.

Anotácia vybraných hodnôt faziet

Pomocou hodnôt vo fazetách sprístupňujeme používateľovi dodatočné informácie, ktoré mu pomáhajú pri výbere ďalšieho smeru navigácie:

- Vyjadrenie relatívnej početnosti $P_X(H)$ inštancií danej hodnoty H fazety X pomocou veľkosti písma (9), pričom hodnotu $P_X(H)$ zobrazíme do použitého rozsahu veľkostí písma.

$$P_X(H) = * \frac{\text{Count}_H(X)}{\text{Count}(X)} \quad (9)$$

- Vyjadrenie relevancie danej hodnoty fazety pomocou farby pozadia (8), pričom získanú hodnotu relevancie zobrazíme do použitého farebného priestoru jasných a tmavých farieb.

Anotácie výsledkov vyhľadávania

Pre zlepšenie pochopenia výsledkov vyhľadávania používateľom sme navrhli použitie anotácie jednotlivých inštancií pomocou farby pozadia, emotikonov a semaforov. Konkrétna realizácia anotácie je však značne doménovo závislá. Pre doménu pracovných ponúk sme navrhli zobrazenie týchto dodatočných informácií:

- *Mieru do akej používateľ spĺňa požiadavky kladené zadávateľom ponuky na uchádzačov* vo forme farby pozadia v rozsahu od červených po zelené odtiene (nesplňa/spĺňa podmienky).
- *Mieru do akej ponuka spĺňa kritériá používateľa* vo forme emotikonu.
- *Celkovú vhodnosť ponuky pre používateľa* vo forme svetiel semaforu, po zohľadnení miery splnenia požiadaviek používateľom, splnenia kritérií používateľa a prípadných dodatočných doménových znalostí.

Odvodenie uvedených vlastností (miery do akej sú splnené) nie je triviálne a preto je nutné na ich určenie využiť externý nástroj na určenie podobnosti konceptov, resp. vyhodnotenia doménových znalostí. Následne výstup nástroja – mieru podobnosti/vhodnosti normujeme a podľa jej výslednej hodnoty vyberieme vhodný spôsob anotácie. Napr. pre mieru $m = 1,0$ to bude v prvom prípade zelená farba, v druhom „☺“, v treťom zelené svetlo na semafore.

5.3.4 Prispôsobovanie práce s výsledným dokumentom

Pre rýchle pochopenie podrobných informácií o konkrétnej inštancii informačného priestoru – výsledku vyhľadávania sme navrhli pokročilý spôsob ich zobrazenia:

- Zobrazenie atribútov inštalácie pomocou tabuľky a ich usporiadanie v prirodzenom usporiadaní, lexikograficky, resp. podľa relevancie.
- Voliteľné zobrazenie jednotlivých atribútov inštalácie rekurzívne pre objektové vlastnosti inštalácií, t.j. rekurzívne vloženie tabuľky s atribútmi naviazanej inštalácie.
- Generovanie odkazov pre vybrané atribúty inštalácií, tak aby bolo možné prejsť na zobrazenie informácií o nich. Napr. pracovná ponuka môže obsahovať odkazy na podobné pracovné ponuky, resp. vedecká publikácia môže obsahovať odkazy na použitú literatúru (vedecké publikácie) alebo ďalšie publikácie autorov, ktoré sú príbuzné.
- Generovanie odkazov pre pridanie vybraného atribútu inštalácie ako ohraničenia v príslušnej fazete s návratom na prehľad výsledkov vyhľadávania.
- Zvýraznenie vybraných informácií o inštaláciách (napr. informácia o plate) daných výberom v „alternatívnej sade faziet“ pomocou farby pozadia. Zvýraznenie informácií je obzvlášť užitočné, ak sú výsledky vyhľadávania dlhšie, resp. obsahujú súvislé bloky textu. Zvýraznenie realizujeme priamo na základe dát z ontológie (ak obsahuje anotované zdrojové texty) alebo pomocou externého nástroja.

5.4 Návrh grafického používateľského rozhrania

Pri návrhu používateľského rozhrania sme vychádzali z analýzy viacerých existujúcich fazetových prehliadačov (Instone, 2004). Obrázok 15 obsahuje predbežný návrh grafického rozhrania rozšíreného fazetového sémantického prehliadača pre doménu pracovných príležitostí. Vytvorený návrh predstavuje len návrh koncepcie rozhrania, pričom samotný grafický návrh (dizajn) by mal spraviť profesionál v oblasti návrhu používateľských rozhraní.

Jednoduchý Pokročilý A AND OR NOT XOR		Región: Všetky > USA (1 100) C Odvetvie: Všetky (1 100) Plat: Všetky (1 100)			
		<História> D			
 Región B	Porovnaj	 Názov  Región  Plat  Rating	Stručný Detaily	Plat F Prac. náplň	
 Seattle (50)  New York (550)  New Jersey (300)  Washington (100)  Kalifornia (150)	☐ E	 C++ Programátor \$55,000 New Jersey G  Softvérová firma hľadá zamestnanca na post C++ programátora. Predchádzajúce skúsenosti v ... 			
	☐	 Projektový manažér \$85,000 New York  Hľadáme zamestnancov na pozície projektových manažérov pre softvérové projekty stredného rozsahu. Nutné výborné komunikačné schopnosti... 			
 Odvetvie	☐	 Vodič \$25,000 Seattle  Taxi služba mesta Seattle hľadá uchádzačov o pozíciu vodiča. Skúsenosti v oblasti v dĺžke 5+ rokov sú výhodou. Ponúkame základný plat \$25,000 za rok + odmeny v závislosti od výkonov. Pohyblivý pracovný čas, 1 deň plat. voľna mesačne. 			
 Plat	☐				
 >\$150,000/rok (3)  ≤\$150,000/rok (50)  ≤\$100,000/rok (350)  ≤\$50,000/rok (250)  ≤\$20,000/rok (50)					
		Nájdenných výsledkov: 1 100 Na stranu: 3 10 25 50 H Strana: 1 2 3 4 5 6 7 Ďalšia >			

Obrázok 15: Predbežný návrh grafického rozhrania rozšíreného fazetového prehliadača.

Celú pracovná plocha pozostáva z troch hlavných častí (Obrázok 15):

- *Fazety* pre prácu s ohraničeniami nad inštanciami doménovej ontológie (oblasti A, B).
 - Oblasť A umožňuje používateľovi voľbu režimu na definovanie ohraničení v jednotlivých fazetách (jednoduchý režim / pokročilý režim) a tiež prácu s pokročilým režimom, napr. výber kombinačných funkcií faziet.

- Oblasť B obsahuje zoznam faziet, ktoré je možné použiť na definovanie ohraničení pre dopyt do doménovej ontológie. Pre každú fazetu je uvedený jej názov, zoznam možných ohraničení a im zodpovedajúci počet inštancií z doménovej ontológie. Odporúčané ohraničenia vo fazete sú anotované zelenou farbou pozadia. Jednotlivé fazety je možné aktivovať/deaktivovať pomocou tlačidla pred názvom fazety.
- *Aktuálne ohraničenia* (oblasť C) obsahuje aktuálne ohraničenia z jednotlivých faziet – aktuálny dopyt do doménovej ontológie a umožňuje používateľovi odstránenie niektorých jeho častí.
- *História prezerania stránok* (oblasť D), resp. história dopytov do ontológie. Predpokladáme tiež, že prípadné ďalšie prvky, ktoré identifikujeme ako podstatné pre prácu s informáciami budú umiestnené v tejto časti. Pre všeobecný nástroj určený na prezeranie ontológií to je napr. možnosť zvoliť si triedu, s ktorej inštanciami chceme pracovať, alebo rozhranie externého nástroja pre vyhľadávanie/usporiadanie.
- *Výsledky vyhľadávania* pomocou ohraničení a nástroje na ich ďalšie spracovanie (oblasti E, F, G).
 - Oblasť E umožňuje používateľovi vyznačiť viaceré z výsledkov a zobrazit' ich porovnanie napr. v podobe prehľadnej tabuľky ich vlastností.
 - Oblasť F obsahuje nástroje na ďalšie spracovanie výsledkov vyhľadávania. Zľava doprava postupne obsahuje nástroje na vzostupné/zostupné *usporiadanie výsledkov* podľa rôznych kritérií (názov, plat, región), voľbu *aktuálneho pohľadu* na výsledky (jednoduchý/detaily) a *zvýraznenie informácií* (plat, pracovná náplň).
 - Oblasť G obsahuje prehľad výsledkov vyhľadávania pomocou ohraničení – adaptívne anotovaný zoznam inštancií z doménovej ontológie (☺/☹, farba pozadia, „semafor“). Spôsob zobrazenia údajov v tejto oblasti je ovplyvnený nastaveniami v oblasti E. Napr. podľa typu zvoleného pohľadu (oblasť E – stručný, resp. detailný) je možné pre každú inštanciu zobrazit' buď len sumárne údaje (názov pozície, plat, región) , alebo aj dodatočné údaje v podobe súvislého textu. V uvedenom príklade uchádzač spĺňa podmienky na pozíciu C++ programátor (zelená farba), spĺňa väčšinu podmienok na pozíciu vodič (žlto-zelená farba) a spĺňa len časť podmienok na pozíciu projektový manažér (žltá farba). Dodatočné zobrazenie väčšieho množstva informácií (textu) je sprístupnené pomocou šípok dole/hore.
- Oblasť H zobrazuje počet nájdených inštancií a umožňuje používateľovi navigáciu medzi viacerými stránkami výsledkov, ako aj zmenu počtu súčasne zobrazovaných položiek.

6 Softvérové nástroje realizujúce metódu

Overenie metódy personalizovanej navigácie sme realizovali pomocou návrhu dvoch softvérových nástrojov, ktoré ju realizujú a ich následnou implementáciou a overením v rámci projektov NAZOU (Návrat et al., 2006) a MAPEKUS (Bieliková a kol., 2005a), kde sú zaradené do vrstvy personalizovanej prezentácie.

Nástroj *Factic* – *Fazetový sémantický prehliadač* (Faceted Semantic Browser) realizuje personalizovanú fazetovú navigáciu v informačnom priestore reprezentovanom ontológiou. Nástroj *SemanticLog* – *Sémantické logovanie* realizuje záznam akcií používateľa so sémantikou pomocou ontológie udalostí. Príloha D obsahuje technickú dokumentáciu k nástrojom, postup ich inštalácie a konfigurácie, ako aj spôsob ich použitia.

6.1 Kontext a architektúra riešenia

Z dôvodu riešenia nástrojov v kontexte projektov NAZOU a MAPEKUS vyplynulo viacero ohraničení na implementačné prostredie, nástroje a technológie, ktoré sme použili:

- Implementácia v jazyku Java verzie 5.
- Použitie prezentačného rámca Apache Cocoon⁶ ako nástroja pre prezentáciu webového rozhrania a zabezpečenie interakcie s používateľom.
- Použitie ontologického úložiska Sesame⁷ ako databázového stroja pre ontologické dáta.
- Použitie relačného úložiska MySQL⁸ ako databázového stroja pre relačné dáta.
- Použitie servletu Apache Axis⁹ ako kontajnera na nasadenie webových služieb.
- Použitie servlet kontajnera Apache Tomcat¹⁰, v ktorom je nasadený prezentačný rámec Cocoon, ontologické úložisko Sesame a Axis.
- Použitie odporúčaného integrovaného vývojového prostredia Eclipse¹¹.
- Použitie nástroja Subversion¹² na správu verzií zdrojových kódov (nasadený na vývojovom serveri projektu NAZOU).

Obrázok 16 znázorňuje kontext nasadenia nástrojov *Factic* a *SemanticLog*. Používateľ komunikuje s nástrojom *Factic* prostredníctvom webového prehliadača, ktorým prístupuje k prezentačnému rámcu Apache Cocoon a v rámci neho k nástroju *Factic*. Tento počas prevádzky generuje stránky podľa dopytov používateľa. Údaje potrebné na generovanie stránok načítava z ontologického úložiska Sesame, ktoré obsahuje doménovú ontológiu a ontológiu používateľa. Cocoon aj Sesame sú nasadené v kontajneri Tomcat. Pomocou API nástroja *SemanticLog* fazetový prehliadač vytvára záznam o akciách používateľa. Samotný

⁶ The Apache Cocoon Project, <http://cocoon.apache.org/> (10.12.2006)

⁷ openRDF.org...home of Sesame, <http://www.openrdf.org/> (10.12.2006)

⁸ MySQL, <http://www.mysql.com/> (10.12.2006)

⁹ Apache Axis, <http://ws.apache.org/axis/> (10.12.2006)

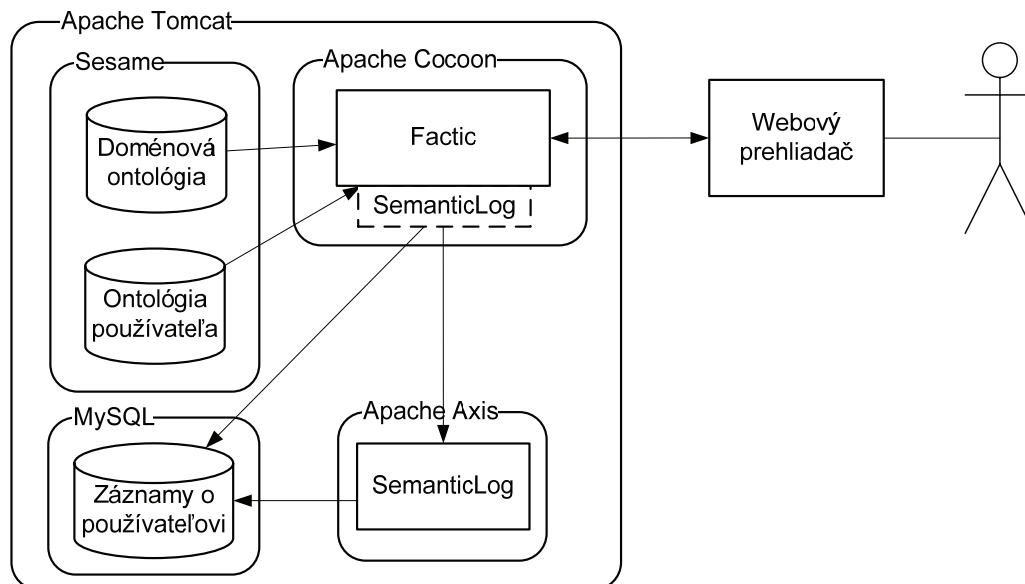
¹⁰ Apache Tomcat, <http://tomcat.apache.org/> (10.12.2006)

¹¹ Eclipse open source IDE, <http://www.eclipse.org/> (10.12.2006)

¹² Tigris.org: Open Source Software Engineering Tools – Subversion, <http://subversion.tigris.org/> (10.12.2006)

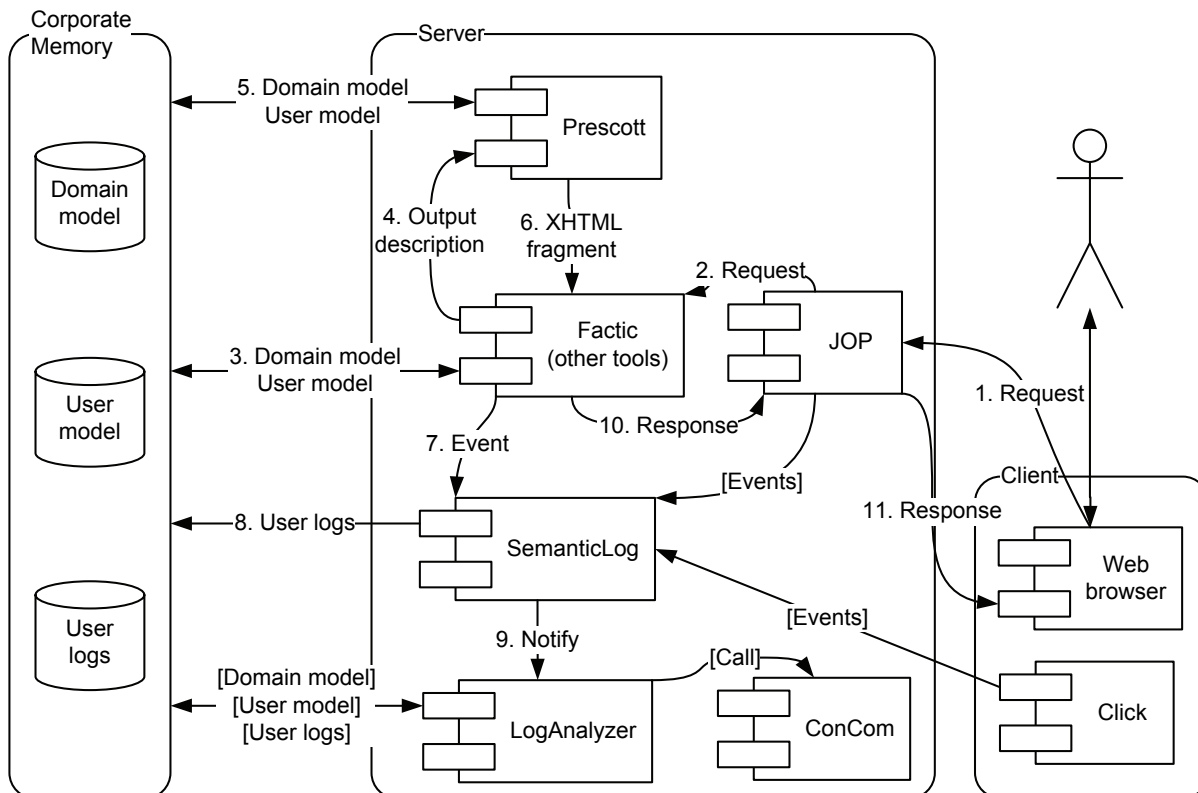
SemanticLog je nasadený ako webová služba v kontajneri Apache Axis a používateľovi ukladá do MySQL databázy, ktorú využívajú aj ďalšie nástroje v rámci projektu.

Zdroj dát pre *Factic* je ontologické úložisko Sesame, ktoré obsahuje doménovú ontológiu a ontológiu používateľa, obe zapísané v jazyku OWL.



Obrázok 16: Kontext nasadenia nástrojov *Factic* a *SemanticLog*.

Nástroje *Factic* a *SemanticLog* sú integrované s ďalšími nástrojmi v oboch uvedených projektoch (Tvarožek et al., 2007). Obrázok 17 znázorňuje architektúru vrstvy personalizovanej prezentácie v projekte NAZOU spolu s prepojeniami jednotlivých nástrojov (Tvarožek et al., 2006e). Prichádzajúce požiadavky klienta presmeruje portál pracovných príležitostí *JOP* (Filkmorn et al., 2006) príslušnému nástroju, v našom prípade nástroju *Factic*. Ten pri spracovaní požiadavky pristupuje ku korporatívnej pamäti, ktorá obsahuje doménovú ontológiu a ontológiu používateľa. Následne voliteľne spracuje časť výstupu pomocou prezentačného rámca *Prescott* (Bieliková et al., 2005b) a zaznamená akciu používateľa pomocou webovej služby *SemanticLog* (Barla, Tvarožek, 2006b). Tento vytvorí záznam v databáze akcií používateľov a oznámi zmenu nástroju na modelovanie používateľa *LogAnalyzer* (Barla, 2006a). Výsledná webová stránka vygenerovaná nástrojom *Factic* je na záver prostredníctvom portálu *JOP* odoslaná klientovi. *SemanticLog* môže byť voliteľne zavolaný aj nástrojom *Click* na strane klienta. Počas procesu odvodzovania charakteristík používateľa nástroj *LogAnalyzer* môže voliteľne využívať služby nástroja *ConCom* (Andrejko et al., 2006b) na porovnanie konceptov z ontológie.

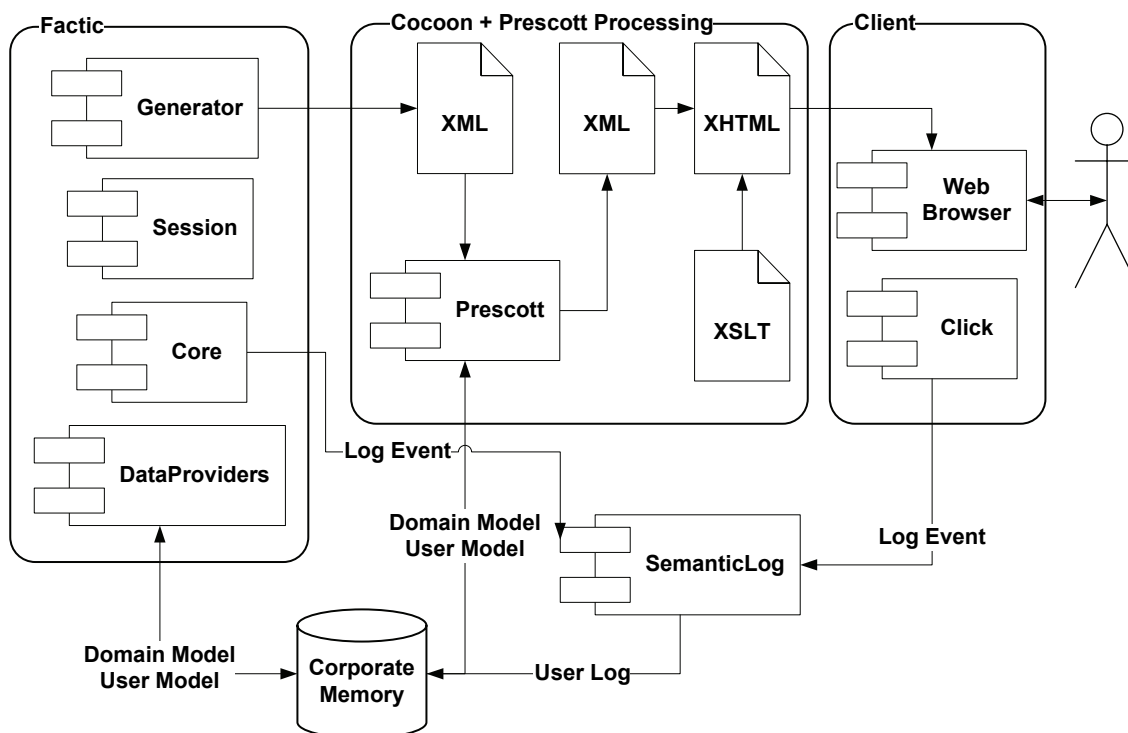


Obrázok 17: Architektúra vrstvy personalizovanej prezentácie v projekte NAZOU.

6.2 Factic – návrh a implementácia

Nástroj *Factic* má modulárny návrh pozostávajúci zo štyroch základných častí (pozri Obrázok 18, vľavo). Nástroj je navrhnutý ako generátor pre prezentačný server Apache Cocoon, pričom príslušnú funkcionality generátora realizuje komponent *Generator*. Tento počas prevádzky inicializuje komponent *Session*, ktorý zaobľuje funkcionality fazetového prehliadača pre jedno používateľské sedenie. Komponent *Session* udržiava stav sedenia používateľa a riadi jadro fazetového prehliadača – komponent *Core*, ktorý zabezpečuje hlavnú funkcionality – prácu s fazetami, ohraňovaniami a inštanciami. Na prístup k dátovým úložiskám (Sesame, resp. MySQL) slúžia komponenty *DataProviders*, ktoré implementujú rozhranie *IDataProvider* a zakrývajú tak prácu s rôznymi typmi úložisk.

V strednej časti obrázka je zobrazený výstup z generátora predstavujúci XML súbor s opisom rozhrania prehliadača, ktorý je následne spracovaný v rámci servera Cocoon pomocou XSL šablón. Tieto z opisu rozhrania vytvoria XHTML, ktoré je následne ako výstupná stránka zaslané webovému prehliadaču klienta. Návrh počíta s prepojením XML výstupu z generátora s prezentačným rámcom *Prescott* v rámci servera Cocoon (Tvarožek, 2006d). Na zaznamenanie akcií používateľa komponent *Core* využíva API webovej služby nástroja *SemanticLog*, zobrazeného v pravej dolnej časti obrázka.



Obrázok 18: Prehľad architektúry a integrácie adaptívneho fazetového prehliadača.

Činnosť nástroja možno opísať nasledovnou postupnosťou krokov:

- 1) Inicializácia nástroja a načítanie opisu faziet z konfiguračného súboru.
- 2) Čakanie na vstup od používateľa.
 - a) Prijatie požiadavky od používateľa. Ak je požiadavka na ukončenie činnosti, pokračuje sa krokom 3, inak krokom b.
 - b) Obnovenie aktuálneho počtu inštancií pre jednotlivé ohraničenia v rámci faziet.
 - c) Formulácia a vykonanie dopytu do ontologického úložiska, spracovanie získaných údajov podľa prijatej požiadavky, vytvorenie záznamu v logu.
 - d) Vygenerovanie výstupu vo formáte XML podľa prijatej požiadavky.
 - e) Transformácia XML výstupu pomocou XSLT do XHTML webovej stránky.
- 3) Deinicializácia nástroja, uloženie údajov a ukončenie činnosti.

Realizované časti navrhnutej metódy

Pomocou nástroja *Factic* sme vzhľadom na rozsah projektu overili vybrané časti navrhnutej metódy:

- Rozšírenie fazetového prehliadača o podporu nepriamych faziet.
- Zaznamenávanie akcií používateľa so sémantikou.
- Prispôsobovanie navigácie vo fazetovom prehliadači pomocou adaptácie a anotácie faziet a ohraničení vo fazetách.

Vytvorená implementácia nástroja *Factic* realizuje nasledovnú funkcionálnosť:

- Zobrazenie inštancií z doménovej ontológie (pracovných ponúk, vedeckých publikácií).
- Definíciu a vykonanie dopytov nad niekoľkými dimenziami informačného priestoru pomocou vopred definovanej množiny priamych i nepriamych faziet.
- Fazety skonštruované na základe hierarchických klasifikácií, číselníkov aj intervalov.
- Usporiadanie výsledkov podľa zvoleného atribútu.
- Navigácia vo výsledkoch a výber počtu súčasne zobrazených výsledkov.
- Vykonanie vybraných akcií so zvolenými inštanciami prepojením s portálom *JOP*, (napr. editovanie, zmazanie).
- Zobrazenie detailných informácií o vybranej inštancii s rekurzívnym výpisom atribútov.
- Zaznamenanie vybraných akcií používateľa so sémantikou (napr. výber ohraničenia, zobrazenie detailov, prechod na ďalšiu stránku).
- Adaptácia – výber a usporiadanie faziet na základe ich relevancie určenej z modelu používateľa a klikania používateľa v aktívnom sedení.
- Anotácia a jednoduché usporiadanie hodnôt vo fazetách.

Duty location

Arizona (3)

California (19)

Colorado (3)

Connecticut (3)

D.C. (2)

Florida (3)

Georgia (2)

Illinois (3)

Maryland (4)

Massachusetts (2)

Michigan (2)

Missouri (1)

New Hampshire (1)

New Jersey (10)

New York (11)

North Carolina (2)

Oregon (2)

Pennsylvania (12)

South Carolina (2)

Tennessee (2)

Texas (7)

Virginia (1)

Washington (2)

Active in sector/industry

Engineering and Architecture (1)

Health Care and Medical Activities (1)

Information technology service activities (4)

Job term

Full-time (16)

Contract Type

Contract (12)

Temporary (5)

Duty location: All > World > America > North America > United States (99)

Active in sector/industry: All (99)

Job term: All (99)

Contract Type: All (99)

Offered benefits: All (99)

Travelling involved: All (99)

Hours/week: All (99)

Acquisition date: All (99)

Start date: All (99)

Offered management level: All (99)

Offered position: All (99)

Total Matches: 99 on 7 pages.

Page: 1 | 2 | 3 | 4 | 5 | 6 | 7 | Next >

Sort by: Name | Salary | Organization | Region

Item per page: 10 | 15 | 25 | 50 | 100

#	Name	Salary	Organization	Region	Actions
1	Software Programmer; VC++ Makefile generation	30.0	Manpower Professional	Phoenix	
2	Programmer; VC++ Makefile generation	30.0		Phoenix	
3	Programmer/Analyst, Corporate Oracle Applications	54.03		Tempe	
4	6181 - Systems Programmer	85000.0	Genesis10	Pasadena	
5	C# / ASP.NET Programmers	35000.0	At-Tech	California	
6	Manpower Professional,San Diego, CA US	29.0	Manpower Professional	California	
7	At-Tech, Orange, CA US	25000.0	At-Tech	California	
8	Systems Programmer	50.0	Genesis10	Pasadena	
9	COBOL BASIC Programmers VMS DEC Alpha	35.0	Indotronix International Corp.	Glendale	
10	Programmer Analyst II	40.0	Genesis10	Berkeley	
11	Senior Programmer Analyst	50.0	Genesis10	Berkeley	
12	Oracle 7-9i Programmer (PL/SQL, DataStage, AIX)		Strategi	Cypress	
13	Programmer	80000.0	Harte_Enterprise	Glendale	
14	Junior Programmer	35000.0		Glendale	
15	Programmer - C++ & .net	45000.0	Harte Enterprise	Glendale	

Obrázok 19: Ukážka používateľského rozhrania nástroja *Factic*.

Obrázok 19 znázorňuje ukážku výsledného používateľského rozhrania nástroja *Factic*. Fazety usporiadané podľa relevancie sú zobrazené vľavo (miesto vykonávania práce, odvetvie, úväzok, typ zmluvy). Hodnoty vo fazetách sú anotované textovo (počet inštancií vyhovujúcich ohraničeniu) a pomocou veľkosti fontu text (relatívna početnosť inštancií pre dané ohraničenie). Aktuálny výber ohraničení vo fazetách je zobrazený v hornej časti obrázka (ponuky v USA).

Výsledky vyhľadávania v podobe vyhovujúcich inštancií pracovných ponúk sú zobrazené vpravo, pričom je možné ich usporiadať na základe vybraného atribútu, resp. navigovať vo viacerých stranách výsledkov.

Obrázok 20 znázorňuje ukážku prezerania podrobných informácií o pracovnej ponuke so zobrazením hodnôt jej atribútov. V poli „Prerequisites“ a „Offers salary“ vidíme rekurzívny výpis atribútov inštancie pomocou tabuľky v tabuľke pre objektové vlastnosti pracovnej ponuky.

[Back](#)

User:	unknown								
Apply information:	Text: Manpower Professional, San Diego, CA US								
Active:	true								
Prerequisites:		<table border="1"> <tr><td>XML</td></tr> <tr><td>Oracle ERP</td></tr> <tr><td>5 to 8 years of software development experience</td></tr> <tr><td>PL/SQL</td></tr> <tr><td>ODBC</td></tr> <tr><td>2 plus years of experience in TIBCO Business Works and WorkFlow development</td></tr> </table>		XML	Oracle ERP	5 to 8 years of software development experience	PL/SQL	ODBC	2 plus years of experience in TIBCO Business Works and WorkFlow development
	XML								
Oracle ERP									
5 to 8 years of software development experience									
PL/SQL									
ODBC									
2 plus years of experience in TIBCO Business Works and WorkFlow development									
	Text:	Position available to new or recent college graduates? NO Bachelor of Science degree in a technical field (i.e. Computer Science, Information Technology, Engineering or related) required							
Offers salary:	Text:	Salary: \$29/HOURLY To \$36/HOURLY							
	Base:	29.0							
	Maximum:	36.0							
Duty location:	rdfs:label:	California							
Offered by:	Name:	Manpower Professional							
Date:	20.10.2006								
Name:	Manpower Professional, San Diego, CA US								
jo:hasSource:									

Obrázok 20: Príklad prezentácie podrobných informácií o vybranej pracovnej ponuke.

6.3 SemanticLog – návrh a implementácia

Pomocou nástroja SemanticLog realizujeme záznam akcií používateľa so sémantikou. Z dôvodov výkonnosti je nástroj navrhnutý ako hybridná webová služba. Základné zaznamenanie udalosti je riešené priamo pomocou rozhrania webovej služby, zaznamenanie podrobných informácií o stave systému (zobrazenia) je realizované pomocou priameho volania API nástroja SemanticLog.

Na uloženie údajov o akciách používateľa používame jednoduchú ontológiu udalostí (Príloha C) uloženú v MySQL databáze (Obrázok 16, s. 42) z dôvodov výkonnosti, ďalšieho spracovania nástrojmi pre modelovanie používateľa a samotnej povahy údajov – prúdu záznamov dát. Prácu s databázou realizujeme pomocou objektovo-relačného mapovača Hibernate¹³, ktorý transparentne zabezpečuje perzistenciu objektov JavaBean.

Počas svojej činnosti *SemanticLog* čaká na požiadavky iných nástrojov o zaznamenanie udalostí. Po príchode požiadavky vytvorí z údajov o udalosti záznam, ktorý priradí konkrétnemu sedeniu prihláseného používateľa. Súčasne naviaže záznam na predchádzajúci stav sedenia, zaznamená ho do SQL databázy (Obrázok 17, s. 43) a notifikuje nástroj *LogAnalyzer* na modelovanie používateľa. Vybrané udalosti a informácie o nich taktiež zaznamenáva do pomocného XML súboru pre potreby ladenia.

Realizované časti navrhutej metódy

Vytvorená implementácia nástroja *SemanticLog* realizuje nasledovnú funkcionálnu:

- Prijímanie požiadaviek nástrojov o zaznamenanie udalostí.
- Sledovanie sedení používateľa a vytváranie súvislých postupností udalostí pre jednotlivých používateľov (pre každé sedenie zvlášť).
- Vytvorenie záznamu o udalosti so sémantikou a jeho zapísanie do databázy.

Obrázok 21 znázorňuje výpis zo záznamov udalostí zachytených pomocou nástroja *SemanticLog*. V stĺpcoch je uvedený čas zaznamenania udalosti, počiatočný a koncový stav zobrazenia, typ udalosti a sedenie používateľa. Stav zobrazenia zachytáva položky, ktoré boli zobrazené používateľovi, napr. fazety, ohraničenia alebo inštancie informačného priestoru. Napr. pre fazetu je zaznamenané URI vlastnosti, z ktorej bola vytvorená, pre ohraničenie je zaznamenaný počet vyhovujúcich inštancií a URI. Zaznamenáva sa tiež URI inštancií, resp. URI atribútov, ktoré boli pre inštancie zobrazené.

	id	timestamp	fromState	toState	typeOfEvent	session
1	1	2006-12-09 15:17:48.0	1	2	3	1
2	2	2006-12-09 15:17:52.0	2	3	1	1
3	3	2006-12-09 15:24:03.0	1	4	3	2
4	4	2006-12-09 15:24:07.0	4	5	1	2
5	5	2006-12-09 15:27:48.0	5	6	8	2
6	6	2006-12-09 15:31:37.0	6	7	8	2
7	7	2006-12-09 15:32:56.0	7	8	8	2
8	8	2006-12-09 15:35:37.0	8	9	8	2
9	9	2006-12-09 16:00:21.0	1	10	3	3
10	10	2006-12-09 16:00:25.0	10	11	1	3
11	11	2006-12-09 16:02:03.0	1	12	3	4
12	12	2006-12-09 16:02:07.0	12	13	1	4
13	13	2006-12-09 16:04:51.0	1	14	3	5
14	14	2006-12-09 16:04:55.0	14	15	1	5
15	15	2006-12-09 16:08:50.0	1	16	3	6
16	16	2006-12-09 16:08:54.0	16	17	1	6
17	17	2006-12-09 16:13:14.0	1	18	3	7

Obrázok 21: Ukážka zo záznamu udalostí zachytených pomocou nástroja *SemanticLog*.

¹³ Hibernate, <http://www.hibernate.org/> (22.11.2006)

Pre každú udalosť *SemanticLog* zaznamenáva aj viaceré atribúty (Obrázok 22). V stĺpcoch je pre jednotlivé atribúty uvedený typ atribútu, jeho hodnota a udalosť ku ktorej patrí, resp. nadradený atribút v prípade rekurzívnych atribútov.

	id 	value	TypeOfEventAttribute	event	parentAttribute
1	1	anonymous	2	1	(null)
2	2	anonymous	2	3	(null)
3	3	http://nazou.fiit.stuba.sk/nazou/ontologies/v0.6.17/offer-job#hasDutyLocation	4	5	(null)
4	4	http://nazou.fiit.stuba.sk/nazou/ontologies/v0.6.17/region#World	3	5	(null)
5	5	http://nazou.fiit.stuba.sk/nazou/ontologies/v0.6.17/region#North_America	3	6	(null)
6	6	http://nazou.fiit.stuba.sk/nazou/ontologies/v0.6.17/offer-job#hasDutyLocation	4	6	(null)
7	7	http://nazou.fiit.stuba.sk/nazou/ontologies/v0.6.17/offer-job#hasDutyLocation	4	7	(null)
8	8	http://nazou.fiit.stuba.sk/nazou/ontologies/v0.6.17/region#US	3	7	(null)
9	9	http://nazou.fiit.stuba.sk/nazou/ontologies/v0.6.17/classification#pcProfessionals	3	8	(null)
10	10	http://nazou.fiit.stuba.sk/nazou/ontologies/v0.6.17/offer-job#offersPosition	4	8	(null)
11	11	anonymous	2	9	(null)
12	12	anonymous	2	11	(null)

Obrázok 22: Ukážka zo záznamu atribútov udalostí zachytených nástrojom *SemanticLog*.

7 Experimentálne overenie metódy

Vlastnosti vybraných častí navrhutej metódy sme overili implementáciou nástrojov *Factic* a *SemanticLog* a vykonaním viacerých experimentov. Cieľom experimentov bolo overenie nasledovných aspektov navrhutej metódy:

- Záznam akcií používateľa so sémantikou.
- Efektívnosť práce s veľkým informačným priestorom.
- Prispôsobovanie sa charakteristikám používateľa.

Experimenty sme vykonali v doméne pracovných príležitostí (projekt NAZOU) a v čiastočne aj v doméne vedeckých publikácií (projekt MAPEKUS). Pre jednotlivé experimenty sme použili doménové ontológie, resp. ontológie používateľa z príslušných projektov. Počas experimentovania sme sa priamo nezaoberali tvorbou modelu používateľa, ktorú sme riešili s využitím špecializovaných externých nástrojov (Barla, 2006a).

7.1 Záznam akcií používateľa so sémantikou

V rámci overenia záznamu akcií používateľa so sémantikou sme úspešne overili tvorbu záznamu akcií používateľa počas práce s nástrojom *Factic* s využitím nástroja *SemanticLog*. Jednotlivé udalosti boli zachytené spolu s atribútmi a stavom zobrazenia vo fazetovom prehliadači.

Návrh nástroja *SemanticLog* ako „čistej“ webovej služby sa ukázal nevhodný, pretože prenos komplexných údajov o stave zobrazenia pomocou SOAP rozhrania webovej služby – ich serializácia a deserializácia boli veľmi časovo náročné. V závislosti od množstva údajov spracovanie volania webovej služby trvalo rádovo minúty, čo nie je akceptovateľné pre praktické použitie. Z tohto dôvodu sme zmenili návrh nástroja *SemanticLog* na hybridnú webovú službu, kde základné informácie o udalosti a ich integráciu sú riešené pomocou rozhrania webovej služby. Samotné uloženie údajov o stave zobrazenia v nástroji *Factic* je následne riešené pomocou priameho volania API nástroja *SemanticLog*. Použitie API na záznam udalostí skrátilo dobu spracovania záznamu na rádovo stovky milisekúnd.

Z vytvoreného záznamu akcií používateľa so sémantikou bolo následne možné odvodiť charakteristiky používateľa pomocou nástrojov na modelovanie používateľa (Andrejko et al., 2006a). Takto odvodené charakteristiky sme použili pre overenie prispôsobovania sa fazetového sémantického prehliadača používateľovi.

7.2 Efektívnosť práce s veľkým informačným priestorom

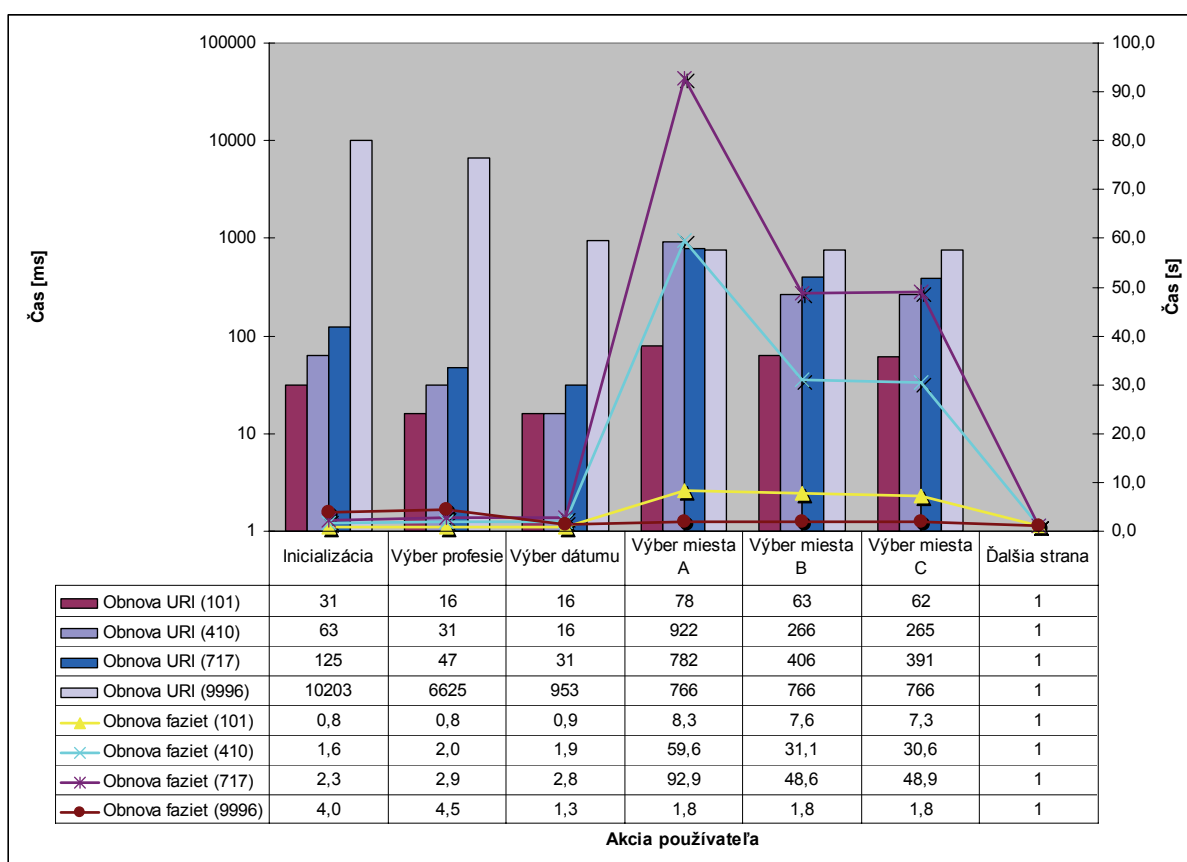
Vlastnosti metódy vzhľadom na jej použiteľnosť pre prácu s veľkým informačným priestorom (napr. dobu odozvy, zložitosť spracovania) sme experimentálne overili pre rôzne veľkosti databázy a rôzne počty faziet a ohraňení.

Overili sme:

- Celkovú dobu odozvy počas navigácie vo fazetovom sémantickom prehliadači.
- Rýchlosť vyhľadania výsledkov ohraňovania priestoru – vykonania dopytu do ontologického úložiska pre rôzne zložité dopyty.
- Rýchlosť obnovenia stavu faziet a ohraňování pre rôzne počty faziet a ohraňování.

Experimenty s väčšími dátovými vzorkami (stotísíce inštancií) odhalili nedostatky pri práci s ontologickým úložiskom Sesame, ktoré už pri relatívne jednoduchých dopytoch malo veľmi dlhú dobu odozvy. Použitie väčších dátových vzoriek si tiež vynútilo zmenu typu úložiska z úložiska v pamäti RAM na úložisko na v relačnej databáze MySQL na disku.

Veľké dátové vzorky, s rádovo státisícami inštancií, sme použili v doméne vedeckých publikácií, kde sa v najhorších prípadoch doba odozvy predĺžila na minúty, čo znemožnilo efektívnu prácu so systémom. Dôvodom spomalenia je „nezrelosť“ ontologického úložiska Sesame, ktoré nepodporuje základné operácie ako sú operátor COUNT, resp. ORDER BY, ktoré sme museli následne realizovať zistením všetkých inštancií a ich manuálnym spočítaním, resp. usporiadaním. Ďalším dôvodom dlhej doby odozvy je pravdepodobne veľmi obmedzené použitie vyrovnávacej pamäte v Sesame a tiež presun dát z operačnej pamäte na pevný disk. Na zlepšenie doby odozvy sme následne realizovali dočasné ukladanie medzivýsledkov na strane nástroja *Factic*. Graf 1 znázorňuje závislosť doby odozvy od veľkosti databázy.



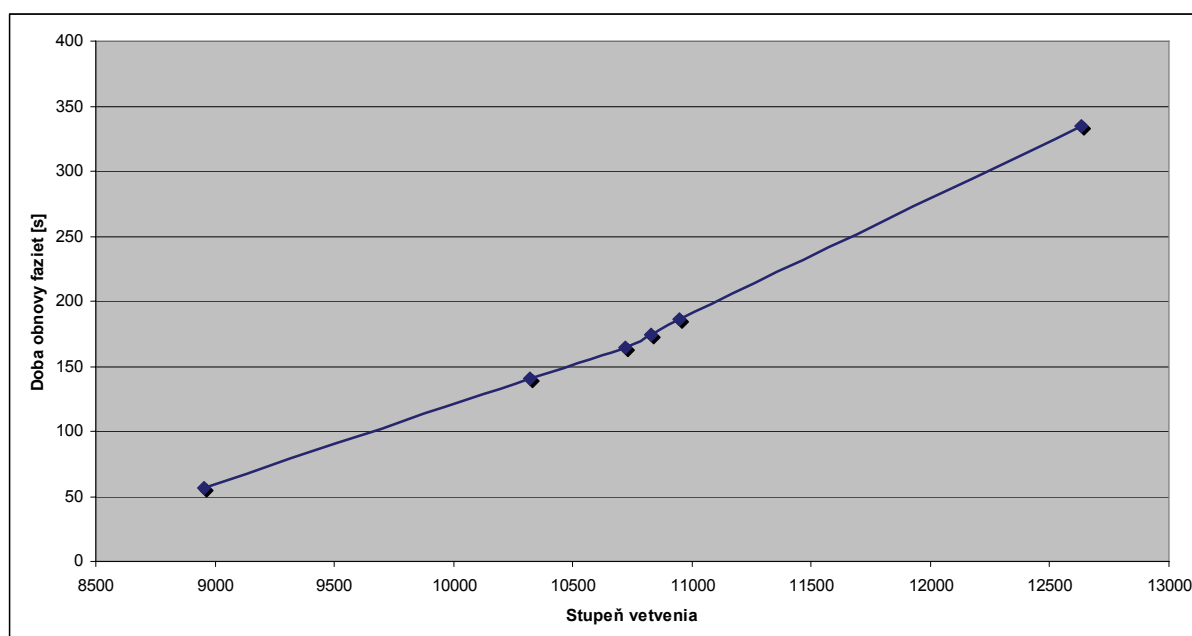
Graf 1: Závislosť doby obnovy faziet a URI inštancií od veľkosti databázy a zložitosti dopytu.

Stĺpce znázorňujú dobu vykonania dopytu – obnovu URI v milisekundách. V zátvorke je uvedená veľkosť databázy – počet inštancií pracovných ponúk (101, 410, 717) a počet publikácií (9996). Čiary znázorňujú dobu obnovenia stavu faziet v sekundách v závislosti od vykonanej akcie používateľa. Všetky uvedené hodnoty predstavujú spracovanie bez adaptácie.

Z grafu vyplýva, že doba obnovy faziet výrazne narastá pre komplexné hierarchické fazety (napr. miesto vykonania práce), ktoré majú vysoký stupeň vetvenia. Nárast doby spracovania je navyše silne ovplyvnený veľkosťou databázy, kde už pri 717tich inštanciách spracovanie trvá 92,9 sekundy. Po zjemnení výberu vo fazete sa hierarchia zjednoduší (posunieme sa hlbšie) a čas spracovania klesá na cca. 48 sekúnd, čo je stále nevyhovujúce.

Následne sme skúmali náročnosť obnovenia faziet v závislosti od počtu aktívnych faziet a ohraničení v nich (stupňa vetvenia). Počas experimentov sme definovali 11 faziet na základe doménovej ontológie pracovných ponúk, avšak komplexne štruktúrovaný informačný priestor umožňuje definovať ďaleko väčší počet faziet s vysokým stupňom vetvenia ohraničení. Keďže vopred nie je možné predpočítať hodnoty pre všetky kombinácie faziet a ohraničení, je nutné ich vyhodnocovať pri každom zvolení ohraničenia. Pre N faziet a priemerný stupeň vetvenia ohraničení vo fazete K je teoretická zložitosť obnovenia počtov inštancií vo fazetách $O(N * K)$, pričom celkový počet kombinácií faziet a ohraničení je K^N .

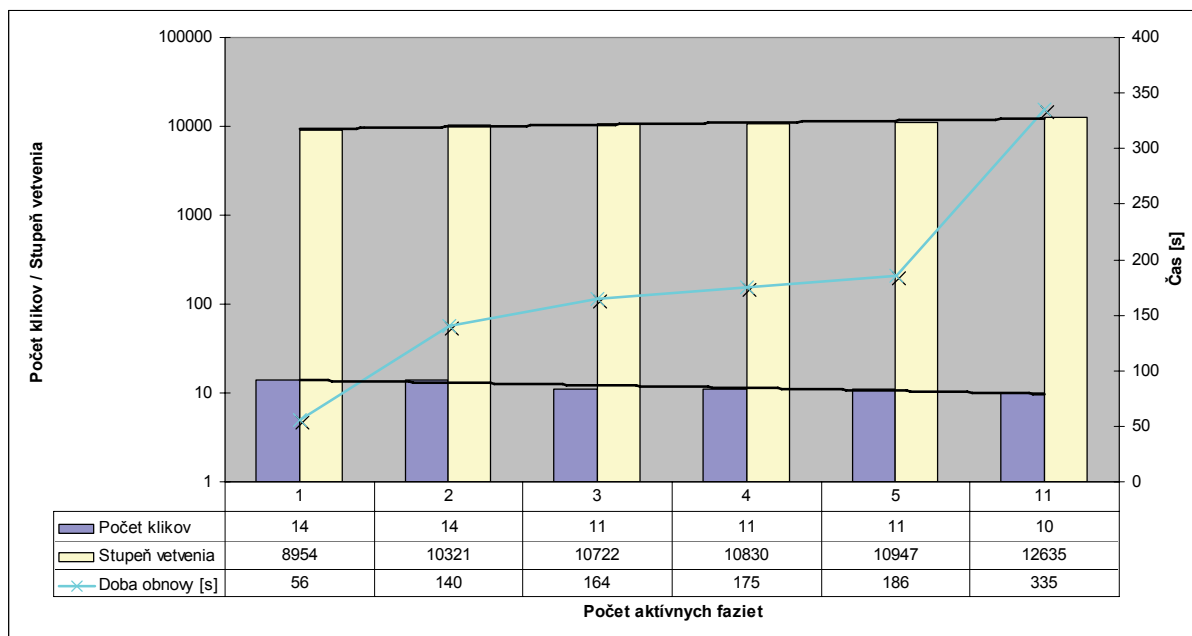
Experimentálne sme overili lineárnu závislosť doby obnovy faziet od počtu faziet pri danom priemernom stupni vetvenia (Graf 2). Pri experimente sme zbierali sumárne údaje doby odozvy a stupňa vetvenia počas formulácie komplexného dopytu tvoreného pomocou striedavej kombinácie viacerých hierarchických faziet (dátum nástupu do práce, profesia, miesto výkonu práce).



Graf 2: Závislosť celkovej doby obnovy faziet od celkového stupňa vetvenia.

Na príklade uvedeného komplexného dopytu sme analyzovali vplyv počtu aktívnych faziet na dobu obnovy stavu faziet, stupeň vetvenia a počet klikov nutný na formuláciu dopytu (Graf 3).

Experimenty sme vykonali s adaptívnymi fazetami pre 1, 2, 3, 4, 5 a 11 súčasne aktívnych faziet, pričom ostatné fazety boli neaktívne, avšak dostupné na vyžiadanie. Z experimentov vyplynulo, že s narastajúcim počtom aktívnych faziet približne lineárne narastá aj celkový stupeň vetvenia a doba spracovania, v závislosti od typu, resp. zložitosti fazety. Ekvivalentne k tomu klesá počet klikov potrebných na formulovanie dopytu, keďže nie je potrebné aktivovať neaktívne fazety.



Graf 3: Závislosť počtu klikov potrebných na formulovanie dopytu, stupňa vetvenia a doby obnovy od počtu aktívnych faziet.

Experimentom sme overili, že pomocou adaptácie faziet – usporiadaním na základe ich relevancie z modelu používateľa a výberom najrelevantnejších do obmedzenej množiny aktívnych faziet, je možné efektívne zrýchliť prácu s fazetovým prehliadačom za cenu vyššieho počtu klikov myšou. Pri 11 fazetách sa podarilo formulovať dopyt na 10 klikov za 335 sekúnd. Pri adaptívnom použití maximálne 3 faziet bolo nutných 11 klikov a len 164 sekúnd (približne polovica). Pri použití len jednej aktívnej fazety stačilo na formuláciu dopytu 56 sekúnd (približne šestina času), avšak bolo nutných celkovo 14 klikov.

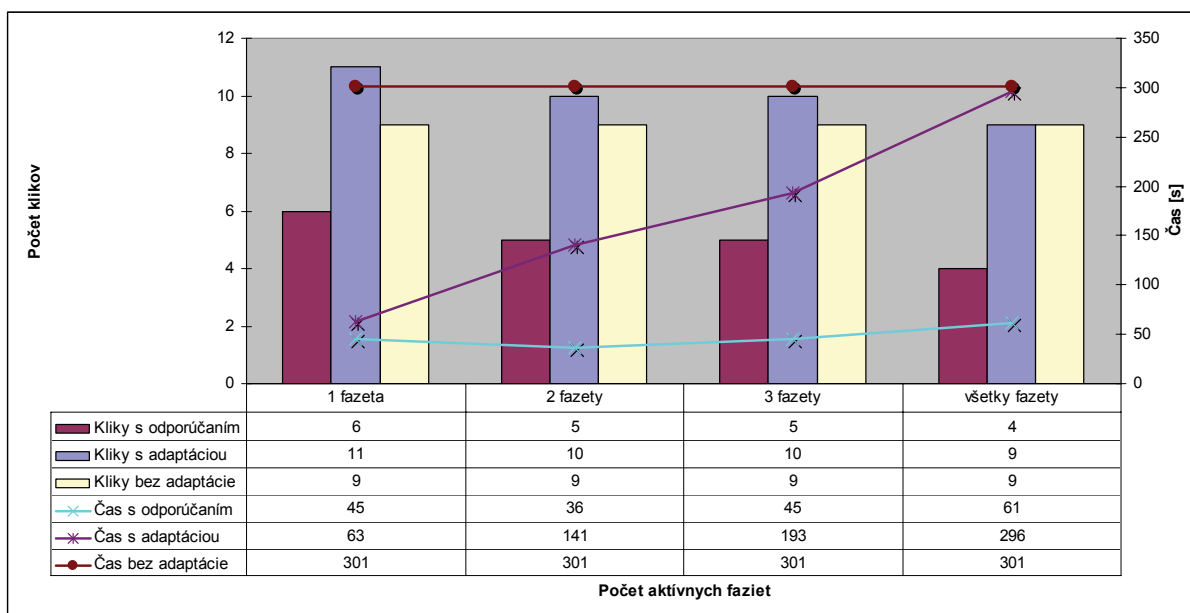
7.3 Prispôbovanie sa charakteristikám používateľa

Na overenie efektívnosti prispôbovania faziet a odporúčania ohraničení sme vykonali experimenty s vopred definovanými profilmi používateľov (napr. programátor, manažér, výskumník). Pre profil programátor v Kalifornii s nástupom do práce v októbri 2005 sme identifikovali množinu 9 cieľových inšancií pracovných ponúk. Experimentálne sme merali:

- Počet klikov myšou nutný na formulovanie dopytu a nájdenie inšancií.
- Čas potrebný na formulovanie dopytu a nájdenie cieľových inšancií.

Jednotlivé experimenty sme vykonali pre viaceré nastavenia fazetového prehliadača (Graf 4):

- Klasický fazetový prehliadač bez využitia adaptácie, ktorý simuluje prácu bez modelu používateľa.
- Adaptívny fazetový prehliadač s rôznym počtom aktívnych faziet s definovaným modelom používateľa.
- Adaptívny fazetový prehliadač s rôznym počtom aktívnych faziet a odporúčaním ohraničení na základe definovaného modelu používateľa.



Graf 4: Výsledky experimentov prehliadania informačného priestoru s a bez adaptácie.

Väčší počet aktívnych faziet vyžadoval nižší počet klikov na splnenie cieľa. Bez adaptácie bolo potrebných konštantne 9 klikov, s adaptáciou sa počet klikov pohyboval v rozsahu 11 až 9, jednu, resp. všetky aktívne fazety. Pomocou adaptácie s odporúčaním bolo možné počet potrebných klikov znížiť na 6 až 4, s využitím priameho výberu ohraničení v hĺbke hierarchie na základe odporúčania relevantných ohraničení z modelu používateľa.

Najdlhšie nájdenie cieľových inštancií pracovných ponúk bez využitia adaptácie (301 sekúnd). S využitím adaptácie fazetového prehliadača s premenlivým počtom aktívnych faziet sa potrebný čas skrátil na 63 sekúnd pre jednu fazetu a približne lineárne stúpil až k času 296 sekúnd pre všetky aktívne fazety. Pre všetky aktívne fazety bol adaptívny prístup o 5 sekúnd rýchlejší ako neadaptívny prístup.

Najkratšie trvalo splnenie cieľa pri adaptácii fazetového prehliadača s odporúčaním, kde pre jednu aktívnu fazetu bolo potrebných 45 sekúnd, pre dve aktívne fazety najkratšie – 36 sekúnd a pre všetky aktívne fazety 61 sekúnd. Pri všetkých aktívnych fazetách bolo odporúčanie rýchlejšie ako čisto adaptívny prístup pre jednu aktívnu fazetu.

Experimentálne sme určili približne optimálny počet aktívnych faziet na úrovni jednej až troch faziet pre prístup s odporúčaním a na úrovni jednej fazety bez odporúčania. Presný počet závisí od počtu a presnosti odporúčania ohraničení a prípadne od zložitosti dopytu.

8 Zhodnotenie

Cieľom našej práce bol návrh a overenie metód na zefektívnenie práce používateľa s veľkými otvorenými informačnými priestormi, ktoré v súčasnosti prinášajú viacero problémov – veľkú časovú náročnosť, obtiažnosť strojového spracovania informácií, „stratenie sa“ používateľa, resp. jeho zahltenie informáciami.

V rámci problematiky sme analyzovali aktuálne prístupy k navigácii v informačných priestoroch na webe ako aj prístupy personalizácii na základe modelu používateľa. Tieto využívajú napr. rôzne spôsoby klasifikácie informácií (jednoduché, hierarchické, fazetové), rôzne spôsoby ich prezentácie, či už textovo alebo graficky v 2D alebo 3D priestore. Rastúcu úlohu zohráva prispôbovanie sa systémov potrebám používateľa na základe modelu používateľa, avšak najčastejšie sa používa v uzavretých informačných priestoroch napr. pre e-vzdelávanie.

Na základe poznatkov získaných analýzou problematiky sme navrhli metódu personalizovanej navigácie so sémantikou pomocou rozšíreného fazetového sémantického prehliadača. Tento využíva tri základné prístupy k práci s informáciami:

- Fazetovú klasifikáciu a následne fazetový prehliadač a fazetovú navigáciu.
- Web so sémantikou v podobe reprezentácie informácií pomocou ontológií.
- Adaptívne webové systémy, konkrétne techniky adaptívnej navigácie.

Prínosy navrhutej metódy navigácie so sémantikou sú:

- Navigácia v ontológiách (webe so sémantikou).
- Podpora vizuálnej formulácie dopytov so sémantikou.
- Dynamické generovanie faziet na základe doménovej ontológie a modelu používateľa.
- Podpora pokročilého vyhľadávania a komplexných kombinácií ohraničení.
- Podpora orientácie vo veľkom otvorenom informačnom priestore.
- Zamedzenie zahltenia používateľa informáciami.
- Vedenie používateľa k naplneniu jeho potrieb a cieľov.

Realizáciu uvedených prínosov metóda dosahuje využitím viacerých prístupov, z ktorých každý prispieva k zabezpečeniu potrebnej funkcionality pre používateľa. Fazetová navigácia a pokročilý režim vyhľadávania umožňujú používateľovi efektívne vizuálne definovať dopyty so sémantikou do použitej doménovej ontológie. Dynamické generovanie faziet a ohraničení používateľovi umožňuje nielen pracovať s otvoreným informačným priestorom ale aj zvyšovať vyjadrovaciu silu dopytov pomocou zjemňovania ohraničení.

Zahlteniu používateľa informáciami predchádzame pomocou adaptácie faziet a ohraničení. Sprístupňujeme najmä relevantné fazety a ohraničenia, pričom skrývame tie menej dôležité. Orientáciu používateľa v informačnom priestore a pochopenie dôsledkov navigácie dosahujeme pomocou anotácie faziet, ohraničení a výsledkov vyhľadávania, ktoré už pred nasledovaním odkazu používateľa informujú o približnom informačnom obsahu

cieľového dokumentu. Globálnu pomoc používateľovi pri naplnení jeho cieľa riešime pomocou vedenia používateľa optimálnou cestou informačným priestorom.

Praktické použitie navrhutej metódy navigácie je možné v mnohých aplikačných oblastiach. Metódu možno principiálne použiť na vyhľadanie informácií a navigáciu v ľubovoľnom veľkom otvorenom informačnom priestore opísanom ontológiou. Ako príklad uvádzame doménu ponúk práce, dovolení, verejných objednávok, doménu vedeckých publikácií alebo doménu elektronického obchodu. Zaujímavým použitím je tiež doména výučbových systémov pre e-vzdelávanie.

Overenie navrhutej metódy sme realizovali v rámci dvoch väčších projektov na FIIT STU – projektu NAZOU (Návrat et al., 2006) a projektu MAPEKUS (Bieliková a kol., 2005a), ktoré definovali aj ohraničenia riešenia v podobe implementačného prostredia a kontextu nasadenia. Samotné overenie vybraných častí navrhutej metódy sme realizovali implementáciou prototypu dvoch softvérových nástrojov – nástroja *Factic* a nástroja *SemanticLog* a následným vykonaním experimentov v aplikačnej doméne pracovných ponúk a vedeckých publikácií.

Výsledky experimentov ukázali, že adaptácia fazetového prehliadača prináša významné zlepšenie výsledkov oproti neadaptívnym prístupom, najmä z hľadiska doby spracovania požiadaviek. Doplnenie adaptácie o odporúčanie na základe modelu používateľa, vytvoreného s využitím sémantiky akcií používateľa pri práci so systémom, prinieslo ďalšie zefektívnenie práce používateľa v podobe skrátenia času potrebného na vykonanie pokusných scenárov.

Zaznamenanie akcií používateľa v rozšírenom fazetovom sémantickom prehliadači pomocou nástroja *SemanticLog* umožnilo relatívne efektívne odvodenie charakteristík používateľa bez potreby zložitého predspracovania údajov. Použité charakteristiky bolo následne možné použiť pre prispôsobovanie v nástroji *Factic*.

Experimenty preukázali relatívnu nezrelosť súčasných riešení pre oblasť webu so sémantikou a prácu s ontológiami. Použité ontologické úložisko Sesame pracovalo značne neefektívne, pretože už pre relatívne malé množstvo údajov (stovky inštancií) a relatívne jednoduché dopyty malo celkovú dobu odozvy rádovo desiatky sekúnd.

Vybrané aspekty riešenia sme prezentovali vo forme viacerých vedeckých článkov v zborníkoch konferencií (Príloha C) IIT.SRC 2006 (Tvarožek, 2006a), Adaptive Hypermedia 2006 (Tvarožek, 2006b), ACM.SRC 2006 (Tvarožek, 2006c), a samotnou prezentáciou autorom. Ďalšie články boli publikované v spoluautorstve s ďalšími kolegami na konferencii DATAKON 2006 (Andrejko, Barla, Bieliková, Tvarožek, 2006a), SOFSEM 2007 (Tvarožek, Barla, Bieliková, 2007) a v rámci zborníka z pracovnej dielne projektu NAZOU (Návrat et al. 2006).

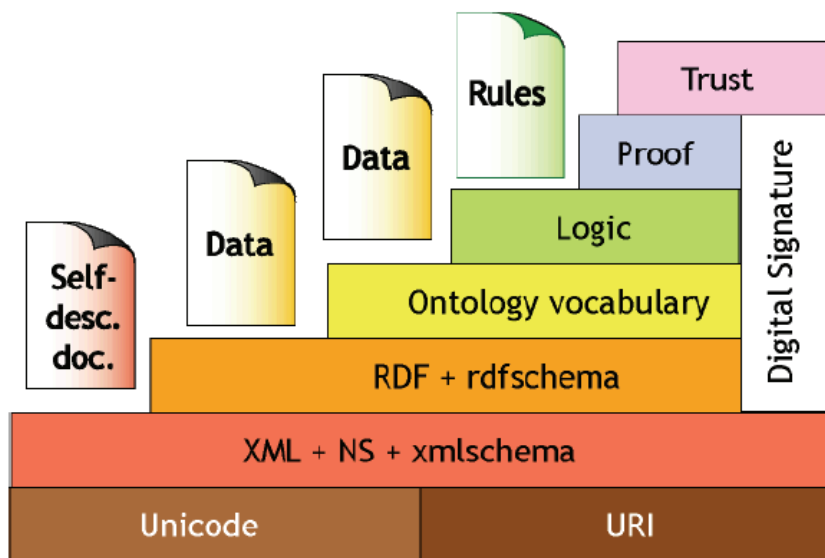
Možný smer ďalšej práce predstavuje realizácia navrhnutých pokročilých techník prispôsobovania výsledkov vyhľadávania a prepojenie s inými prístupmi v oblasti fazetovej navigácie so sémantikou, napr. pomocou techniky medzifaziet (angl. inter-facet) (Kruk, 2005). Zaujímavou alternatívou budúcej práce je aj prepojenie fazetového sémantického prehliadača s inými spôsobmi prezentácie a navigácie s využitím vizuálnej navigácie, napr. pomocou navigácie v zhlukoch.

Zjednodušiť navigáciu a zlepšiť prehľad používateľa o navštívenej časti informačného priestoru je možné pomocou dopracovania navrhutej metódy o vizualizáciu histórie navštívených stránok (dopytov) pomocou (hyper)grafu s prípadnou anotáciou a možnosťou návratu k zvolenej stránke. Zvýraznenie informácií vo výslednom dokumente možno realizovať prepojením s externým anotačným nástrojom, čím sa zvýši efektívnosť práce používateľa s výsledkami vyhľadávania. Zvýšenie praktickej použiteľnosti metódy možno tiež dosiahnuť jej upravením pre použitie na mobilných zariadeniach s prihliadnutím na ich špecifické vlastnosti.

Príloha A Princípy webu so sémantikou

Úspech a rozšírenie súčasného webu so sebou prinieslo aj viacero zo spomínaných problémov. Navrhnutým riešením časti uvedených problémov je aj projekt *webu so sémantikou* (angl. semantic web), ktorý v súčasnosti vedie hlavný tvorca súčasného webu – Tim Berners Lee z World Wide Web konzorcia (W3C). Cieľom tohto projektu je vytvoriť univerzálny spôsob na výmenu informácií pomocou pridania významu (sémantiky) k obsahu webových dokumentov. Prostriedkom na dosiahnutie tohto cieľa je rozšírenie klasického webu s využitím štandardov a nových značkových jazykov (Wikipedia, 2006a).

Web so sémantikou je však podľa svojho autora Tima Berners Lee len jedným z míľnikov na ceste k *webu dôvery* (angl. web of trust), ktorý si navyše kladie za cieľ aj opísať dôveryhodnosť, resp. pravdivosť uvádzaných údajov. Obrázok A-1 opisuje víziu webu so sémantikou a webu dôvery podľa Tima Berners Lee. Samotný web so sémantikou pozostáva zo štyroch spodných vrstiev, ktoré zahŕňajú opis údajov na webe pomocou samoopisných dokumentov a ontológií. Tieto sú vytvorené pomocou XML a XML Schema s využitím menných priestorov, pričom údaje a vzťahy medzi dokumentmi sú opísané pomocou RDF (Resource Description Framework), RDFS (RDF Schema) a OWL (Web Ontology Language).



Obrázok A-1: Vízia webu so sémantikou podľa Tima Berners Leeho, prevzaté z (Shah et al., 2002).

Po vytvorení webu so sémantikou bude možné vytvoriť web dôvery, ktorý zahŕňa tri vrchné vrstvy – logiku, dokazovanie a vierohodnosť, ktoré spolu s elektronickým podpisom budú slúžiť k určeniu vierohodnosti zdrojov informácií na webe a umožnia tak rozlíšiť informácie pravdivé od nepravdivých alebo neoverených.

Hlavným prínosom webu so sémantikou je zavedenie štandardov na opis sémantiky informácií na webe pomocou ontológií a následná možnosť strojového spracovania webových dokumentov.

A.1 Hlavné črty webu so sémantikou

Klasický web tvoria dokumenty. Web so sémantikou tvoria vzťahy medzi zdrojmi, ktoré označujú *objekty reálneho sveta*, ako sú napr. ľudia, miesta alebo udalosti (Guha et al., 2003). Ako príklad možno uviesť Beethovena, jeho Deviatu symfóniu a dirigenta Herberta von Karajana.

Samotný web so sémantikou je rozšírením súčasného webu. *Zdroje na webe so sémantikou obsahujú mnoho odkazov na HTML dokumenty*. Tieto odkazy spravidla spájajú koncepty webu so sémantikou s webovými stránkami, ktoré sú k nim najrelevantnejšie (Guha et al., 2003). Napr. zdroj opisujúci Herberta von Karajana obsahuje niekoľko odkazov na webové stránky, ktoré obsahujú relevantné informácie o tomto dirigentovi.

Informácie v dokumentoch dostupných v rámci klasického webu sú prispôbené na *prezentáciu a spracovanie ľuďmi* (čítanie) a je ťažké ich spracovať strojovo. Naproti tomu informácie na webe so sémantikou sú už od začiatku ukladané spôsobom, ktorý priamo podporuje *strojové spracovanie* informácií.

Zdroj zodpovedajúci dirigentovi Herbertovi von Karajanovi nie je textový reťazec „Herbert von Karajan“, ale zdroj označujúci osobu. Vo svete môže existovať viacero ľudí s týmto menom, avšak tento zdroj opisuje práve jedného z nich.

Dôležitou je však spomínaná možnosť strojového spracovania týchto údajov vďaka ich sémantickému zápisu, ktorý umožňuje stroju napr. určiť, že sa jedná o človeka, o skladateľa a že sa volá Herbert von Karajan. Klasická webová stránka o tomto dirigentovi obsahuje podobné informácie a spôsob ich zobrazenia v prehliadači, pričom stroj z nej nie je (jednoducho) schopný zistiť o čo sa v skutočnosti jedná.

Význačným aspektom webu so sémantikou je *distribúovaná rozšíriteľnosť*, čo znamená, že rôzne webové stránky môžu prispievať údajmi k jednému zdroju informácií. V nadväznosti na predchádzajúci príklad môžu existovať rôzne stránky o tom istom dirigentovi. Jedna stránka môže obsahovať jeho biografiu, iná môže predávať CD disky s jeho predstaveniami a ďalšia môže obsahovať zoznam jeho minulých predstavení. Každá z týchto stránok môže zverejňovať informácie o danom zdroji a tak rozširovať spoločnú bazu znalostí.

Distribúovateľná rozšíriteľnosť webu so sémantikou však prináša aj problémy, pretože niektoré zverejnené informácie nemusia byť pravdivé. Ľudia dokážu použiť svoju inteligenciu na to, aby určili či daná informácia je pravdivá alebo nie, resp. či pochádza z hodnoverného zdroja alebo nie. Stroje však v súčasnosti nemajú túto schopnosť (Guha et al., 2003). Riešenie tohto problému by mala priniesť až ďalšia verzia webu – *web dôvery*.

A.2 Ontológia

V informatike existuje viacero definícií pôvodne filozofického pojmu ontológia. Zjednodušene možno povedať, že ontológia je slovník pojmov s uvedením vzťahov medzi týmito pojmami. Formálnejšou definíciou ontológie je *explicitná formálna špecifikácia zdieľanej konceptualizácie* (Studer et al., 1998). Veľký význam v tejto definícii má práve slovo *zdieľanej*, pretože efektívne využitie ontológií je možné až vtedy, keď sa dostatočne veľká skupina ľudí, resp. poskytovateľov informácií zhodne na spoločnej ontológii.

Vo svojej podstate ontológia predstavuje spôsob zápisu informácií pomocou konceptov a vzťahov medzi nimi. Existuje viacero spôsobov zápisu ontológie s rôznou úrovňou expresívnosti. „Slabú“ sémantiku je možné vyjadriť pomocou taxonómií a entitno-relačného modelu. „Silnejšiu“ sémantiku možno zapísať pomocou synonymických slovníkov (tezaurus) a konceptuálnych modelov zapísaných v RDF. Na zápis „silnej“ sémantiky sa používajú jazyky DAML+OIL, resp. OWL, pričom na zápis vlastností a vzťahov využívajú deskriptnú logiku, resp. logiku prvého rádu (Grlický, 2006). Pretože v oblasti webu so sémantikou bol za štandard prijatý zápis ontológie v OWL, bližšie opíšem práve tento jazyk.

Význam zápisu informácií pomocou ontológie spočíva v tom, že ontológia umožňuje zapísať aj význam (sémantiku) informácií. Ak je napr. na nejakej stránke opísaný procesor (CPU), tak okrem samotného textu ontológia umožňuje zapísať aj jeho sémantiku. V tomto prípade by to napr. mohla byť skutočnosť, že sa jedná o recenziu, opis architektúry a porovnanie výkonu nového procesoru s inými. Iná stránka by mohla obsahovať informáciu o tom, že je to obchod s výpočtovou technikou s opisom názorov na jednotlivé produkty od zákazníkov.

A.2.1 Web Ontology Language (OWL)

Jazyk OWL vychádza z jazykov RDFS a DAML+OIL. Od roku 2004 predstavuje štandard pre zápis ontológií pre web so sémantikou. Sám jazyk OWL je rozšírením XML a RDF, resp. RDFS, pričom rozširuje slovník pojmov RDF a jeho formálnu sémantiku. OWL pozostáva z troch „podjazykov“, z ktorých každý poskytuje inú úroveň expresívnosti (Horridge et al., 2004):

- *OWL Lite* predstavuje najjednoduchšiu variantu s obmedzenou vyjadrovacou schopnosťou a je primárne určený na jednoduchú migráciu existujúcich tezaurov a jednoduchých hierarchií do OWL.
- *OWL DL* je založené na deskriptnej logike, ktorá tvorí významnú časť logiky prvého rádu. OWL DL má značne vyššiu vyjadrovaciu schopnosť ako OWL Lite a je možné ho využiť aj pre automatické odvodzovanie nových znalostí. Odvodzovače (napr. Racer¹⁴, Pellet) umožňujú automatické overenie konzistencie konceptov v ontológii, ako aj odvodzovanie klasifikačných hierarchií pre ontológie zapísané v OWL DL.

¹⁴ RacerPro – OWL odvodzovač, <http://www.racer-systems.com/> (10.12.2006)

- *OWL Full* je najexpresívnejšia verzia z uvedených jazykov. Oproti *OWL DL* umožňuje v reláciách uvádzať aj triedy a nie len inštancie. Použitie *OWL Full* je však v súčasnosti obmedzené na aplikácie, ktoré vyžadujú vysokú expresívnosť bez záruky konečnosti procesu odvodzovania nad takto zapísanou ontológiou. Súčasné odvodzovače zatiaľ nepodporujú odvodzovanie nad *OWL Full*. Navyše toto odvodzovanie je veľmi výpočtovo a pamäťovo náročné a nie je zaručená jeho konečnosť.

Každý z uvedených variantov jazyka *OWL* je vhodný na niečo iné, pričom v súčasnosti je asi najvhodnejšie pre nové systémy použiť variant *OWL DL*. Tento umožňuje zachytiť mnohé vzťahy medzi triedami a inštanciami tried ako prieniky, odvodenia, hierarchie, kardinalitu vlastností a ďalšie. Následne umožňuje odvodzovanie nad takto zapísanou bázou znalostí.

Pre praktické použitie ontológií zapísaných v *OWL* je nevyhnutná aj existencia vhodných nástrojov na ich tvorbu a editovanie. Keďže jazyk *OWL* je relatívne nový, existuje zatiaľ len obmedzená množina nástrojov, ktoré ho podporujú. Voľne dostupný nástroj *Protegé*¹⁵ často používaný na prácu s ontológiami má podporu pre *OWL* v podobe *OWL* zásuvného modulu (angl. plugin). Pri práci s *OWL* sa však často vyskytujú problémy, nástroj „padá“ alebo sa správa značne nedeterministicky.

A.2.2 Ontológia domény

Predpoklad k využitiu webu so sémantikou je vznik ontológií opisujúcich jednotlivé oblasti ľudskej činnosti. Príkladom môže byť doména pracovných príležitostí, ktorá opisuje koncepty a entity pôsobiace na trhu práce a vzťahy medzi nimi. Inou doménou môže byť oblasť medicíny, napr. doména zaoberajúca sa diagnostikou a liečbou srdcových chorôb. Celkovo je tvorba ontológie pre danú doménu zložitý proces, ktorý vlastne nikdy nekončí, pretože v každej oblasti dochádza časom k zmenám, ktoré sa musia premietnuť aj do príslušnej doménovej ontológie.

Z hľadiska použiteľnosti doménovej ontológie je dôležité, aby mala dostatočnú vyjadrovaciu schopnosť na to, aby dokázala zachytiť všetky relevantné aspekty danej oblasti. Pre web so sémantikou je ďalej dôležité, aby táto ontológia nebola „tajná“ ale aby bola rozšíriteľná, verejná a zdieľaná aj s inými inštitúciami. Nezverejnenie ontológie vedie k vzniku viacerých ontológií pre tú istú doménu, ktoré budú potom využiteľné len na interné účely tej ktorej organizácie. Prípadné ďalšie využitie týchto ontológií bude podmienené vznikom mapovania jednej ontológie na druhú, čo spôsobí zbytočné komplikácie a v konečnom dôsledku nemusí byť ani možné.

A.2.3 Ontológia používateľa

Spôsob zápisu informácií pomocou ontológie nie je obmedzený na zápis informácií z konkrétnej domény, ale je možné ho použiť aj na zápis znalostí o používateľovi. V tejto

¹⁵ The Protégé Ontology Editor and Knowledge Acquisition System, <http://protege.stanford.edu/> (10.12.2006)

oblasti ontológie prinášajú mnohé výhody, pretože predstavujú štandardný zápis informácií o používateľovi ako aj zápis ich sémantiky. Práve štandardný spôsob zápisu informácií následne umožňuje aj zdieľanie týchto informácií medzi rôznymi systémami, resp. nástrojmi a ich strojové spracovanie. Nezanedbateľným prínosom zápisu modelu používateľa pomocou ontológie ak už existuje doménová ontológia je skutočnosť, že oba modely majú zhodný spôsob zápisu, môžu obsahovať spoločné časti a prípadne zdieľať vybrané koncepty.

A.3 Vyhľadávanie s využitím sémantiky

Doplnenie klasického webu o sémantiku a následná možnosť strojového spracovania informácií je hlavným prínosom webu so sémantikou. Sémantické vyhľadávanie na rozdiel od klasického fulltextového vyhľadávania nevyhľadáva na základe výskytu a počtu kľúčových slov v dokumentoch, ale podľa sémantiky jednotlivých zdrojov a súvisiacich dokumentov. Z uvedeného vyplýva, že sémantické vyhľadávanie sa svojou povahou značne líši od fulltextového vyhľadávania.

Typickým príkladom rozdielov sú výsledky vyhľadávania. Fulltextové vyhľadávanie využíva textové informácie a spravidla vráti veľký počet výsledkov, z ktorých mnohé nie sú relevantné, ale vyhovujú dopytu – obsahujú zadané kľúčové slová. Na druhú stranu sémantické vyhľadávanie využíva len sémantiku informácií a vráti buď všetky správne výsledky spĺňajúce zadaný dopyt alebo nevráti žiadne výsledky (Zhang et al., 2005). Rozdiel je práve v tom, že zatiaľ čo fulltextové vyhľadávanie nájde aj všetky iné významy kľúčových slov, sémantické vyhľadávanie nájde len želané výsledky.

V súčasnosti ešte je väčšina informácií na webe uložená v podobne, ktorú sémantické vyhľadávanie nie je schopné využiť, a preto nie je možné prejsť k čisto sémantickému vyhľadávaniu.

A.3.1 Kombinované vyhľadávanie

Alternatívnym riešením je kombinácia fulltextového a sémantického vyhľadávania. V tejto oblasti existuje viacero prístupov s rôznou mierou úspešnosti. Najjednoduchším je využitie dopytu so sémantikou v prípade neúspešnosti jeho konverzia na fulltextový. Táto metóda však neposkytuje dostatočne dobré výsledky a nie je schopná postihnúť zložitejšie dopyty. S opačným prístupom – fulltextovým vyhľadávaním a dodatočným použitím sémantického vyhľadávania som sa nestretol. Pokiaľ by totiž sémantické vyhľadávanie vedelo dať lepšie výsledky je vhodné ho použiť priamo.

V (Zhang et al., 2005) autori navrhujú komplexnejšiu metódu vyhľadávania, kde sa vybrané časti dopytu od používateľa vyhľadávajú pomocou sémantického vyhľadávania a odvodzovania a na nájdenie zvyšných častí sa použije fulltextové vyhľadávanie. Uvádzajú komplexný model, v ktorom kombinujú prístupy modelu založenom na fuzzy deskripčnej logike a formálnom dopytovaní v deskripčnej logike. Napr. pre dopyt „nájdi dokumenty napísané vyšším manažmentom súvisiace s webom so sémantikou“ môžu byť všetci autori

z vyššieho manažmentu určení odvodzovaním nad databázou so sémantikou (obsahuje zamestnancov podniku) a súvislosť s webom so sémantikou môže byť určená pomocou hľadania kľúčových slov (napr. web so sémantikou), pretože databáza neobsahuje sémantiku obsahu dokumentov.

V porovnaní s jednoduchým kombinovaným modelom fulltextového a sémantického vyhľadávania dosahoval navrhnutý model podstatne lepšie výsledky. Aj napriek tomu však uvedený model má niekoľko obmedzení. Jedným z nich je obmedzená schopnosť používateľov formulovať dopyty v navrhnutom modeli. Ďalšie obmedzenie modelu vyplýva z obmedzenej vyjadrovacej schopnosti použitej fuzzy logiky.

V súčasnosti existuje niekoľko nástrojov na sémantické vyhľadávanie na webe s obmedzenou funkcionalitou. Viaceré z týchto nástrojov zbierajú sémantické anotácie z webových stránok a ukladajú ich do bázy znalostí (SHOE, OWLIR). Používateľ má následne možnosť vykonávať vyhľadávanie nad takto vytvorenou bázou znalostí. Niektoré nástroje v prípade neúspechu umožňujú konverziu formálneho dopytu na dopyt vhodný pre webové vyhľadávacie stroje (SHOE). Iné umožňujú aj jednoduché odvodzovanie nad vytvorenou bázou znalostí a kombináciu fulltextových a sémantických dopytov (OWLIR) (Zhang et al., 2005).

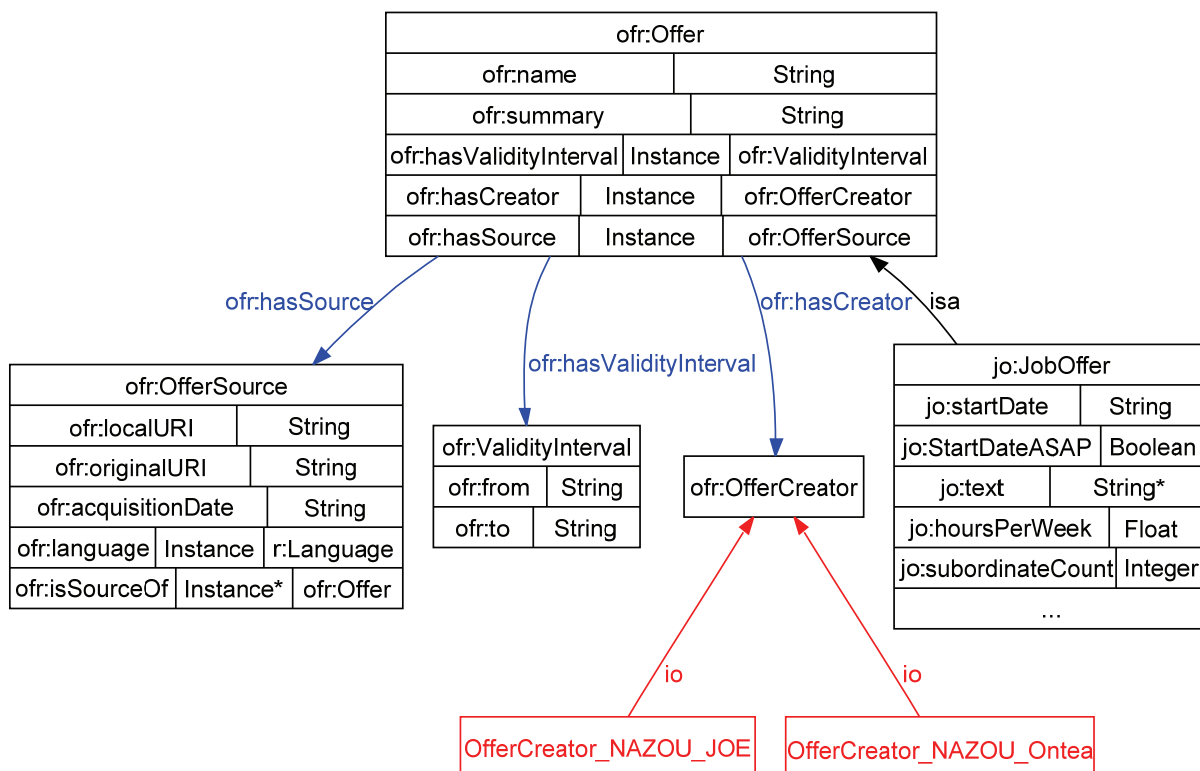
Pri práci s otvoreným informačným priestorom bez centrálnej bázy znalostí, je možné použiť technológie dynamického vyhľadávania (angl. dynamic search), ktoré v čase spracovania dopytu prehľadávajú množinu zdrojov informácií na (sémantickom) webe. Existujúce prístupy v tejto oblasti sú Fish search, Shark search alebo InfoSpiders (Levene, Wheeldon, 2003).

Príloha B Ontologické modely projektov NAZOU a MAPEKUS

B.1 Doménová ontológia projektu NAZOU – model pracovnej ponuky

Doménová ontológia predstavuje explicitnú konceptualizáciu pracovnej ponuky. Pri konceptualizácii pracovnej ponuky využíva a prípadne aj dopĺňa ďalšie ontológie, ktorých doména je nezávislá od zvolenej aplikačnej domény pracovných ponúk. Tieto ontológie sa môžu uplatniť aj pri inej aplikačnej doméne, ktorá využíva nimi definované koncepty.

- Ontológia *region* (používaná predpona „r:“) – opisuje doménu regiónov, krajín, jazykov a mien, ktoré sa používajú v daných regiónoch;
- Ontológia *classification* (predpona „c:“) – definuje hierarchie pre priemyselné odvetvia, profesie, úrovne vzdelania, kvalifikácie a rôzne usporiadania;
- Ontológia *offer* (predpona „ofr:“) – opisuje doménu všeobecnej ponuky (Obrázok B-1). Všeobecná ponuka je reprezentovaná triedou *ofr:Offer*. Každá ponuka má svoj zdroj (*ofr:OfferSource*), interval platnosti (*ofr:ValidityInterval*) a identifikáciu nástroja/spôsobu, ktorý bol použitý na vloženie ponuky do ontológie (*ofr:OfferCreator*).



Obrázok B-1: Vzťahy medzi triedami Offer, OfferSource a JobOffer.

Trieda JobOffer

- `jo:hoursPerWeek` – dátového typu `xsd:float` – počet hodín práce za týždeň;
- `ofr:name` – dátového typu `xsd:string` – názov ponuky;
- `jo:startDate` – dátového typu `xsd:date` – dátum nástupu;
- `jo:startDateASAP` – dátového typu `xsd:boolean` – nástup ihneď;
- `jo:subordinateCount` – dátového typu `xsd:int` – počet podriadených.

- *jo:ApplyInformation* – informácie k podávaniu žiadostí o zamestnanie;
- *jo:Benefit* – ponúkaná výhoda (auto, byt, poistenie,...);
- *jo:ContractType* – pracovný pomer (trvalý, dočasný, na dohodu);
- *jo:JobTerm* – pracovný úväzok (plný, polovičný,...);
- *c:ManagmentLevel* – vyjadruje úroveň ponúkanej pracovnej pozície z hľadiska umiestnenia v hierarchii manažmentu spoločnosti;
- *jo:Organization* – organizácia, ktorá ponúka alebo sprostredkúva prácu;
- *jo:Prerequisite* – predpoklady na uchádzača o zamestnanie požadované zamestnávateľom;
- *c:ProfessionClassification* – klasifikácia profesií podľa OSN;

- *r:Region* – región v ktorom sa vykonáva práca;
- *jo:Responsibility* – zodpovednosť;
- *jo:Salary* – plat;
- *jo:TravelingInvolved* – vyjadruje či a nakoľko je náplňou práce cestovanie.

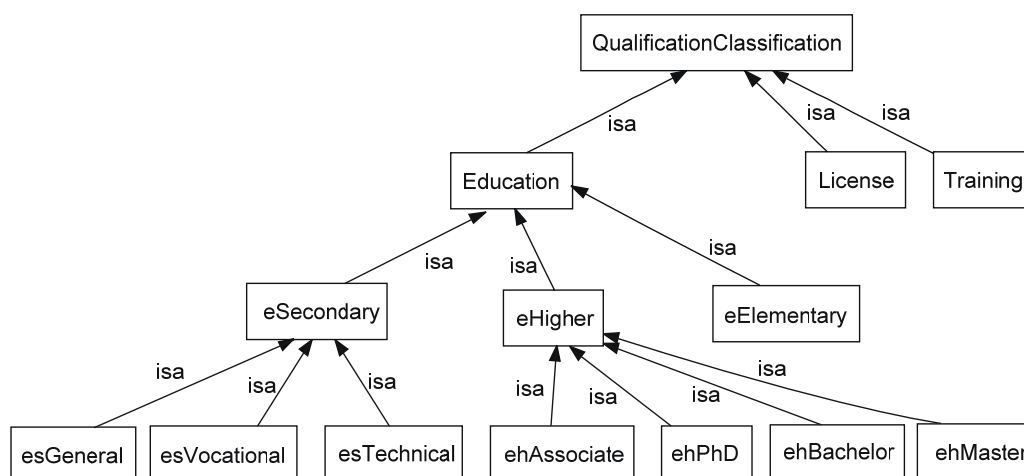
Trieda Prerequisite

Pri pracovnej ponuke sú veľmi dôležité predpoklady, kladené na uchádzača. Tie modeluje trieda *Prerequisite*. Predpoklady delí na dve skupiny:

- Požadované (angl. required) – musí uchádzač jednoznačne spĺňať;
- Uprednostňované (angl. preferred) – predstavujú pre uchádzača výhodu pri ich splnení.

Požiadavky, ktoré má zamestnávateľ na zamestnanca smerujú na triedy ontológie *classification* a môžu byť troch typov:

- Požiadavky na kvalifikáciu zamestnanca – *QualificationClassification*;
- Požiadavky na skúsenosti zamestnanca – *ExperienceClassification*;
- Požiadavky súvisiace so zamestnancom ako takým – *PersonalAttributeClassification*.



Obrázok B-3: Klasifikácia kvalifikácií.

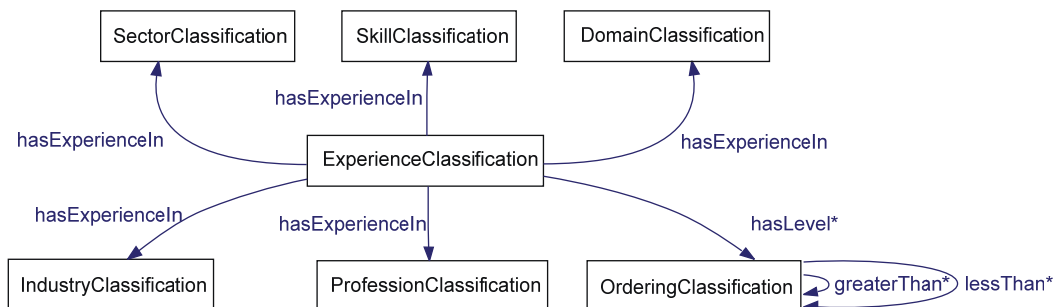
Trieda QualificationClassification

Klasifikácia kvalifikácií (Obrázok B-3) pozostáva z klasifikácie vzdelania (*Education*), z licencií, ktoré preukazujú absolvovanie určitej skúšky uchádzačom (*Licenses*) a zo zručností nadobudnutých na školeniach a kurzoch (*Training*).

Trieda ExperienceClassification

Klasifikácia skúseností umožňuje modelovať požiadavky na skúsenosti uchádzača o zamestnanie. Každá skúsenosť má priradený určitý stupeň (*hasLevel*) alebo môže byť vyjadrená hodnotou určitej veličiny (napríklad počet najazdených kilometrov). Skúsenosť môže byť vyjadrená vo vzťahu k určitej doméne znalostí (zvyčajne študijný odbor), zručnosti,

sektoru (napr. akademický, súkromný), priemyselnému odvetviu, alebo profesii (Obrázok B-4).



Obrázok B-4: Vzťah klasifikácie skúseností k iným klasifikáciám.

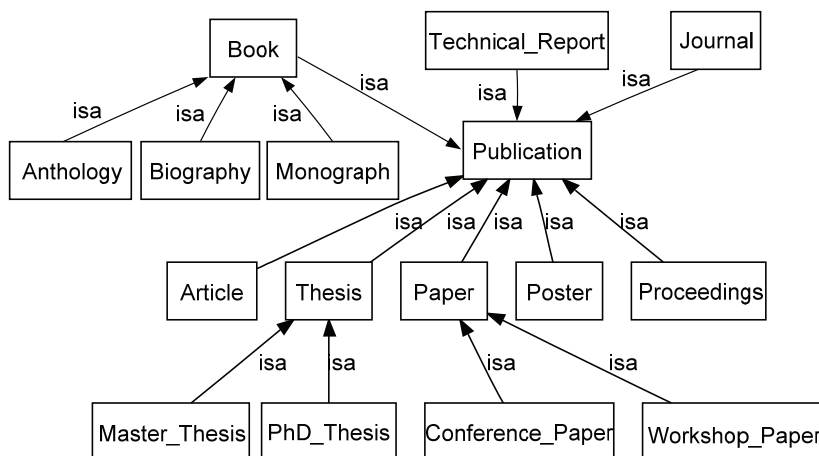
Trieda PersonalAttributeClassification

Klasifikácia osobných atribútov uchádzača o zamestnanie umožňuje modelovať požiadavky, ktoré súvisia s osobnostnými predpokladmi uchádzača o zamestnanie. Ide napríklad o schopnosti analytického myslenia, či komunikácie, alebo bezchybný zrak (nulové dioptrie).

B.2 Doménová ontológia projektu MAPEKUS – model publikácie

Doménová ontológia predstavuje explicitnú konceptualizáciu publikácie. Pri konceptualizácii publikácie využíva ďalšie ontológie, ktorých doména je nezávislá od zvolenej aplikačnej domény.

- Ontológia *region* (používaná predpona „r:“) – opisuje doménu regiónov, krajín, jazykov a mien, ktoré sa používajú v daných regiónoch;
- Ontológia *party* (používaná predpona „p:“) – konceptualizuje (zmluvnú) stranu vo vzťahu. Stranou môže byť na najvyššej úrovni fyzická osoba alebo organizácia.



Obrázok B-5: Hierarchia publikácií.

Hierarchia podtried triedy *Publication* (Obrázok B-5) obsahuje všetkých typy publikácií nachádzajúce sa v ontológii.

Opis vybraných tried

- *Article* – trieda reprezentujúca článok v časopise.
- *Book* – predstavuje knihu a obsahuje ďalšie podtriedy :
 - *Anthology* – zbierka prác od rôznych autorov,
 - *Biography* – životopisné dielo,
 - *Monography* – odborné dielo opisujúce jednu problematiku.
- *Journal* – reprezentuje periodikum určené pre odbornú verejnosť.
- *Paper* – predstavuje odborný príspevok. Delí sa na ďalšie podtriedy:
 - *Conference paper* – konferenčný odborný príspevok.
 - *Technical paper* – odborný príspevok opisujúci napríklad technické aspekty určitého systému.
- *Proceedings* – súbor odborných príspevkov vydaných v kontexte nejakej konferencie alebo iného stretnutia odbornej verejnosti. Obyčajne sa vydáva v knižnej podobe.
- *Poster* – reprezentuje plagáty s odbornou tematikou.
- *Technical report* – formálna správa, ktorá opisuje prínos v oblasti aplikovaného výskumu, poukazuje na detaily a výsledky riešenia nejakého vedeckého problému.
- *Thesis* – práca obsahujúca výsledky výskumu vypracovaná kandidátom na titul v rámci štúdia. Člení sa na podtriedy:
 - *MasterThesis* – práca vypracovaná na druhom stupni vysokoškolského štúdia.
 - *PhDThesis* – práca vypracovaná na treťom (doktorandskom) stupni vysokoškolského štúdia.

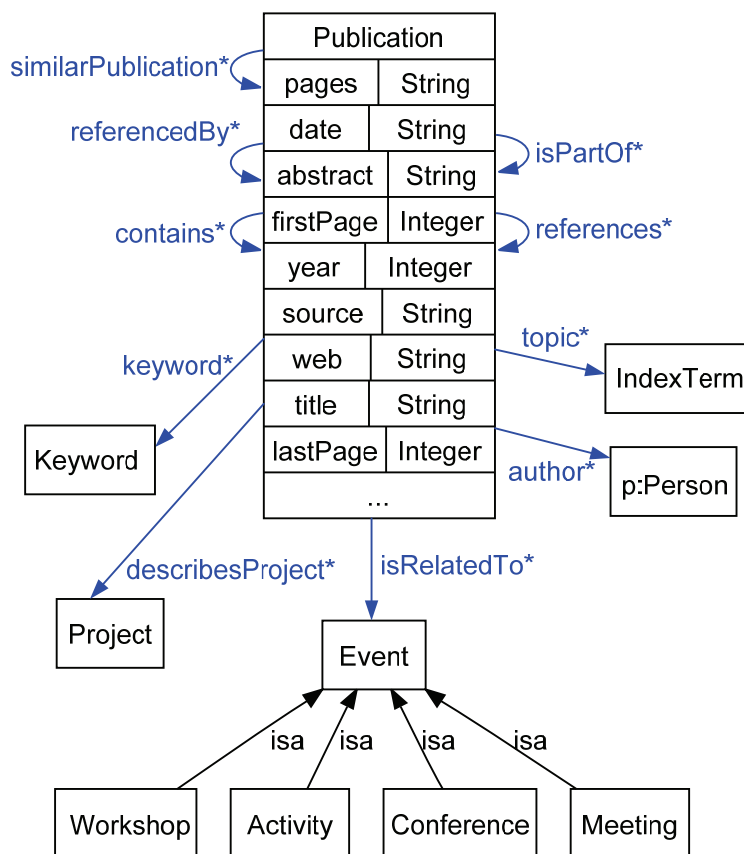
Trieda Publication

Obrázok B-6 znázorňuje dátové a objektové vlastnosti triedy *Publication*, ktoré opisujú hlavné atribúty publikácií. Dátové vlastnosti:

- *pages* – typu *xsd:string* – strany na ktorých sa nachádza publikácia ak je časťou inej publikácie.
- *date* – typu *xsd:string* – dátum vydania/sprístupnenia publikácie.
- *abstract* – typu *xsd:string* – abstrakt publikácie.
- *firstPage* – typu *xsd:int* – prvá strana publikácie v rámci inej publikácie.
- *year* – typu *xsd:int* – rok vydania publikácie.
- *source* – typu *xsd:string* – zdroj publikácie (napr. doi odkaz).
- *web* – typu *xsd:string* – odkaz na asociovaný zdroj na webe.
- *title* – typu *xsd:string* – názov publikácie.
- *lastPage* – typu *xsd:int* – posledná strana publikácie v rámci inej publikácie.

Objektové vlastnosti:

- *Keyword* – trieda `Keyword` – jednoduché kľúčové slovo.
- *describesProject* – trieda `Project` – výskumný/vývojový projekt.
- *isRelatedTo* – trieda `Event` – udalosť, ktorá je zvyčajne spojená s publikáciou (konferencia,...)
- *author* – trieda `Person` – autor publikácie.
- *topic* – trieda `IndexTerm` – indexovaný výraz opisujúci publikáciu.
- *contains* – trieda `Publication` – opisuje publikáciu, ktorá je časťou rodičovskej publikácie (napr. články v časopise).
- *isPartOf* – trieda `Publication` – je časťou publikácie (inverzná ku *contains*).
- *references* – trieda `Publication` – odkazuje sa na publikáciu.
- *referencedBy* – trieda `Publication` – je odkazovaná publikáciou (inverzná k *references*).
- *similarPublication* – trieda `Publication` – je podobná publikácii.



Obrázok B-6: Vlastnosti triedy `Publication`.

B.3 Model používateľa

Model používateľa vychádza z opisu domény pomocou doménovej ontológie. Dôležité sú ďalšie možnosti použitia modelu ako aj jeho schopnosť zachytiť všetky potrebné

charakteristiky. Tieto môžu byť úzko späté so zvolenou informačnou doménou alebo také, ktoré má význam modelovať aj pri iných informačných doménach, iných aplikáciách a je výhodné ich zdieľať.

Ontológia používateľa sa skladá z troch samostatných ontológií, čím oddeľuje tieto dva druhy charakteristík:

- ontológia *generic-user* – definuje všeobecné charakteristiky používateľa;
- ontológia *job-offer-user* – definuje charakteristiky používateľa viažuce sa na doménovú ontológiu pracovných ponúk;
- ontológia *publication-user* – definuje charakteristiky používateľa viažuce sa na doménovú ontológiu publikácií.

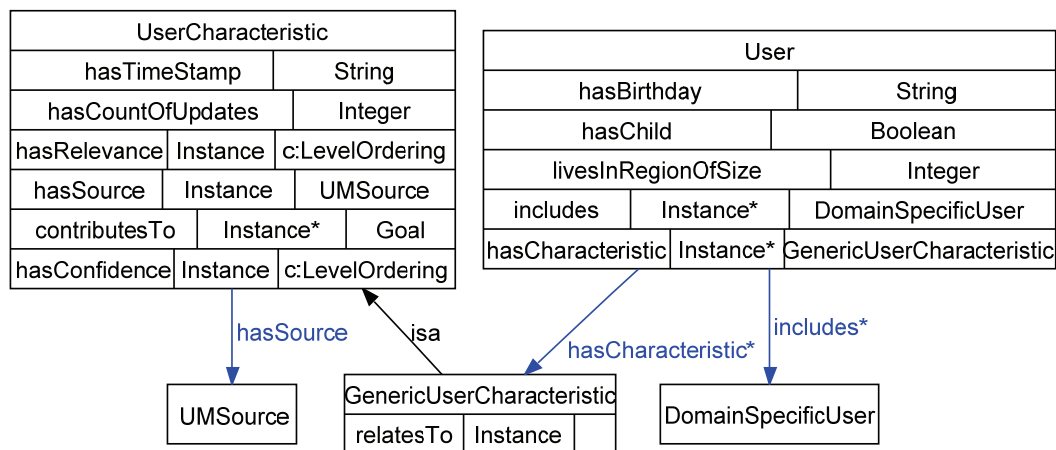
B.3.1 Doménovo nezávislý model

Primárnou triedou tohto modelu je trieda *User* (Obrázok B-7) s týmito dátovými vlastnosťami:

- *hasName* – dátového typu *xsd:string* – meno používateľa;
- *hasBirthday* – dátového typu *xsd:date* – dátum narodenia;
- *hasChild* – dátového typu *xsd:boolean* – či má používateľ aspoň jedno dieťa.
- *livesInRegionOfSize* – dátového typu *xsd:int* – počet obyvateľov regiónu, v ktorom používateľ žije.

Objektové vlastnosti spájajú triedu *DomainIndependentUser* s nasledovnými konceptmi:

- *GenericUserCharacteristic* – doménovo nezávislé charakteristiky
- *DomainSpecificUser* – časti modelu špecifické pre jednotlivé domény



Obrázok B-7: Všeobecné charakteristiky používateľa.

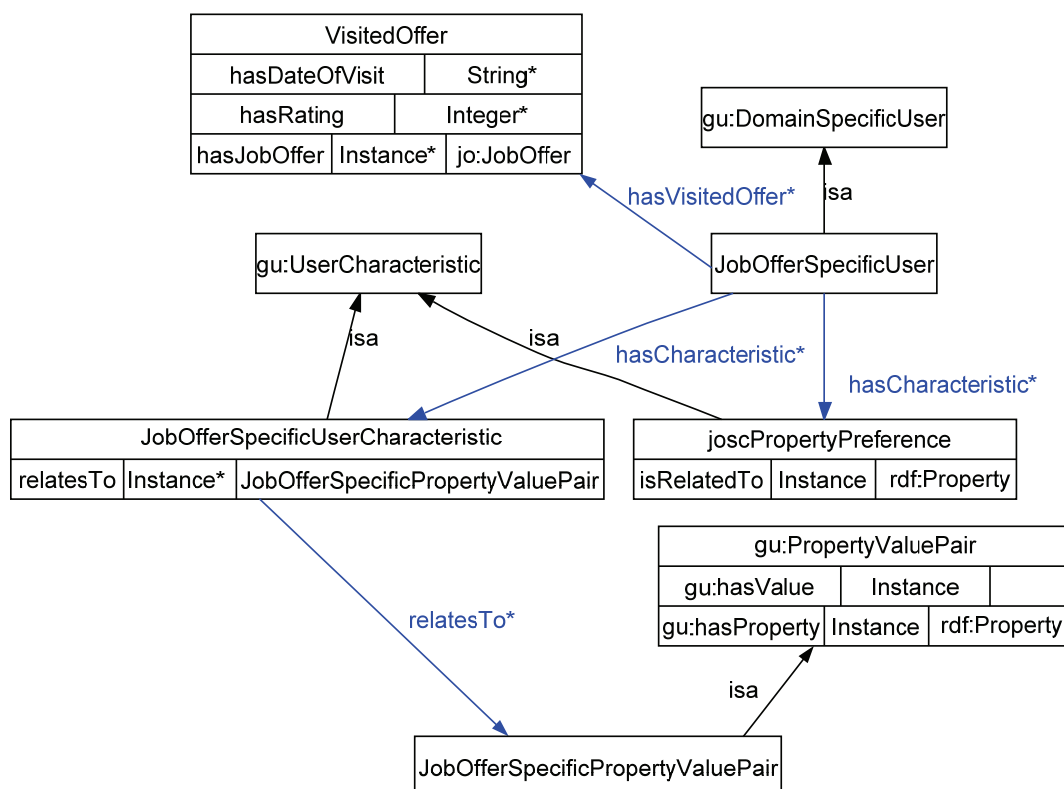
O každej charakteristike zaznamenávame:

- *hasConfidence* – spoľahlivosť,
- *hasRelevance* – relevantnosť,
- *relatesTo* – hodnotu,
- *contributesTo* – cieľ, ku ktorému táto charakteristika prispieva,
- *hasSource* – identifikácia nástroja, ktorý charakteristiku pridal do modelu,
- *hasCountOfUpdates* – počet úprav charakteristiky.

B.3.2 Doménovo závislý model

Vo všeobecnosti môže existovať viacero doménovo závislých modelov používateľa, ktoré sú prepojené s doménovo nezávislým modelom. Z domén projektov NAZOU a MAPEKUS vznikli dva doménovo závislé modely používané v príslušných doménach ponúk a publikácií.

V projekte NAZOU je základnou triedou *JobOfferSpecificUser* (Obrázok B-8), ktorá je podtriedou triedy *DomainSpecificUser* z doménovo nezávislého modelu, čím je zabezpečené prepojenie modelov.



Obrázok B-8: Doménové charakteristiky používateľa v projekte NAZOU.

Objektové vlastnosti spájajú triedu *JobOfferSpecificUser* s nasledovnými konceptmi:

- *VisitedOffer* – predstavuje záznam o zobrazení ponuky používateľovi a jej ohodnotení používateľom.

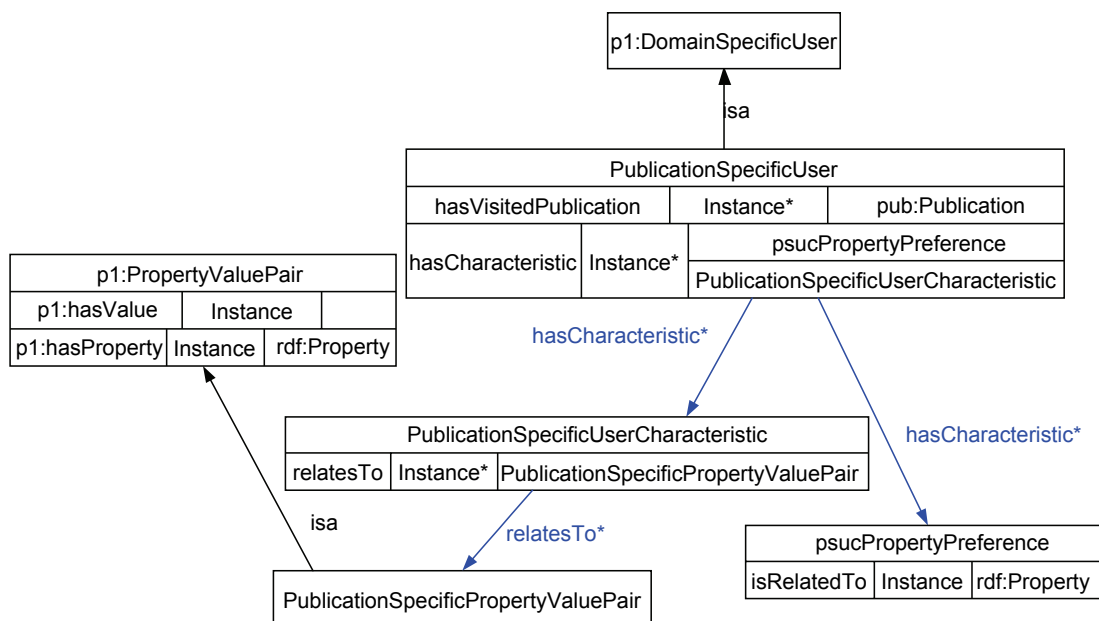
- *JobOfferSpecificUserCharacteristic* – charakteristika orientovaná na hodnoty doménových konceptov reprezentované párom dvojica – hodnota triedy *JobOfferSpecificPropertyValuePair*.
- *joscPropertyPreference* – charakteristika orientovaná na preferencie vlastností realizujúcich konceptualizáciu.

Triedy *JobOfferSpecificUserCharacteristic* a *PropertyPreference* sú podtriedami triedy *UserCharacteristic* z doménovo nezávislého modelu. Tieto triedy majú teda tak isto ako *GenericUserCharacteristic* vlastnosti spoľahlivosť, relevantnosť, hodnotu, cieľ atď.

V projekte MAPEKUS je model používateľa riešený takmer identicky. Základnou triedou je *PublicationSpecificUser* (Obrázok B-9), ktorá je podtriedou triedy *DomainSpecificUser* z doménovo nezávislého modelu, čím je zabezpečené prepojenie modelov.

Objektové vlastnosti spájajú triedu *PublicationSpecificUser* s nasledovnými konceptmi:

- *PublicationrSpecificUserCharacteristic* – charakteristika orientovaná na hodnoty doménových konceptov reprezentované párom dvojica – hodnota triedy *PublicationSpecificPropertyValuePair*.
- *psucPropertyPreference* – charakteristika orientovaná na preferencie vlastností realizujúcich konceptualizáciu.



Obrázok B-9: Doménové charakteristiky používateľa v projekte MAPEKUS.

Príloha C Publikácie vzťahujúce sa k výsledkom diplomovej práce

Personalized navigation in the semantic web

Príspevok prijatý na Doctoral Consortium na konferencii Adaptive Hypermedia 2006 v Dubline.

Tools for User Characteristics Acquisition

Príspevok v spoluautorstve s A. Andrejkom, M. Barlom a M. Bielikovou prijatý na konferenciu DATAKON 2006 v Brne.

Personalized Presentation in Web-Based Information Systems

Príspevok v spoluautorstve s M. Barlom a M. Bielikovou prijatý na konferenciu SOFSEM 2007 v Harachove.

Personalized Navigation in the Semantic Web*

Michal Tvarožek**

Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technologies, Slovak University of Technology,
Ilkovičova 3, 842 16 Bratislava, Slovakia
`tvarozek@fiit.stuba.sk`

Abstract. Effective navigation and information retrieval is difficult and time consuming due to the increasing size of hyperspace. The introduction of the semantic web allows us to enhance traditional search methods with semantic search capabilities that take advantage of machine readable semantic information ideally stored in an ontology. Nevertheless issues concerning the user-friendly construction of search queries and a simple yet effective presentation of search results must still be addressed. The proposed approach takes advantage of adaptive hypermedia in an enhanced faceted browser capable of dynamically adapting the set of available facets with additional support for data retrieval from an ontology and adaptive annotation of search results.

1 Introduction

By definition, navigation is the process of following links and browsing web pages. At present, navigation is necessary because it is not (yet) normally possible to satisfy all user needs on a single web page or on the first visited web page. Generally speaking, user needs can be classified into the following types [1]:

- Informational, when the user seeks information
- Navigational, when the user seeks a starting point for further browsing
- Transactional, when the user wants to perform an action

In order to fulfill either type of need a user typically first enters a query into a search engine to find a list of the most relevant pages. Next, he selects the most promising links and initiates a navigation session that involves the browsing of the selected web pages. If the respective need cannot be satisfied, the user modifies the original query and starts from the beginning.

Several studies have shown that the recursion rate of navigation is roughly 60% [2] and describes the number of repeatedly visited pages to the total number of visited pages. Furthermore, the shift from the use of closed and relatively small information spaces towards large, open and ever growing information spaces is bound to further escalate this problem. Perhaps a good example of such an open information space is the domain of job offers on the Internet.

* This work was supported by Science and Technology Assistance Agency under the contract No. APVT-20-007104 and State programme of research and development “Establishing of Information Society”.

** Supervisor: Professor Mária Bieliková, `bielik@fiit.stuba.sk`

Problems and Issues. We intend to address several contemporary navigation problems with the proposed approach. These include the navigation in open and relatively large information spaces, which is too time consuming and often results in the “lost in hyperspace” syndrome that occurs when users loose track of their position in hyperspace due to insufficient navigation aids or because the hyperspace is “too large”.

While the problems of effective searching and information retrieval in these spaces can be addressed by semantic search [3, 4] this requires the use of more complex queries than full-text search. As a result, new user-friendly ways of query construction and simple yet effective methods for the presentation of search results must be employed. Adaptive hypermedia technologies [5] can be used to make it easy for users to read and understand the results and allow them to quickly choose the most promising ones for further navigation.

2 Research Agenda

2.1 Proposed Approach

We propose the use of a faceted browser¹ as a solution for the aforementioned problems. A faceted browser is a browser that supports faceted navigation that takes advantage of faceted classification defined by a classification ontology that describes important aspects of instances from a domain ontology. The attributes of instances correspond to facets and the values of attributes correspond to facet values. The user can reduce the total number of displayed instances by enabling one or more restrictions defined by facet values thus decreasing the size of the visible information space. Individual facet values can be further combined to form complex restrictions allowing the user to perform more precise queries.

While we are primarily interested in large information spaces with many similar instances represented by a domain ontology with semantic markup (e.g., OWL), the proposed approach should also be applicable to other areas. The advantages of ontologies are twofold. First, it is easier to create a classification ontology from a domain ontology than from unstructured data and it should also be possible to automate this process. Second, reasoning on ontologies allows us to perform “more complex” queries with higher quality results. Several reasoning tools are presently available. Simpler tools are supplied with ontological databases (e.g., Sesame), others are free or commercial such as RacerPro².

Large information spaces and additional data obtained by reasoning on ontologies would result in too much information thus overloading the user with information. To address this issue we propose personalization and user adaptation as means of reducing information overload by focusing on current user needs and goals defined by a user model. To simplify the process of adaptation and more importantly presentation of data, we propose the use of a presentation framework for ontological data based on the Fresnel presentation ontology [6].

¹ Use of Faceted Classification,

<http://www.webdesignpractices.com/navigation/facets.html> (2.2.2006)

² <http://www.racer-systems.com/products/racerpro/index.phtml> (27.3.2006)

✕ <u>USA</u> A		C		◀Miscellaneous, History▶	
Simple Advanced					
	Region B	Compare	♥♥ Name ♥♥ Salary	♥♥ Region ♥♥ Rating	<u>Simple</u> Detailed <u>Salary</u> <u>Experience</u>
All > USA (1 100)		D		C++ Programmer	\$55,000 New Jersey F
➤ New York (550)			☐		A software company is seeking employees for the position of a C++ programmer. Previous ...
➤ New Jersey (300)				Project manager	\$85,000 New York
➤ Washington (100)			☐		We hire project managers for software projects of medium size. Successful candidates will have outstanding communication abilities ...
➤ California (150)					
	Industry		☐		Taxi driver
All (1 100)				The taxi driver union of Seattle seeks new drivers. Candidates with prior experience in the taxi business are preferred. We offer a base salary of \$25,000 p.a. + a performance based bonus. Flexible working hours and one day of paid leave per month.	
➤ IT (500)			☐		
➤ Finance (300)					
➤ Transportation (200)			☐		
	Salary				
All (1 100)				...	
➤ >\$100,000 p.a. (50)					
➤ <\$100,000 p.a. (350)					
➤ <\$50,000 p.a. (250)					
➤ <\$20,000 p.a. (50)					
Matching results: 1 100 Results per page: 3 <u>10</u> <u>25</u> <u>50</u> G					
Page: 1 <u>2</u> <u>3</u> <u>4</u> <u>5</u> <u>6</u> <u>7</u> <u>Next ></u>					

Fig. 1. A sample user interface of an enhanced faceted browser. Area A shows the currently selected restrictions, area B contains the available set of facets (Region, Industry, Salary) and restrictions. Area C contains miscellaneous data, like navigation or query history and additional settings. Area D is used to compare a set of instances while area E allows the user to sort instances, change views or highlighting options. Area F displays individual instances (search results) and depicts instance data enriched with adaptive annotation techniques (emoticons, background color, traffic lights). Area G serves for navigation between different results pages.

2.2 Progress to Date

We implemented a basic version of a faceted browser that supports the use of simple search queries and the browsing of search results (job offer instances). We use an ontological database of job offers, which was developed within the scope of a larger project conducted at the Slovak University of Technology [7].

Based on an initial evaluation of the usability of the browser, we see a strong need for adaptation due to the high number of available facets and facet values with the user's goals and background as the primary sources of adaptation stored in a user ontology. Furthermore, users have specific requirements which translate into the need and/or preference of some facets and facet values over others.

From a user's standpoint, the usability of a tool is not only defined by the array of available functions but also by the usability of its graphical user interface. While this concerns a somewhat different area of research we designed an initial outline of a user interface for an enhanced faceted browser (see Fig. 1).

2.3 Future Research Challenges

We intend to explore the adaptation of the faceted browser based on a user model. This includes the adaptation of facets and facet values and adaptation of search results e.g. by means of adaptive annotation techniques. Furthermore, the visual representation of navigation history in the form of a (hyper)graph appears to have some potential in improving a user's understanding of hyperspace.

Although the primary source of adaptation will be a user model, it appears to be an interesting research prospect to evaluate the usability of the observed user navigation and history data as an additional source of adaptation.

Adaptation of Predefined Facets. The adaptation of predefined facets includes adaptive navigation techniques like reordering, hiding [5] and the presentation of facets and facet values. Also interesting is the possibility to present facets as enumerated lists or as graphs created by OWL visualization tools.

For example, in the case of the job offer ontology, if a user was interested in high paying jobs with specific experience requirements, the facets including these restrictions would be displayed first. Less important facets or restrictions that are assumed to be irrelevant would be either shown later or completely hidden.

Dynamic Facet Generation. Dynamic generation of facets goes one step beyond simple adaptation as described above. Ideally it will be able to create new facets and the respective restrictions at run time based on the knowledge contained in the domain ontology and in the user model. This feature would also improve on the usability of the faceted browser with different domain ontologies which would thus not require extensive manual definition of facets.

Assuming that a facet for experience level requirements was not yet defined, dynamic facet generation would be able to create a new facet definition if experience levels were present in the domain ontology. This new facet would then be displayed together with the original facets in the top part of the facet list.

Advanced Query Mode. The selection of individual restrictions in facets is transformed into a query that is executed on an ontological database. In simple query mode, exactly one restriction per facet can be selected and all facets are combined with the logical AND function resulting in relatively simple queries.

The proposed advanced query mode allows users to create and execute more complex queries with multiple restrictions per facet thus resulting in a more precise description of the desired instances. We also plan support for additional facet combination functions, such as the logical OR function or braces. Thus if a user wanted a job either in the USA or in Canada, he would select both USA and Canada in one facet – “Region” and combine them with the OR function.

Presentation and Processing. The result of a search performed by a faceted browser is normally an unordered list of instances that satisfy the search criteria. This however is not ideal for effective evaluation of the search results by the user. Additional means for processing of the search results are necessary to improve usability. We intend to add simple ordering of search results as well as support for

more complex external sorting tools and several adaptive views with different levels of detail [8]. The possibility to compare the attributes of the selected instances (search results) also seems to be good for improving usability.

While sorting only displays the rating of instances with regard to one attribute, a user will often be interested in ratings based on several attributes simultaneously. The use of adaptive annotation techniques [5] to present these attributes appears to be a promising direction for further research.

For example, in the domain of job offers background color can indicate how well a user satisfies the requirements of the employer, an emoticon can indicate how well a job offer satisfies user criteria, while the job offers are ordered in descending order based on the offered salary. A traffic light signal can indicate a composite suitability rating of a job offer based on a heuristic function or the overall rating of the employing company by previous applicants.

3 Summary

We proposed an enhanced faceted browser as a solution for several navigation and search related problems. The usability evaluation of a basic faceted browser, confirmed the need for its adaptation due to the high number of available facets and facet values. Further work will include enhancements to facets – their adaptive reordering, hiding and dynamic generation and support for adaptive annotation based on a user model and domain ontology. We will also continuously evaluate the usability of the browser and the viability of the proposed concept based on feedback from sample users in the domain of job offers [7].

References

1. Broder, A.: A taxonomy of web search. SIGIR Forum (2002)
2. Lavene, M., Wheeldon, R.: Navigating the World-Wide-Web. In Lavene, M., Poulouvassilis, A., eds.: Web Dynamics, Springer (2003)
3. Guha, R., McCool, R., Miller, E.: Semantic Search. In: The 12th International Conference on World Wide Web. (2003) 700–709
4. Zhang, L., Yu, Y., Zhou, J., Lin, C.X., Yang, Y.: An Enhanced Model for Searching in Semantic Portals. In: WWW 2005, ACM Press (2005) 453–462
5. Brusilovsky, P.: Adaptive Hypermedia. In Kobsa, A., ed.: User Modeling and User-Adapted Interaction, Ten Year Anniversary Issue. (2001) 87–110
6. Bieliková, M., Grlický, V., Kuruc, J.: Framework for presentation of information represented by an ontology. In Vojtáš, P., ed.: ITAT 2005 – Workshop on Theory and Practice of Information Technologies, Račkova dolina (2005) 325–334
7. Návrát, P., Bieliková, M., Rozinajová, V.: Methods and Tools for Acquiring and Presenting Information and Knowledge in the Web. In: International Conference on Computer Systems and Technologies – CompSysTech' 2005, Varna, Bulgaria (2005)
8. Domingue, J., Dzbor, M., Motta, E.: Magpie: Supporting Browsing and Navigation on the Semantic Web. In: Intelligent User Interfaces. (2004) 191–197

Softvérové nástroje pre získavanie charakteristík používateľa

Anton ANDREJKO, Michal BARLA, Mária BIELIKOVÁ, Michal TVAROŽEK

*Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií
Slovenská technická univerzita v Bratislave
{andrejko, barla, bieliko, tvarozek}@fiit.stuba.sk*

Abstrakt. Prispôsobovanie sa charakteristikám používateľov sa stáva významnou vlastnosťou súčasných webových informačných systémov. Bez prispôsobovania prezentácie môže byť veľmi ťažké a dokonca často nemožné používať tieto systémy na efektívnu prácu s informáciami. Na prispôsobovanie je potrebné poznať charakteristiky jednotlivých používateľov. V príspevku opisujeme získavanie charakteristík používateľa pomocou spolupracujúcich softvérových nástrojov, kde každý nástroj vykonáva špecifickú časť tohto procesu. Charakteristiky získavame na základe monitorovania správania sa používateľa, čo minimalizuje mieru potrebného zapojenia používateľa do procesu tvorby modelu používateľa. Opisujeme štyri nástroje s dôrazom na ich vzájomné prepojenia, ktoré vedú k získavaniu charakteristík používateľa.

Kľúčové slová: model používateľa, objavovanie charakteristík používateľa, vzory správania, zaznamenávanie akcií používateľa.

1 Úvod

Dôležitou súčasťou adaptívnych webových systémov je modelovanie používateľa, ktoré zahŕňa najmä získanie a vyhodnotenie charakteristík používateľa tak, aby sa mohli použiť pri prispôbovaní. S nárastom množstva dostupných informácií sa prispôbovanie zamerané na konkrétneho používateľa – personalizácia stáva nevyhnutnou súčasťou súčasných webových informačných systémov. Dopyt po generických riešeniach adaptívnych systémov vyúsťuje do potreby metód a nástrojov, ktoré umožňujú automatické získavanie a vyhodnocovanie charakteristík používateľa.

Vytvorenie vhodného modelu používateľa je zložitý proces, ktorý zahŕňa viacero úloh. K ich realizácii môžeme pristúpiť vytvorením sady softvérových nástrojov, ktoré sa špecializujú na jednotlivé úlohy. Príkladom môže byť nástroj, ktorý spája informácie získané od používateľa počas registrácie z príslušných formulárov s informáciami získanými z iných nástrojov do modelu používateľa, ktorého cieľom je reprezentovať dobre odhadnuté charakteristiky používateľa.

Cieľom nášho výskumu je tvorba modelu používateľa na odporúčanie vhodných ponúk z konkrétnej domény (napr. pracovné alebo cestovné ponuky), ktorý tvorí základ pre personalizovanú navigáciu a filtrovanie obsahu pre jednotlivých používateľov. Na vyhodnotenie výsledkov nášho výskumu používame doménový model a model

používateľa vytvorený v rámci výskumného projektu „Nástroje pre získavanie, organizovanie a udržovanie znalostí v prostredí heterogénnych informačných zdrojov“ [6], ktorý sa orientuje na doménu pracovných ponúk.

V tomto príspevku prezentujeme niekoľko softvérových nástrojov, ktoré podporujú proces získavania charakteristík používateľa spolu s ich prepojeniami. V časti 2 prezentujeme proces vytvárania modelu používateľa a možné zdroje informácií, ktoré do procesu vstupujú. Definujeme sadu nástrojov, ktoré realizujú potrebné úlohy. V časti 3 sa venujeme špecifikám fázy zberu dát a opisujeme príslušné nástroje tejto fázy (*Click* – *Client Side Action Recorder* a *SemanticLog*). V časti 4 opisujeme spracovanie zozbieraných informácií do podoby odhadov charakteristík používateľa a ich realizáciu nástrojom *LogAnalyzer*. Nakoniec sumarizujeme príspevok a závery z našej práce prezentovanej v tomto príspevku.

2 Proces vytvárania modelu používateľa

Tvorba modelu používateľa pozostáva z dvoch hlavných fáz. Najprv sa získajú dáta o správaní používateľa, ktoré sa ďalej analyzujú a interpretujú s cieľom objavenia relevantných charakteristík používateľa. Takto objavené charakteristiky sa použijú pri personalizácii. Existuje viacero spôsobov získavania dát o používateľoch:

- formuláre, ktoré umožňujú používateľom priamo zadať požadované charakteristiky alebo odpovedať na špecifické otázky, ktoré charakteristiky odhaľujú nepriamo;
- explicitná spätná väzba používateľa na zobrazený obsah; spätná väzba môže byť buď pozitívna alebo negatívna;
- dokumenty dodané používateľom (napr. CV), z ktorých sa dajú extrahovať relevantné charakteristiky;
- záznam prístupov používateľa k zdrojom; používa sa na odvodenie stupňa znalosti používateľa o prezentovaných informáciách;
- záznam správania používateľa v systéme, ktorý zahŕňa implicitnú spätnú väzbu používateľa na zobrazený obsah.

Keďže je vhodné minimalizovať mieru spoluúčasti používateľa na procese vytvárania modelu používateľa, sústredíme sa na modelovanie používateľa založené na pozorovaní správania používateľa a automatickom odvodení jeho charakteristík. Identifikovali sme sadu softvérových nástrojov, ktoré pokrývajú obidve fázy procesu. V tomto príspevku opisujeme nástroje *Click* a *SemanticLog*, ktoré sú určené pre fázu získavania dát a nástroj *LogAnalyzer*, ktorý je zameraný na objavovanie charakteristík používateľa v získaných dátach.

3 Zber dát

Existujú dva hlavné prístupy monitorovania akcií používateľa: monitorovanie vykonávané na strane servera a monitorovanie na klientskej strane systému, pričom sa môže použiť aj kombinácia oboch prístupov.

Monitorovanie na strane servera sleduje prístupy používateľa ku zdrojom. Hlavná nevýhoda tohto prístupu je v tom, že neumožňuje zaznamenávanie niektorých typov interakcie používateľa so systémom (napr. interakcia s prvkami formulára) a neposkytuje presné informácie o čase zobrazenia informácií. Monitorovanie na strane servera závisí od správania webových prehliadačov, ktoré zvyčajne nežiadajú znovu od servera už navštívené stránky, ale použijú namiesto toho kópiu uloženú vo vyrovnávacej pamäti. Systém teda nemá žiadne informácie o dobe, ktorú používateľ strávil prezeraním určitej stránky. Väčšina v súčasnosti najrozšírenejších webových prehliadačov nerešpektuje direktívy HTTP protokolu, ktoré zakazujú používanie lokálnej vyrovnávacej pamäte, takže ich použitie nerieši uvedený problém. Takisto treba vziať do úvahy znížený komfort používania systému bez vyrovnávacej pamäte. Aj v [8] autori uvádzajú závery, že pre získanie presných záznamov o interakcii používateľa so systémom treba využiť monitorovanie na strane klienta. Aj napriek vyššie uvedeným problémom je prístup s využitím záznamov servera postačujúci pre veľa adaptívnych systémov. Napr. systém AHA! (Adaptive Hypermedia for All, `aha.win.tue.nl/`) tieto záznamy využíva pre sledovanie toho, aké dokumenty sa už zobrazili používateľovi [1].

Monitorovanie na strane klienta vytvára podrobný záznam akcií používateľa s presnými časovými pečiatkami. Monitorovanie sa môže vykonávať špeciálnou klientskou aplikáciou (napr. *User Action Recorder* v [10]) alebo použitím klientských webových technológií akými sú JavaScript alebo Java applety. Pretože považujeme prvý zmienený prístup za pomerne invazívny a málo flexibilný, sústreďujeme sa druhý prístup. JavaScript a Java applety v súčasnosti podporujú všetky významné webové prehliadače na hlavných platformách. Možnou nevýhodou je, že nie každý používateľ akceptuje takéto podrobné monitorovanie a niektorí môžu vo svojich prehliadačoch zablokovať vykonávanie vložených skriptov. Napokon môže nastať aj prípad, keď používateľ nemá na počítači nainštalovaný potrebný softvér.

Existuje viacero nástrojov, ktoré využívajú JavaScript na monitorovanie na strane klienta ako sú WebVip (Web Variable Instrumenter Program, `zing.ncsl.nist.gov/webTools/WebVIP/overview.html`) alebo WET (Web Event-logging Tool, [3]). Obe dva nástroje sú primárne určené pre účely vyhodnocovania použiteľnosti webových stránok. Nevýhodou nástroja WebVIP je vytvorenie kópie celého webového sídla, do ktorej nástroj do jednotlivých stránok pridáva identifikátory a obsluhy udalostí na HTML odkazoch. Nástroj sa preto nedá použiť pri dynamicky vytváraných webových stránkach, keďže mu nedokážeme dodať kompletne webové sídlo. Ďalšou nevýhodou nástroja WebVIP je jeho obmedzenie na štandardné HTML odkazy ignorujúce ďalšie HTML elementy stránok. Aj keď nástroj WET rieši uvedené problémy, je príliš naviazaný na doménu vyhodnocovania stránok, čo znemožňuje jeho použitie ako všeobecného nástroja na zaznamenávanie akcií používateľa. Nástroj vytvára nad stránkou ďalšiu vrstvu, ktorá obsahuje tlačidlá na ovládanie procesu zaznamenávania, pričom dáta sa posielajú na server až po stlačení tlačidla stop.

Z uvedenej analýzy vyplýva, že pri použití monitorovania na strane klienta sa vystavujeme riziku, že používateľ nebude súhlasiť s podrobným monitorovaním, čo by znamenalo, že nezískame žiadne dáta. To je silný argument proti použitiu čistého monitorovania na strane klienta. Monitorovanie na strane servera je z tohto pohľadu spoľahlivejšie, keďže sa mu vždy podarí získať nejaké dáta. Nevýhodou je, že môžeme stratiť časový aspekt zaznamenaných akcií. Naš prístup je založený na myšlienke

skombinovania oboch prístupov – naše nástroje extrahujú maximum informácií na strane servera a používajú záznam z klientskeho monitoringu ako zdroj prídavných, časovo presných informácií o aktivitách používateľa.

3.1 Nástroj na zaznamenávanie akcií používateľa na strane klienta

Click je nástroj, ktorý realizuje monitorovanie používateľa na strane klienta a zachytáva udalosti, ktoré vyvoláva webový prehliadač počas interakcie používateľa s elementmi zobrazovaných stránok:

- *Load*: zobrazenie stránky používateľovi,
- *Unload*: keď používateľ stránku opustí,
- *Click*: keď používateľ nasleduje niektorý odkaz na stránke,
- *Mouseover*: keď používateľ nasmeruje kurzor na aktívny element stránky,
- *Mouseout*: keď používateľ odstráni kurzor z aktívneho elementu stránky.

Uvažujeme aj zachytávanie ďalších udalostí, ktoré sú špecifické pre stranu klienta a nie sú dostupné na serveri. Medzi také udalosti patrí udalosť *change*, ktorá sa vyvolá po zmene obsahu prvku formulára. Sekvencia týchto udalostí informuje o poradí, v ktorom používateľ vyplňal formulár. Iným príkladom je udalosť *scroll*, ktorá sa vyvolá, ak sa používateľ posunie po stránke.

Click zbiera nasledovné údaje o každej zachytenej udalosti:

- typ udalosti,
- čas spustenia udalosti a
- kontext zachytenej udalosti, napr. ktorý odkaz používateľ nasledoval.

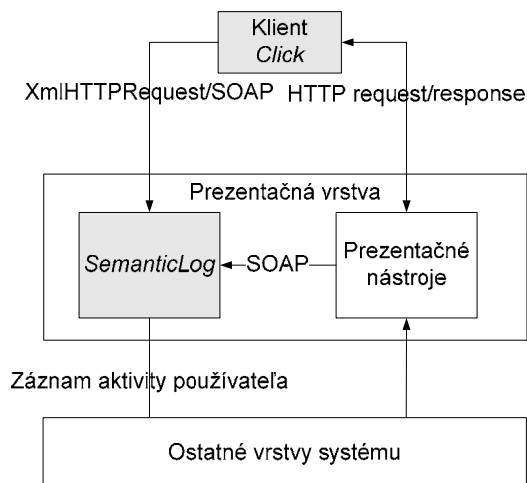
Komunikácia so serverom je realizovaná v dávkach N zachytených udalostí. Ak je N nastavené na jednu, *Click* posielala na server záznam po každej zachytenej udalosti.

Click je implementovaný s využitím JavaScript technológie, ktorá nekladie žiadne špeciálne požiadavky na klienta, keďže je podporovaná väčšinou súčasných webových prehliadačov. JavaScript poskytuje natívny prístup k objektovému modelu HTML dokumentu (DOM), takže *Click* má prístup k štruktúre aj obsahu stránky.

Zachytávanie udalostí je založené na špecifikácii *DOM Level 2 Events* (www.w3.org/TR/DOM-Level-2-Events), ktorá definuje obsluhovače udalostí dynamicky priradené k elementom objektového modelu. To umožňuje jednoducho integrovať nástroj do existujúcich ako aj dynamicky generovaných HTML stránok pridaním odkazu na príslušný skript v hlavičke HTML dokumentu.

Komunikácia so serverom je asynchrónna s použitím objektu *XmlHttpRequest* (pozri obr. 1), ktorý predstavuje jadro technológie AJAX. Nástroj *Click* posielala záznamy zachytených udalostí na SOAP rozhranie nástroja *SemanticLog*. Ten agreguje získané dáta s informáciami získanými z prezentačných nástrojov, ktoré poznajú sémantiku zobrazeného obsahu. Nakoniec sa záznam obohatený o sémantiku zašle na ďalšie spracovanie vnútornými vrstvami systému.

Serverovú časť monitorovacieho systému sme realizovali ako samostatný nástroj, ktorý prijíma záznamy udalostí vytvárané nástrojom *Click* cez SOAP rozhranie webovej služby. Pre volanie metódy webovej služby z prostredia JavaScript používame knižnicu od IBM (Call SOAP Web services with Ajax).



Obr. 1. Štruktúra časti pre získavanie dát.

3.2 Nástroj na zaznamenávanie udalostí so znalosťou sémantiky

SemanticLog je generický nástroj na vytváranie serverového záznamu, ktorý vykonáva dve hlavné úlohy. Prvou je zaznamenávanie udalostí dodaných od nástrojov prezentačnej vrstvy, druhou je prijímanie a zaznamenávanie udalostí od nástrojov vykonávajúcich monitorovanie na strane klienta ako napr. nástroj *Click*.

Nástroj je implementovaný ako webová služba s verejným rozhraním, ktoré umožňuje ostatným nástrojom posilať dáta, ktoré sa spájajú do jedného spoločného záznamu. Napr., fazetový prehliadač, ktorý slúži v projekte [6] ako prezentačný nástroj, môže zaznamenávať akcie používateľa pomocou SOAP protokolu. Keďže nástroj pozná sémantiku vykonaných akcií a “rozumie” obsahu, ktorý produkuje, dokážeme v konečnom dôsledku vytvoriť podrobnejší a zároveň sémanticky obohatený záznam v porovnaní so štandardným záznamom webového servera.

Napr. ak používateľ kliknutím vyberie ohraničenie informačného priestoru založené na mieste výkonu práce zo zoznamu možných miest, nástroj nezaznamená *URL* odkazu, na ktorý používateľ klikol, ale význam vykonanej akcie spolu s *URI* zvolenej inštalácie tak, ako je uložená v ontológii.

Príklad udalosti zaznamenananej nástrojom *SemanticLog*:

```
<event>
  <datetime>2006-04-23 00:54:06</datetime>
  <type>SelectRestriction</type>
  <facet>Region</facet>
  <value>http://job-offers/region#US_AZ</value>
</event>
```

Navyše, nástroje pre monitorovanie na strane klienta, ako napr. *Click*, môžu tiež využiť SOAP rozhranie a dodať udalosti z klientskej strany interakcie ako napr. použitie tlačidla „späť” alebo umiestnenie kurzora nad vybranými položkami. Tieto informácie možno asociovať s udalosťami získanými na strane servera a vytvoriť tak podrobný záznam interakcie používateľa so systémom.

Hlavnou výhodou navrhnutého prístupu je zachovanie sémantiky akcií, ktoré vykonáva používateľ ako aj sémantiky zobrazených stránok pre ďalšie spracovanie nástrojmi pre analýzu správania používateľa.

Nástroje, ktoré spracúvajú záznamy potrebujú mať informácie o zaznamenaných URL adresách a asociovaných parametroch tak, aby „pochopili“ sémantiku obsiahnutú v záznamoch, čo vedie k silnej zviazanosti jednotlivých nástrojov. Pridanie ďalšieho prezentačného nástroja alebo prípadná zmena už existujúceho nástroja vedú vždy k (väčším) zmenám nástrojov analyzujúcich záznamy.

Nástroj *SemanticLog* rieši problém silnej zviazanosti prezentačných nástrojov s nástrojmi pre analýzu poskytnutím spoločnej reprezentácie udalostí získaných od všetkých prezentačných nástrojov. Ďalšou výhodou použitia nástroja *SemanticLog* oproti tradičnému zaznamenávaniu URL adries je teda aj to, že nástroje pre analýzu záznamov (napr. *LogAnalyzer*) nepotrebujú rozumieť špecifickým parametrom jednotlivých prezentačných nástrojov a sú teda len voľne zviazané.

4 Objavovanie charakteristík používateľa

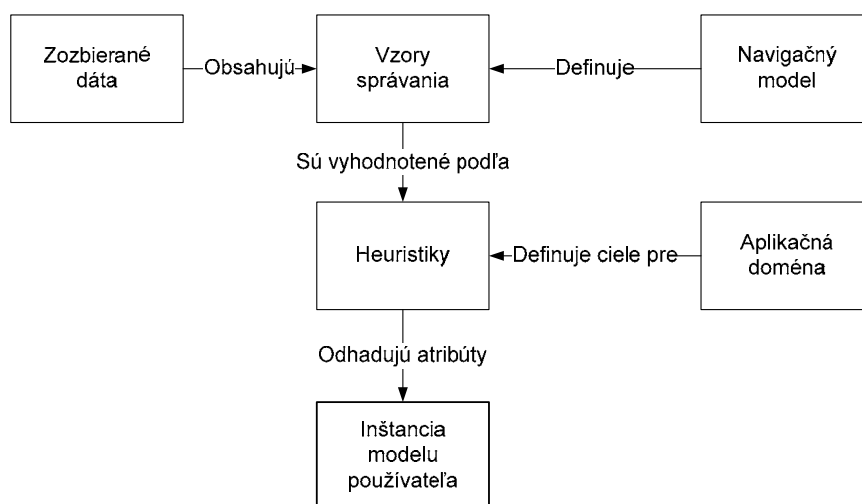
Po fáze zberu dát treba interpretovať ich význam, ktorý nepriamo súvisí so zachyteným správaním používateľa – buď zo správania vyplýva alebo je s ním asociovaný [4] (napr. identifikovať ciele alebo odhadnúť znalosti používateľa o určitých konceptoch). Niektoré charakteristiky vieme odvodiť zo spôsobu navigácie používateľa vo webovom sídle. Odvodenie predpokladá použitie takého navigačného modelu, ktorý umožňuje používateľovi relatívne slobodný pohyb medzi jednotlivými stránkami tak, aby sa v správaní prejavili charakteristiky používateľa.

Iný spôsob odhadu charakteristík používateľa predstavuje analýza reakcií používateľa na zobrazený obsah, ktorá vyhodnocuje implicitnú spätnú väzbu vyplývajúcu z akcií používateľa [7].

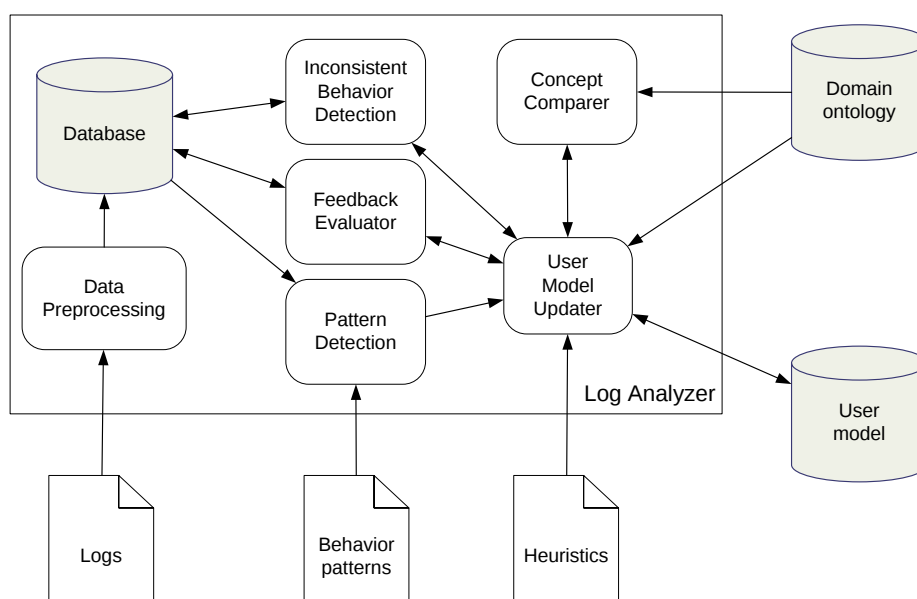
V oboch prípadoch hľadáme preddefinované vzory správania v postupnosti akcií používateľa vo webovom sídle. Následne využijeme heuristiku asociovanú s identifikovaným vzorom a vykonáme príslušnú zmenu v modeli používateľa. Heuristiky sú definované s ohľadom na typické ciele používateľa v príslušnej aplikačnej doméne. Obr. 2 zobrazuje zdroje informácií na vytvorenie alebo aktualizovanie inštancie modelu používateľa.

Na analýzu zozbieraných dát s využitím vyššie uvedených princípov sme navrhli softvérový nástroj *LogAnalyzer*, ktorý na základe analýzy zozbieraných dát upravuje model používateľa.

Činnosť nástroja je rozdelená do troch samostatných fáz: (i) predspracovanie získaných dát, (ii) objavovanie charakteristík používateľa a (iii) aktualizácia modelu používateľa. Architektúru nástroja *LogAnalyzer* sme navrhli s ohľadom na tieto tri fázy (pozri Obr. 3). Prvú fázu realizuje modul *DataPreprocessing*. (predspracovanie dát) Druhú fázu, objavovanie charakteristík používateľa, je zabezpečujú moduly *PatternDetection* (odhaľovanie vzorov), *FeedbackEvaluator* (vyhodnocovanie spätnej väzby) a *InconsistentBehaviorDetection* (odhaľovanie nekonzistentného správania). Poslednú fázu – aktualizáciu modelu používateľa, realizuje modul *UserModelUpdater* (aktualizácia modelu používateľa), ktorý používa modul *ConceptComparer* (porovnávanie konceptov).



Obr. 2. Zdroje informácií pre aktualizovanie inštancie modelu používateľa.



Obr. 3. Architektúra nástroja *LogAnalyzer*.

Predspracovanie dát

Modul *DataPreprocessing* identifikuje v získaných záznamoch jednotlivé sedenia používateľa a uloží dáta v podobe vhodnej pre ďalšiu analýzu. V tejto fáze sa vykonávajú aj ďalšie transformácie dát, ktoré odstránia identické, po sebe idúce akcie (napr. keď používateľ kvôli pomalejšej odozve systému zopakuje vykonanú akciu).

Dôležitou úlohou tohto modulu je úprava hodnôt, ktoré predstavujú dobu čítania stránky. Zvyčajne je táto doba vo vzťahu so záujmom používateľa o obsah danej stránky, avšak ak je extrémne dlhá, môžeme predpokladať, že používateľ prestal pracovať so systémom a nevenuje sa zobrazenému obsahu [9]. Preto čas upravujeme do primeraných intervalov.

Odhaľovanie nekonzistentného správania

Modul *InconsistentBehaviorDetection* realizuje metódu dolovania sekvenčných vzorov nad predchádzajúcimi sedeniami používateľa. Získané vzory pokladáme za typické vzory správania používateľa. Aktuálne sedenie porovnáme so získanými vzormi a zisťujeme tak, či je konzistentné s predchádzajúcimi sedeniami. Podobný prístup je použitý aj v [5], kde sa dolovanie sekvenčných vzorov používa na objavovanie zaujímavých navigačných ciest v adaptívnych výučbových systémoch.

Odhaľovanie vzorov

Modul *PatternDetection* zisťuje výskyt preddefinovaných vzorov v spracovaných záznamoch. Odhaľovanie vzorov je založené na použitých dátových štruktúrach, v našom prípade od dátovej štruktúry *suffix tree* (komprimovaný *trie* [2]).

Vyhodnocovanie spätnej väzby

Modul *FeedbackEvaluator* vyhľadáva vzory implicitnej spätnej väzby [7] v získaných dátach a odhaduje používateľove hodnotenie zobrazeného obsahu.

Porovnávanie konceptov

Informácia o ohodnotení zobrazených informácií reprezentovaných konceptmi na odhad charakteristík používateľa nepostačuje. Dôležité sú dôvody, prečo je konkrétne ohodnotenie nízke alebo vysoké. Keďže sa toto ohodnotenie pri rôznych konceptoch líši, je vhodné poznať vzťahy medzi jednotlivými konceptmi. Túto úlohu realizuje modul *ConceptComparer*, ktorý porovnáva koncepty reprezentované ontologickými inštanciami a hľadá ich spoločné a rozdielne aspekty (mieru podobnosti).

V prípade domény pracovných ponúk môžeme uviesť ako príklad dve pracovné ponuky, ktoré majú veľmi rozdielne ohodnotenie. Ak sú tieto dve ponuky takmer identické a líšia sa len v mieste výkonu práce, vieme z rozdielného ohodnotenia odvodit', že je miesto výkonu práce pre používateľa dôležité. Navyše vieme povedať, ktorá hodnota je pre používateľa vhodná a ktorá nevyhovuje.

Modul *ConceptComparer* počas porovnávania skúma vzťahy porovnávaných inšancií na základe použitej taxonómie tried. Hľadá spoločnú nadtriedu týchto inšancií – čím viac krokov treba na nájdenie tejto nadtriedy, tým menej sú si porovnávané inšancie podobné. Tento modul tiež analyzuje dátové a objektové atribúty porovnávaných inšancií. Pri porovnávaní dátových atribútov (najmä reťazcov) nepostačuje jednoduché porovnanie zhody – treba vyhodnotiť zhodnosť textov na sémantickej úrovni. Metóda takéhoto vyhodnocovania je navrhnutá v [11].

Pri objektových atribútoch sa algoritmus vykonáva rekurzívne na príslušné asociované inšancie. Čiastkové výsledky sú agregované s použitím váh jednotlivých atribútov, ktoré môžu byť preddefinované alebo uložené v modeli používateľa.

Aktualizácia modelu používateľa

Modul *UserModelUpdater* agreguje výsledky ostatných modulov a vykonáva aktualizáciu inštancie modelu používateľa podľa preddefinovaných heuristík.

Nižšie uvádzame príklad jednoduchej heuristiky v doméne pracovných ponúk:

„Ak si používateľ zobrazil detaily o aspoň ‘dostatočnom počte’ ponúk z odvetvia *A* (napr. zdravotníctvo alebo IT), zvýš relevantnosť tohto odvetvia v modeli ideálnej pracovnej ponuky uloženej v modeli používateľa.“

Iná heuristika, ktorá predpokladá navigáciu v informačnom priestore pomocou sémantického fazetového prehliadača [12] je:

„Ak si používateľ hneď po začiatku sedenia vybral ohraničenie *X*, zvýš relevantnosť charakteristiky spojennej s týmto ohraničením.“

5 Záver

V príspevku sme opísali výskum v oblasti automatizovaného získavania charakteristík používateľa, ktoré je založené na sledovaní používateľa. Definovali sme dvojfázový proces, ktorý postupuje od správania používateľa v rámci webového informačného systému k odhadu charakteristík používateľa v jeho modeli. Každá fáza procesu je pokrytá softvérovými nástrojmi, ktoré vykonávajú jednotlivé definované úlohy.

Prvá fáza – *zber dát* produkuje detailný záznam aktivít používateľa. Navrhli sme nástroje *Click* a *SemanticLog* spolu s ich vzájomným prepojením a napojením na zvyšok webového informačného systému. Nástroje vykonávajú monitorovanie používateľa na strane servera a klienta a vytvárajú spoľahlivý záznam aktivít používateľa v časových súvislostiach so zachovaním sémantiky jednotlivých udalostí.

V druhej fáze procesu, fáze objavovania charakteristík používateľa, sme navrhli nástroj *LogAnalyzer*, ktorý spracúva záznam z predchádzajúcej fázy a analyzuje ho z viacerých hľadísk (navigácia, implicitná spätná väzba, konzistencia správania). Výsledky čiastočných analýz kombinuje s heuristikami a upravuje model používateľa. Modul *ConceptComparer* nástroja *LogAnalyzer* by mohol byť definovaný aj ako samostatný nástroj, ktorý poskytuje službu porovnávania ontologických inštancií pre rôzne časti adaptívneho systému.

Jednotlivé nástroje realizujeme a experimentálne overujeme v rámci výskumného projektu v doméne pracovných príležitostí. V budúcnosti plánujeme tiež ich overenie v doméne vedeckých publikácií.

PodĎakovanie

Táto práca vznikla za čiastočnej podpory Agentúry na podporu vedy a techniky (APVT-20-007104), vedeckej grantovej agentúry VEGA (VG1/3102/06) a štátnym programom pre výskum a vývoj „Budovanie informačnej spoločnosti“ (1025/04).

Literatúra

1. De Bra, P., Calvi, L.: AHA: A generic adaptive hypermedia system. In: Proc. of 2nd Workshop on Adaptive Hypertext. Pittsburgh, USA (1998) 5–12.

2. Dunham, M.: *Data Mining: Introductory and Advanced Topics*. Prentice Hall PTR, Upper Saddle River, NJ, USA (2002).
3. Etgen, M., Cantor, J.: What does getting WET (Web Event-logging Tool) mean for web usability? In: *Proc. 5th Conf. on Human Factors & the Web*. (1999).
4. Judd, T., Kennedy, G.: *Making sense of audit trail data*. *Australasian Journal of Educational Technology* 20(1) (2004) 18–32.
5. Krištofič, A., Bieliková, M.: Improving Adaptation in Web-Based Educational Hypermedia by means of Knowledge Discovery. In *Proc. of 16th ACM Conf. on Hypertext and Hypermedia, HT'05*, ACM Press (2005) 184–192.
6. Návrát, P., Bieliková, M., Rozinajová, V.: Methods and tools for acquiring and presenting information and knowledge in the web. In: *Proc. of Int. Conf. on Computer Systems and Technologies, CompSysTech'05*, Varna, Bulgaria (2005).
7. Oard, D.W., Kim, J.: Implicit feedback for recommender systems. In *Proc. of AAAI Workshop on Recommender Systems*, Madison, WI, USA (1998) 80–82.
8. Paganelli, L., Paterno, F.: Intelligent analysis of user interactions with web applications. In *Proc. of the 7th Int. Conf. on Intelligent User Interfaces, IUI'02*, New York, NY, USA, ACM Press (2002) 111–118.
9. Rafter, R., Smyth, B.: Passive profiling from server logs in an online recruitment environment. In *Proc. of the IJCAI Workshop on Intelligent Techniques for Web Personalisation, ITWP'01*, Seattle, Washington, USA (2001) 35–41.
10. Thomas, R., Kennedy, G., Draper, S., Mancy, R., Crease, M., Evans, H., Gray, P.: Generic usage monitoring of programming students. In Crisp, G., Thiele, D., Scholten, I., Barker, S., Baron, J., eds.: *Proc. of 20th Annual Conf. of the Australasian Society for Computers in Learning in Tertiary Education, ASCILITE'03*, Adelaide, Australia (2003) 715–719.
11. Tury, M., Bieliková, M.: An Approach to Detection Ontology Changes. In *Proc. of 1st Int. Workshop on Adaptation and Evolution in Web Systems Engineering, AEWSE'06 at ICWE 2006*, ACM, Palo Alto, CA, USA (2006).
12. Tvarožek, M.: Personalized navigation in the semantic web. In V. Wade, H. Ashman, and B. Smyth, editors, *Proc. of 4th Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, AH'06*, Springer, LNCS 4018, Dublin, Ireland, (2006) 467–471.

Annotation

Software Tools for User Characteristics Acquisition

Nowadays, many web-based information systems need the ability to adapt to user characteristics. Otherwise it would be difficult or even impossible to use them efficiently. This paper describes an approach to user characteristics acquisition performed by a set of cooperating software tools where each tool performs an individual task of the characteristics acquisition process. Characteristics are acquired based on user behavior, what minimizes the amount of necessary user involvement. Four tools are described with stress on their interconnections in order to provide effective user characteristics acquisition.

Personalized Presentation in Web-Based Information Systems

Michal Tvarožek, Michal Barla, and Mária Bielíková

Institute of Informatics and Software Engineering, Faculty of Informatics
and Information Technologies, Slovak University of Technology
Ilkovičova 3, 842 16 Bratislava, Slovakia
{Name.Surname}@fiit.stuba.sk

Abstract. Large information spaces and complex functionality of contemporary systems together with the advent of the Semantic Web are big challenges for the design of simple yet powerful user interfaces. The complex nature of systems thus requires the use of several presentation, personalization and user modeling methods and techniques to address the growing needs and requirements of both users and systems. We present a design of an architecture for the personalized presentation layer of a web-based information system that employs a set of interconnected software components implemented as autonomous software tools for presentation, personalization and user modeling to support features such as navigation support and different views on the presented data, acquisition and evaluation of user characteristics and user adaptation and personalization. The feasibility of our design is evaluated by its realization in the labor market application domain.

1 Introduction

The web contains vast amounts of information in its visible “surface” part and in the “deep web”. These include both billions of publicly accessible web pages and large searchable databases. Moreover, contemporary web-based information systems (WIS) need to deal with increasing amounts of data while new challenges are introduced by emerging technologies, such as the vision of the Semantic Web.

Furthermore, the enormous amount of available information provided by content-oriented WIS and the overall complexity of applications resulted in the need for adaptation of their interfaces. Presentation tools today need to create personalized output based on the user characteristics stored in a user model maintained e.g., by a user modeling server [1] or by the information system itself. Although the user model can be acquired in many different ways, the current focus is on methods that minimize user involvement in the process.

While many existing WIS work with the aforementioned databases as parts of the “deep web”, a web-based presentation interface is also the preferred means of presentation for many new applications, partly because it can also be used for the integration of different functionality into a single user interface. To facilitate the development of WIS, entire presentation frameworks were created that support

the operation of several interconnected presentation tools. However, the design and architectures of integrated WIS with support for adaptation of content, layout and presentation, (semi)automatic user modeling and integration of a set of different presentation tools have not yet been explored sufficiently.

Our aim is to design and evaluate a presentation layer architecture including also personalization facilities to effectively serve the evolving field of WIS development. The rest of the paper is structured as follows. Section 2 presents related work. In section 3, we propose an architecture for the personalized presentation layer of WIS with the respective “user modeling back-end” that allows for the creation of an effective adaptive user interface. We provide an overview of the concepts, goals and methods used in our design. We also describe the requirements, purpose and integration of individual tools that are necessary to realize the proposed functionality. Next, in section 4 we describe our evaluation of the proposed architecture in the domain of online job offers. Finally, in section 5 we summarize our contribution and present ideas for future work.

2 Related Work

The issues of large Web-based Information Systems (WIS) have already been explored in several works from different perspectives with focus on their design, architecture and performance. In [2] authors discuss the problems of large enterprise WIS on the example of a banking system. They identified the need for strong usability, performance and maintainability and defined a “3G-WIS” as “*realize a pleasant use of high quality services, and they will come*” type of service for customers. The authors propose WIS adaptation based on context-awareness and present the corresponding architecture design on an example e-banking WIS.

Existing reference models for adaptive hypermedia can serve well as a basis for WIS architecture design. Apparently the best known is AHAM – Adaptive Hypermedia Architecture Model [3], which is an extension of the generic Dexter reference model for hypermedia. It focuses on the architecture of educational adaptive hypermedia and adds adaptation support based on a domain, user and teaching model with focus on presentation specifications, concepts and concept relationships. Furthermore, unlike the Dexter model, AHAM introduces a permanent and continuously updated user model used for adaptation throughout multiple sessions. The Munich Reference Model for Adaptive Hypermedia [4] follows object-oriented approach to design. Based on the Dexter and AHAM models it specifies an architecture using UML and formal specification expressed by OCL (Object Constraint Language).

With the growing complexity and requirements of adaptive WIS and their practical deployment, issues concerning efficiency and overall performance increase in importance. In [5] the authors propose a flexible presentation architecture for adaptive hypermedia based on the generic Model-View-Controller architecture. They identify problems associated with caching of personalized content, propose caching at different stages of the presentation pipeline and describe examples and performance measurements on the example of an educational WIS.

To facilitate the design and development of WIS for the Semantic Web, the existing model based Hera methodology for WIS was extended with support for adaptability and adaptivity [6]. The authors proposed the use of a profile and user model for adaptation and as well as the use of a conceptual model and domain model that took advantage of ontologies with RDF(S) markup.

In [7] the authors further elaborate on the use of the Hera methodology for Semantic Web WIS design and describe a series of model-driven transformations that are used to generate a hypermedia presentation for the retrieved data. The proposed architecture consists of a semantic layer that specifies the data content, an application layer that describes the abstract view on the data and a presentation layer that specifies the presentation details.

Furthermore, the authors in [8] claim that personalization and adaptation is inevitable in presentation design but has not yet been the central issue of WIS methodologies unlike adaptation in conceptual and navigation design. They describe the AMACONT project that introduced a component-based XML document format, which enables the composition of personalized ubiquitous web presentations from reusable document components encapsulating adaptive content. They combine Hera at the conceptual and navigation levels and AMACONT at the presentation level into an integrated framework for WIS design.

To summarize, many current approaches focus on the design and modeling of the domain model (WIS information content) with extensions related to personalization and performance improvements. We stress the incorporation of user modeling as autonomous client and server side software tools into the system's architecture and its integration with presentation and personalization tools.

3 Personalized Presentation Layer Architecture

Many contemporary information systems employ a standard three-layer architecture consisting of a data layer, an application layer and a presentation layer. In this context, the data layer stores and retrieves data from a database, the application layer performs the core business logic of the system while the presentation layer takes care of the presentation and user interaction [9]. Our focus in this paper is the presentation layer of the typical three-layer architecture with additional focus on web-based information systems for Semantic Web applications. In this respect, we consider the support for the Semantic Web and adaptive hypermedia technologies to be imperative.

Consequently, we think of the presentation layer as a *personalized presentation layer* that performs two primary tasks. First, it provides a user interface that offers simple access to all of the system's functionality while effectively hiding all of the system's inner complexity from the user. Second, it dynamically adapts the user interface to the needs, usage patterns and goals of individual users in order to increase their comfort, productivity and satisfaction by exploiting information stored in a user model. A third – somewhat related task is to create a comprehensive log of user activity, evaluate it and extract and store meaningful

user characteristics, which will then be used in the adaptation process. Thus a suitable personalized presentation layer must fulfill the following requirements:

- Provide an easy to use, user-friendly and intuitive user interface.
- Provide simple access to the system's functionality.
- Offer support for personalization by individual users, which requires
 - support for user model creation and maintenance (logging and acquisition of user activity and characteristics), and
 - support for user model usage to provide adaptation to individual users' needs and usage patterns.

As such, the aforementioned tasks are performed by the presentation part, the personalization part and the user modeling part of the system respectively. Since these tasks depend on each other, our design employs an integrated set of cooperating software components (tools) to realize the necessary functionality (see Figure 1).

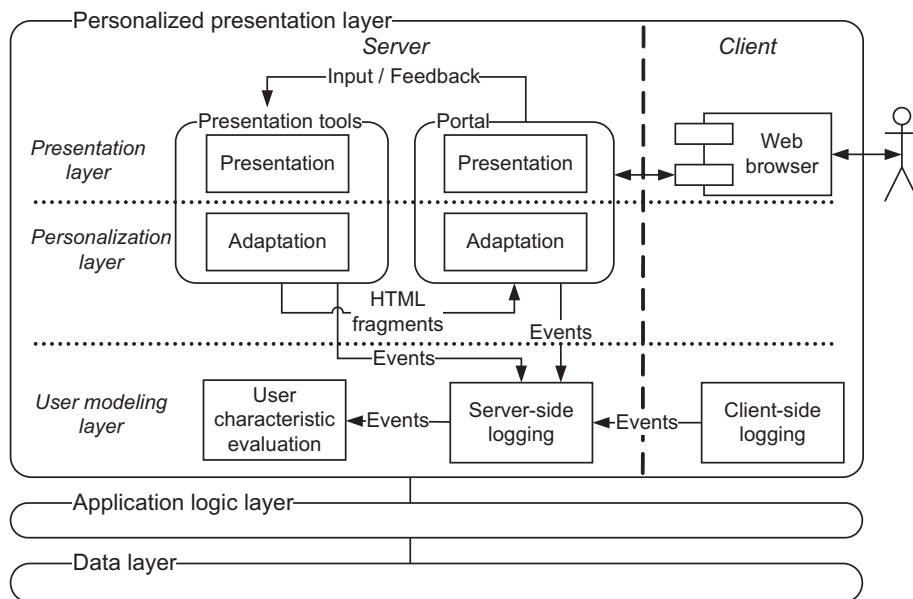


Fig. 1. WIS architecture of the personalized presentation layer

Presentation tools and the portal work with the domain and user models and consist of a presentation and a personalization layer respectively. Individual presentation tools are depicted in the center and forward output to the web portal, which in turn provides an interface to the client web browser (right). The user modeling layer is shown at the bottom and includes both client and server side logging and user characteristic evaluation. All of the already described functionality of a WIS can be examined at the following two levels of abstraction.

Portal Level. The primary purpose of the portal level is to act as means of integration for the system's functionality by providing a common environment for individual tools. The portal provides a common global navigation interface, authentication, authorization and user management services to individual tools (e.g., information about the current user session). It defines the overall high-level functionality and layout of items such as global menus, fields, links and that of individual tools, as well as the overall navigation structure of the whole site.

Tool Level. The purpose of individual tools is to realize or provide access to the functionality of the system. Presentation tools are responsible for the presentation of information and user interaction with each tool being responsible for the handling of its part of the user – system interaction. Thus the tool level defines the internal low-level functionality and layout of items (e.g., controls, menus, text areas, images) for individual presentation tools and the functionality of processing tools which are not directly used for presentation (e.g., for user modeling or business logic).

3.1 Presentation

The task of the presentation part is to provide an effective user interface. While the focus of successful presentation is the layout, style and consistency of the interface, using the most suitable means to present information is also of prime importance. Thus, our portal design consists of pluggable user interface components called portlets. Each portlet represents some logically well defined functionality realized by one or more cooperating tools. The advantage of this approach is that various types of presentation of the same content can be provided to different users. For example, two different presentation tools – one producing graphical output, another producing textual output can be used and the user can choose the most suitable way of presentation.

Consequently, at the tool level, presentation deals with the specifics of individual tools such as layout and style of labels, controls, tables and images all of which are handled by the tools themselves. However, at the portal level the main focus is the overall layout of tools – portlets, each of which contains some specific functionality such as: global navigation (e.g., menus), authentication (e.g., login), notifications (e.g., RSS), user input (e.g., forms), data presentation (e.g., faceted browsers, RDF and OWL visualizers).

3.2 Personalization

Both levels of presentation would be insufficient without proper support for personalization, which can either affect the *visualization* by adapting the layout, style and feel of the user interface or the *functionality* and/or *behavior* of individual tools.

At the portal level, the layout of individual tools can be adapted by changing the order of the displayed portlets. In this way, the usability of the interface can be improved or the accessible functionality can be adapted to the current needs and goals of the user. This can be done by adaptively adding or hiding

items in global menus or by adding or hiding portlets. Furthermore, the portal provides a “skinnable” interface, which allows users to choose their preferred presentation style. Also, additional restrictions can be defined based on user roles and access privileges thus allowing or denying access to specific portlets and their functionality.

The tool level offers possibilities for the adaptation of visualization of individual tools as well as for the adaptation of their functionality. First, a set of guidelines that make it possible to adapt the interface of individual tools to better suit user characteristics can be defined. Next, the functionality of individual tools can be adapted by the tools themselves by exploiting knowledge about user characteristics stored in the user model.

The use of personalization introduces new issues that must be considered in an adaptive information system. The user should have the possibility to explore his own user model in order to know, understand and possibly modify the information, which the system stores about the user and uses for adaptation [1]. Furthermore, support for scrutability is important especially in cases when the system behaves unexpectedly, e.g., recommends illogical items. These needs translate at least into a simple user model browser/editor and a “scrutability explorer” that might be built into each presentation tool. Alternatively it might be a separate tool that would be invoked by other tools.

3.3 User Modeling

Adaptation performed by *adaptation tools* is based on data stored in the user model. This model needs to be created and maintained during the user modeling process. The goal of the process is to identify user characteristics, which might be consecutively used for the adaptation of the system’s operation (e.g., information filtering), for the adaptation of its output (e.g., sorting of search results) or input (e.g., interpretation of user commands) [10]. Thus a successful set of user modeling tools must fulfill the following requirements:

- Provide means for both manual and automatic acquisition of user behavior and user characteristics ideally with little user involvement.
- Provide means of automatic evaluation of the acquired data.
- Allow for the scrutability of the user model by allowing the user to view, inspect and modify its content.

In general, the process of user modeling is perpetual and includes collection of data about the user and processing of the collected data and the update of the user model [11].

User Data Collection. The primary purpose of the user data collection stage is to acquire as much relevant data (e.g., user behavior, documents, preferences) about users as possible while ideally keeping the amount of necessary user involvement as low as possible. This is the reason why among many possible user modeling approaches, we focus on an approach which is based primarily on

the analysis of user behavior within a system. While this approach reduces the amount of user involvement, it requires comprehensive logs of user actions, which must fulfill the following requirements:

1. All relevant user actions (e.g., load of a page, following of hyperlink, use of the back button) and the corresponding (server) events must be logged.
2. For each action the exact time (timestamp) must be logged.
3. Each action must be semantically described (i.e., its meaning must be known).

To fulfill the aforementioned requirements, we chose to expand the presentation and personalization layer onto the client where we use client side monitoring tools to log such events (representing user actions), which the server is normally unaware of (due to browser caching mechanisms, JavaScript interactions etc.). To fulfill the requirement for semantically described actions, we enhance the standard server side logging mechanism with appropriate semantics. Every presentation tool is responsible for supplying the semantics of the performed actions by using a predefined ontology of events.

User Model Update. The aim of the data processing stage is to evaluate the acquired data and estimate meaningful user characteristics with high confidence that might be consequently used for adaptation. The processing depends on the chosen method of data collection and on the nature of collected data. If data represent user actions, individual user characteristics can be estimated by analyzing user navigation on a web site or in the visible information space and by analyzing user feedback on the displayed content. Since user modeling is a perpetual process, each time a user characteristic is identified, the user model is updated to reflect the newly discovered knowledge.

4 Design Evaluation

To evaluate the proposed design we used it for the presentation layer of a system developed in the NAZOU research project¹ aimed at retrieving information in a heterogeneous information space (job offers in the Internet), processing it and presenting it to the user employing adaptive techniques [12].

4.1 Presentation and Personalization

In section 3 we introduced the basic requirements on the functionality of the personalized presentation layer, which consists of several cooperating software tools. We used a single tool for portal level functionality – *JOP – Job Offer Portal* which is the primary user interface and processes all user inputs while returning the corresponding outputs.

The main purpose of *JOP* is to integrate different functionality and provide a form-based user interface tailored for ontology editing with support for

¹ NAZOU Project, <http://nazou.fiit.stuba.sk>

dynamic form generation for a given ontology [13]. *JOP* also allows users to register, log in and customize the global layout of the user interface.

Since *JOP* is based on Apache Cocoon² which supports flexible inclusion of existing functionality into the portal solution, the individual functionality and results of all other tools are easily integrated into the user interface provided by *JOP*. Each tool (or set of tools) is represented by a *coplet* (cocoon portlet) whose position within the overall layout of the portal can be adjusted by the user, who can also minimize, maximize or hide it completely.

On the tool level we developed two cooperating presentation tools, with each tool being responsible for the adaptation of its content according to data from user model. We also developed several simpler tools for the editing of the user profile and for forms filling, both of which are integrated into our portal solution.

Factic – Faceted Semantic Browser [14] is a tool which allows users to effectively navigate the information space by choosing restrictions on the displayed content. It is fully integrated into *JOP*, which supplies it with information about the current user for adaptation purposes. As input it takes user actions and returns the logical description of the content that should be displayed. Its output can be further processed by a set of XSL transformations to directly create valid XHTML output or alternatively it can be sent to *Prescott* for further processing.

Prescott [15] is a presentation tool able to visualize domain dependent content (e.g., job offers) in a flexible and configurable manner using the *Fresnel* presentation ontology. It defines various views on domain content using “lenses”, which can be defined dynamically based on user preferences. As input, *Prescott* takes the logical description of the content (e.g., the URIs of job offer instances) and returns an XHTML fragment with the visualization of the content. Additionally, *Prescott* can take advantage of user characteristics stored in a user model to choose the most appropriate lens to apply on the ontology individuals that should be displayed. *Prescott* is invoked mainly by *Factic* every time the user changes the selected restrictions or decides to view details of a job offer.

4.2 User Modeling

To fulfill the requirements on the data collection stage (see Section 3.3), we developed the client side monitoring tool *Click* [16]. This JavaScript based tool captures and logs browser events and sends them to the server.

Server side logging is enhanced by the *SemanticLog* [16] tool, which combines information from presentation tools and logs acquired by *Click* to create a comprehensive log of user actions with added semantics which are suitable for further processing.

The consecutive data processing is performed by the *LogAnalyzer* [17] tool, which estimates user characteristics and stores them in the user model. Since the used method implies the mainly domain-dependent nature of the revealed characteristics, *LogAnalyzer* needs better “understanding” of the displayed domain content and uses the services provided by the *ConCom* [16] tool, which compares ontological concepts by using various comparison strategies.

² The Apache Cocoon Project, <http://cocoon.apache.org>

4.3 Evaluation

We evaluated the proposed architecture in the domain of job offers by taking advantage of the domain and user ontologies created at our university for project NAZOU. The domain ontology of job offers is used to model domain concepts and their relations as well as to store data about individual job offers. Similarly, the user ontology models user specific concepts and contains data about individual user characteristics. Both of these ontologies are used by the respective tools to process job offer or user related requests (e.g., *Factic* presents job offer instances, while *LogAnalyzer* creates and updates user characteristics).

The domain ontology consists of about 700 classes of which 670 belong to hierarchical classifications with a maximum depth of 6 levels. The ontology contains a test base of about 1 000 job offer instances acquired by different means prior to system evaluation (e.g., manual input, web wrappers)[18]. It is stored in the semantic part of the corporate memory [19] that is responsible for providing means for querying and manipulating the information content enriched by semantics as well as providing the physical back-end for persistent and transient storage of the semantics. The semantic model of the Web content is represented using an ontology (in OWL DL language). Moreover, the user ontology was partially derived from the domain ontology by adding new concepts and relations. The content of the user ontology – individual user characteristics is created and updated by the *LogAnalyzer* tool during system operation.

We developed the above mentioned presentation and user modeling tools and integrated them in a complex WIS thus successfully using the proposed personalized layer architecture design. To verify the integration aspects of the proposed architecture we integrated additional tools that, for example, acquire and present job offer clusters from the application layer or display personalized instance ratings. The integration and interaction of individual tools during request processing is shown in Figure 2.

Client requests are first received by *JOP* which forwards them to appropriate presentation tools or processes them directly in which case it next sends a response to the client. The *Factic* presentation tool prepares the response by querying the domain and user models and invoking *Prescott* to generate XHTML fragments as necessary. It then logs the respective event and resulting display state via the *SemanticLog* web service. Lastly it combines individual XHTML fragments into the final response and sends it to *JOP*, which in turn forwards it to the client.

Furthermore, *JOP* and *Click* independently log events that resulted from user actions via *SemanticLog*, which notifies *LogAnalyzer* about new events for processing. *LogAnalyzer* in turn processes these events asynchronously taking advantage of *ConCom* and updates the user model with newly identified user characteristics.

The user characteristics are used by several software tools from the application layer which contribute to further refinement of user interests. The *Top-k aggregator* tool retrieves the most relevant job offers with respect to user preferences (e.g., salary, education requirements, place) based on ordered lists of user

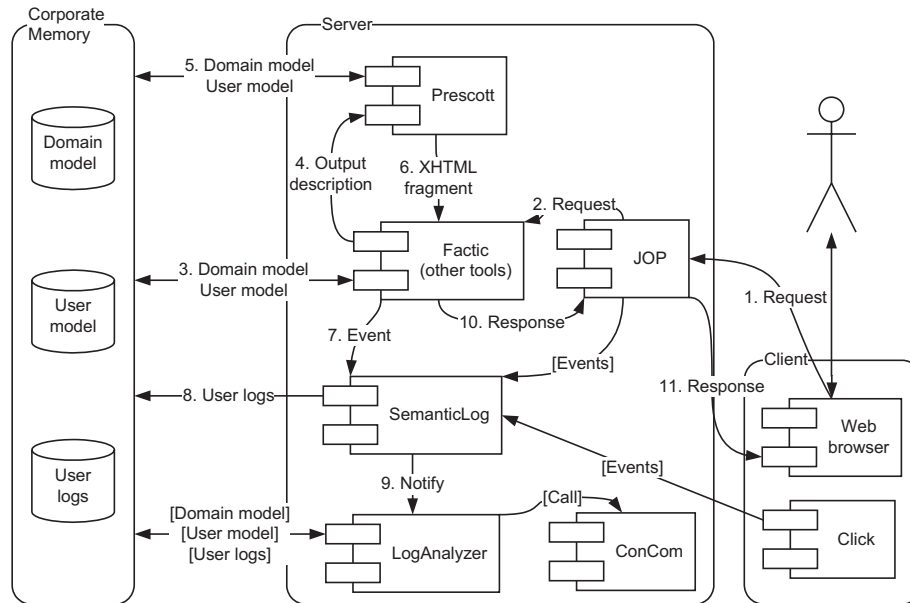


Fig. 2. Request processing in a WIS based on the proposed architecture of the personalized presentation layer

preferences [20]. The *Aspect* tool searches for similar documents (job offers) based on a probabilistic model for soft clustering [21]. Using the described approach for the comparison of domain and user ontology instances the devised clusters are presented to the user based on her characteristics.

5 Conclusions

In this paper we presented requirements that should be met by successful web-based information systems and their respective presentation, personalization and user modeling parts. We proposed an architecture of the personalized presentation layer that consists of several cooperating tools that in combination provide the presentation of data, adaptation of the system's operation and the automatic acquisition and evaluation of user characteristics.

Furthermore, we have realized and evaluated the proposed architecture as well as its individual tools in the domain of online job offers. We continuously cope with the problem of general immaturity of Semantic Web technologies, which are still in a development (e.g., ontologies and ontological repositories).

We also intend to use the proposed architecture in another research project aimed at adaptive presentation and navigation in the space of scientific publications³. This second evaluation will be performed with a different and much

³ MAPEKUS Project, <http://mapekus.fiiit.stuba.sk>

larger domain ontology and with an adapted user ontology, thus putting more focus on domain independence of individual tools and performance issues.

Future work includes deeper analysis and experimentation with the performance and communication overhead associated with the use of independent tools. We also work on the integration of additional tools developed in the course of our research project into the portal *JOP*, the evaluation of additional interconnections between tools and the addition of new tools for specific tasks.

Acknowledgment. This work was partially supported by the Slovak Research and Development Agency under the contract No. APVT-20-007104, the State programme of research and development under the contract No. 1025/04 and the Scientific Grant Agency of Slovak Republic, grant No. VG1/3102/06.

References

1. Bieliková, M. and Kuruc, J.: Sharing User Models for Adaptive Hypermedia Applications. In: 5th Int. Conf. on Intelligent Systems Design and Applications, ISDA'05, Wroclaw, Poland, ACM Press (2005) 506–511
2. Binemann-Zdanowicz, A., Kaschek, R., Schewe, K.D., and Thalheim, B.: Context-Aware Web Information Systems. In: Proc. of the 1st Asian-Pacific Conf. on Conceptual Modelling, APCCM '04, Australian Computer Society, Inc. (2004) 37–48
3. De Bra, P., Houben, G.J., Wu, H.: AHAM: a Dexter-Based Reference Model for Adaptive Hypermedia. In: Proc. of the 10th ACM Conf. on Hypertext and Hypermedia, Hypertext'99, New York, NY, USA, ACM Press (1999) 147–156
4. Koch, N. and Wirsing, M.: The Munich Reference Model for Adaptive Hypermedia Applications. In De Bra, P., Brusilovsky, P., Conejo, R. (eds), Proc. of Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, AH'02, Springer, LNCS **2347** (2002) 213–222
5. Ullrich, C., Libbrecht, P., Winterstein, S., and Mühlenbrock, M.: A Flexible and Efficient Presentation-Architecture for Adaptive Hypermedia: Description and Technical Evaluation. In Kinshuk et al. (ed.), Proc. of 4th IEEE Int. Conf. on Advanced Learning Technologies, ICALT'04, IEEE Computer Society (2004)
6. Frasincar, F. and Houben, G.J.: Hypermedia Presentation Adaptation on the Semantic Web. In De Bra, P., Brusilovsky, P., Conejo, R. (eds), Proc. of 2nd Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, AH'02, Springer, LNCS **2347** (2002) 133–142
7. Houben, G.J., Barna, P., Frasincar, F., and Vdovjak, R.: Hera: Development of Semantic Web Information Systems. In: Proc. of 3rd Int. Conf. on Web Engineering, ICWE'03, Springer, LNCS **2722**
8. Fiala, Z., Frasincar, F., Hinz, M., Houben, G.J., Barna, P., and Meißner, K.: Engineering the Presentation Layer of Adaptable Web Information Systems. In N. Koch et al. (eds), Proc. of 4th Int. Conf. on Web Engineering, ICWE'04, Springer, LNCS **3140** (2004) 459–472
9. Fowler, M.: Patterns of Enterprise Application Architecture. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2002)
10. Kay, J. Human Factors and Ergonomics Series. In: User Modeling for Adaptation. Lawrence Erlbaum Associates (2000)

11. Brusilovsky, P.: Methods and Techniques of Adaptive Hypermedia. *User Model. User-Adapt. Interact.* **6** 2-3 (1996) 87–129
12. Návrát, P., Bieliková, M., Rozinajová, V.: Methods and Tools for Acquiring and Presenting Information and Knowledge in the Web. In: *Int. Conf. on Computer Systems and Technologies, CompSysTech'05*, Varna, Bulgaria (2005)
13. Barla, M., Bartalos, P., Sivák, P., Szobi, K., Tvarožek, M., and Filkorn, R.: Ontology as an Information Base for Domain Oriented Portal Solutions. In: *Proc. of 15th Int. Conf. on Information Systems Development, ISD'06*. (2006)
14. Tvarožek, M.: Personalized Navigation in the Semantic Web. In V. Wade et al. (eds), *Proc. of 4th Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, AH'06*, Springer, LNCS **4018** (2006) 467–471
15. Bieliková, M., Grlický, V., and Kuruc, J.: Framework for Presentation of Information Represented by an Ontology. In Vojtáš, P. (ed.), *Workshop on Theory and Practice of Information Technologies, ITAT'05*, Račková dolina (2005) 325–334
16. Andrejko, A., Barla, M., Bieliková, M., and Tvarožek, M.: Tools for User Characteristics Acquisition. In: *Datakon'06*. (2006) Accepted
17. Barla, M.: Interception of User's Interests on the Web. In V. Wade et al. (eds), *Proc. of 4th Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, AH'06*, Springer, LNCS **4018** (2006) 435–439
18. Bartalos, P., Barla, M., Frivolt, G., Tvarožek, M., Andrejko, A., Bieliková, M., and Návrát, P.: Building an Ontological Base for Experimental Evaluation of Semantic Web Applications. In van Leeuwen, J., Italiano, G.F., van der Hoek, W., Sack, H., Meinel, C., Plášil, F. (eds), *Proc. of SOFSEM '07*, Springer, LNCS (2007) Accepted.
19. Ciglan, M., Babik, M., Laclavik, M., Budinska, I., and Hluchy, L.: Corporate Memory: A Framework for Supporting Tools for Acquisition, Organization and Maintenance of Information and Knowledge. In J. Zendulka (ed.), *Proc. of 9th Int. Conf. on Inf. Systems Implementation and Modelling, ISIM'06*, Perov, Czech Republic (2006) 185–192
20. Gurský, P., Lencses, R., and Vojtáš, P.: Algorithms for User Dependent Integration of Ranked Distributed Information. In et al., M.B. (ed.), *TCGOV 2005 Poster Proceedings*, Bozen-Bolzano, Italy (2005)
21. Polčicová, G. and Tiño, P.: Making Sense of Sparse Rating Data in Collaborative Filtering via Topographic Organisation of User Preference Patterns. *Neural Networks* **17** (2004) 1183–1199

Príloha D Technická dokumentácia

Technickú dokumentáciu sme rozdelili na tri časti – opis použitia fazetového prehliadača (časť D.1), opis nasadenia a konfigurácie nástrojov *Factic* a *SemanticLog* (časť D.2) a opis ich návrhu a implementácie (časť D.3).

D.1 Použitie fazetového sémantického prehliadača

Použitie fazetového sémantického prehliadača pozostáva z dvoch krokov – z definície ohraničení nad informačným priestorom a z prehliadania výsledkov vyhľadávania. Používateľ definuje ohraničenia v informačnom priestore pomocou faziet v ľavej časti používateľského rozhrania (Obrázok D-1). Výber ohraničenia sa uskutoční kliknutím na príslušnú reštrikciu vo fazete (nasledovaním odkazu). Počet inštancií informačného priestoru vyhovujúci ohraničeniu je zobrazený v zátvorke za každým ohraničením. Aktiváciu, resp. zobrazenie skrytej fazety používateľ vyvolá pomocou odkazu pri názve fazety (červené/zelené tlačidlo). Odporúčané ohraničenie je vo fazete zobrazené na začiatku zoznamu ohraničení so zeleným pozadím.

>> has duty location	has duty location: All > World > North America > UNITED STATES > California (23)
Anaheim (1)	offers position: All > Professionals (23)
Berkeley (2)	has benefit: All (23)
Cypress (1)	is active in: All (23)
Glendale (2)	involves travelling: All (23)
Glendale (2)	has contract type: All (23)
Irvine (1)	is of management level: All (23)
Pasadena (2)	has job term: All (23)
Pasadena (1)	date and time of the offer acquisition: All (23)
San Diego (1)	hours per week: All (23)
San Francisco (4)	start date: All (23)
San Jose (1)	
Santa Monica (2)	
US CA Irvine (1)	
US CA San Francisco (4)	
US CA San Jose (1)	
>> offers position	
Computing professionals (17)	
Computing professionals Programmer (4)	
Junior Programmer (2)	
Mainframe Database Programmer (1)	
Physical mathematical and engineering science professionals (17)	
Programmer (4)	
Programmer (1)	
Programmer/Analyst (2)	
Senior Programmer Analyst - Tibco (1)	
System Programmer (2)	
has benefit	

Total Matches: 23 on 2 pages.
Page: 1 | 2 | [Next >](#)

Sort by: offer name | salary | Organization | región Item per page: 10 | 15 | 25 | 50 | 100

#	offer name	salary	Organization	región	Actions
1	Systems Programmer	50.0	Genesis10	California	
2	COBOL BASIC Programmers VMS DEC Alpha	35.0	Indotronic International Corp.	Glendale	
3	At-Tech, Orange, CA US	25000.0	At-Tech	California	
4	6181 - Systems Programmer	85000.0	Genesis10	Pasadena	
5	C# / ASP.NET Programmers	35000.0	At-Tech	Anaheim	
6	Manpower Professional, San Diego, CA US	29.0	Manpower Professional	California	
7	Programmer	55.0	Huntress IT - Contract UK	California	
8	Senior Programmer Analyst	50.0	Genesis10	Berkeley	
9	Programmer Analyst II	40.0	Genesis10	Berkeley	
10	Oracle 7-9i Programmer (PL/SQL, DataStage, AIX)		Strategi	Cypress	
11	Senior Business Analyst - Financial Sector		RDF Resources Ltd	Glendale	
12	Junior Programmer	35000.0		Glendale	
13	Junior Programmer			Glendale	
14	Java J2EE Web Developer - Financial Services	35.0		Irvine	
15	6081 - Sr. Mainframe Database Programmer	75000.0	Kaiser Permanente	Pasadena	

Obrázok D-1: Zobrazenie prehľadu výsledkov vyhľadávania (inštancií z doménovej ontológie) vo fazetovom sémantickom prehliadači.

Aktuálne definované ohraňovania informačného priestoru sú zobrazené v (pravej) hornej časti rozhrania. Výsledky ohraňovania priestoru sú zobrazené v pravej dolnej časti rozhrania, kde sa v tabuľke zobrazujú vybrané atribúty nájdených inštancií. Nad tabuľkou sú umiestnené odkazy pre navigáciu vo výsledkoch po stranách, voľbu počtu výsledkov zobrazených na jednej strane a voľbu usporiadania výsledkov.

Podrobné informácie o konkrétnej inštancii je možné zobrazit' pomocou kliknutia na jej názov (Obrázok D-2). Návrat na prehľad inštancií je možný pomocou odkazu späť – „Back“.

[Back](#)

User:	unknown		
Apply information:	Text:	Manpower Professional, San Diego, CA US	
Active:	true		
Prerequisites:	Text:	XML Oracle ERP 5 to 8 years of software development experience PL/SQL ODBC 2 plus years of experience in TIBCO Business Works and WorkFlow development	
		Position available to new or recent college graduates? NO Bachelor of Science degree in a technical field (i.e. Computer Science, Information Technology, Engineering or related) required	
Offers salary:	Text:	Salary: \$29/HOURLY To \$36/HOURLY	
	Base:	29.0	
	Maximum:	36.0	
Duty location:	rdfs:label:	California	
Offered by:	Name:	Manpower Professional	
Date:	20.10.2006		
Name:	Manpower Professional, San Diego, CA US		
jo:hasSource:			

Obrázok D-2: Zobrazenie podrobných informácií o inštancii z doménovej ontológie.

D.2 Nasadenie a konfigurácia nástrojov

D.2.1 Predpoklady pre použitie nástrojov

Použitie nástrojov *Factic* a *SemanticLog* predpokladá správne nainštalované prostredie:

- Java runtime verzie 5 a vyššia
- Java servlet kontajner Apache Tomcat 5.5.12 a vyšší
 - Prezentačný server Apache Cocoon 2.1.8 (vyššie verzie neboli odskúšané)
 - V Cocoon musí byť nasadený JOP (Portál pracovných príležitostí z projektu NAZOU) alebo P3 (Portál vedeckých publikácií z projektu MAPEKUS)

- Kontajner pre webové služby Apache Axis
- Ontologické úložisko Sesame 1.2.3 a vyššie
 - V Sesame musí byť vytvorené RDFS úložisko s príslušnou doménovou ontológiou a ontológiou používateľa z projektov NAZOU/MAPEKUS.
- Relačná databáza MySQL 5.0 a vyššia
 - V MySQL musí byť vytvorená databáza userlogs s vytvorenou príslušnou databázovou štruktúrou z projektov NAZOU/MAPEKUS pomocou databázových skriptov pre *SemanticLog* a *LogAnalyzer*.

D.2.2 Nasadenie nástroja Factic

Pre nasadenie nástroja *Factic* je potrebné vykonať nasledovnú postupnosť krokov:

- 1) Vložiť doménovú ontológiu v0.6.17 (pracovné ponuky), resp. v0.0.0 (vedecké publikácie) do ontologického úložiska.
- 2) Nasadiť *Factic* ako copleť do portálu JOP/P3.
- 3) Skopírovať knižnice nástroja *Factic* do Cocoonu.
 - a) Jednotlivé knižnice sú v *Factic/lib* a *Factic/out*.
- 4) Nakonfigurovať nástroj *Factic* v Cocoeone
 - a) Nastaviť *sitemap.xmap* – dátovody a mapovanie dopytov v Cocoeone.
 - b) Nakonfigurovať *factic.properties* – prístup k dátovým úložiskám.
 - c) Nakonfigurovať *factic.xml* – konfigurácia faziet a dátových typov.
 - d) Nastaviť *factic.xslt* – konfigurácia zobrazenia používateľského rozhrania.
- 5) Nasadiť/nakonfigurovať nástroj *SemanticLog*, ktorý *Factic* volá na zaznamenanie udalostí o akciách používateľa.

Ukážka konfigurácie *sitemap.xmap*, ktorá vytvára výstup nástroja *Factic*. Spracovanie požiadavky prebieha v nasledovných krokoch:

- 1) Pridanie informácií o používateľovi do parametrov.
- 2) Spustenie generátora, ktorý vytvorí opis rozhrania s dátami o fazetách a inštanciách.
- 3) Transformácia opisu rozhrania do XHTML pomocou XSL šablóny.
- 4) Spracovanie internacionalizácie cez *i18n* transformátor.
- 5) Prepisovanie URL vo výstupe (angl. URL rewriting) pomocou XSL šablóny.
- 6) Serializáciu do XHTML.

```
<map:match pattern="factic">
  <map:call function="AddUserBeanToParams"/>
</map:match>

<map:match pattern="factic-internal">
```

```

<!-- Generate output description. -->
<map:generate type="Factic">
  <map:parameter name="toolPath" value="\portal\coplets\Factic\"/>
</map:generate>

<!-- Format HTML output. -->
<map:transform src="factic.xslt"/>

<!-- Internationalization. -->
<map:transform type="i18n">
  <map:parameter name="locale" value="en_US"/>
</map:transform>

<!-- Create HTML output and process URL rewriting. -->
<map:transform src="factic2html.xslt">
  <map:parameter name="contextPath" value="{request.contextPath}/portal/coplets/Factic"/>
</map:transform>

<map:serialize type="xhtml"/>
</map:match>

```

Ukážka konfigurácie *factic.properties* – dátových úložísk pre nástroj *Factic*, ktorá obsahuje nastavenie ontologického úložiska Sesame a HSQL úložiska pre prístup k dátam o používateľoch z portálu JOP/P3.

```

#Factic Properties
SesameRepositoryURL=http://localhost:8080/sesame/
SesameRepositoryID=nazou-rdfs-mem
SesameRepositoryUsername=user
SesameRepositoryPassword=passoword
HSQLRepositoryURL=../DB/OfferDB
HSQLRepositoryUsername=sa
HSQLRepositoryPassword=

```

Ukážka konfigurácie faziet a dátových typov v súbore *factic.xml*. Samotná konfigurácia pozostáva zo štyroch častí:

- 1) Globálnej konfigurácie nástroja Factic (jazyk, typ inštancií, počet súčasne zobrazených inštancií a hĺbka rekurzívneho zisťovania vlastností inštancií).
- 2) Konfigurácie akcií na usporiadanie inštancií (názov, typ, vlastnosť podľa ktorej sa usporiadajú inštancie)
- 3) Konfigurácie dostupných faziet (typ, vlastnosti priame/nepriame)
- 4) Konfigurácie dátových typov pre prezentáciu s ktorými sa bude pracovať (literály, triedy, vlastnosti, typy, uri)

```

<?xml version="1.0"?>
<!-- Factic (Faceted Semantic Browser) configuration file
      Author: Michal Tvarožek -->

<!-- language      - the default language for label queries -->
<!-- instanceTypeURI - the type of instances that are retrieved and displayed -->
<!-- JobOffers:      jo:JobOffer -->
<!-- ResearchArticles: unnamed:Author -->
<factic language="en" instanceTypeURI="jo:JobOffer" itemsPerPage="15"
      maxInstanceValueDepthOverview="2"      maxInstanceValueDepthDetails="3"
      maxInstanceCount="1000000">

  <!-- Sorting actions that can be executed over search results. -->
  <sortingActions>
    <sortingAction label="ofr:name" type="xsd:string" propertyURI="ofr:name"/>
    <sortingAction label="jo:Salary" type="xsd:float" propertyURI="jo:minAmount"
      indirectPropertyURI0="jo:hasSalary"/>
    <sortingAction label="jo:Organization" type="xsd:string" propertyURI="jo:name"
      indirectPropertyURI0="jo:isOfferedBy"/>
    <sortingAction label="r:Region" type="xsd:string" propertyURI="rdfs:label"
      indirectPropertyURI0="jo:hasDutyLocation"/>
  </sortingActions>

  <!-- Facets that can be used for defining restrictions. -->
  <facets>
    <!-- Initialization for the domain of online job offers. -->
    <facet label="Region" imageFile="Region.png" type="InstanceHierarchyFacet"
      propertyURI="jo:hasDutyLocation" rangeURI="r:Region" transitivePropertyURI="r:isPartOf"/>
    <facet label="Profession" imageFile="Salary.png" type="ClassHierarchyFacet"
      propertyURI="jo:offersPosition" rangeURI="c:ProfessionClassification"/>
    <facet label="Management" imageFile="Region.png" type="ClassHierarchyFacet"
      propertyURI="jo:isOfManagementLevel" rangeURI="jo:ManagementLevel"/>

    <facet label="Date" imageFile="Region.png" type="AbsoluteDateLiteralFacet" propertyURI="jo:startDate"
      rangeURI="xsd:dateTime" minYear="1990" maxYear="2006"/>

    <facet label="Acquisition Date" imageFile="Region.png" type="RelativeDateLiteralFacet"
      propertyURI="ofr:acquisitionDate" indirectPropertyURI0="ofr:hasSource"
      rangeURI="xsd:dateTime"/>

    <facet label="Hours per Week" imageFile="Region.png" type="FloatLiteralFacet"
      propertyURI="jo:hoursPerWeek" rangeURI="xsd:float" minValue="10" maxValue="40"
      partitionCount="5"/>
    <facet label="Traveling" imageFile="Region.png" type="ClassHierarchyFacet"
      propertyURI="jo:involvesTraveling" rangeURI="jo:TravelingLevel"/>
    <facet label="Benefit" imageFile="Region.png" type="ClassHierarchyFacet" propertyURI="jo:hasBenefit"
      rangeURI="jo:Benefit"/>
    <facet label="Contract" imageFile="Region.png" type="ClassHierarchyFacet"
      propertyURI="jo:hasContractType" rangeURI="jo:ContractType"/>

```

```

<facet label="Term" imageFile="Region.png" type="ClassHierarchyFacet" propertyURI="jo:hasJobTerm"
    rangeURI="jo:JobTerm"/>
<facet label="Industry" imageFile="Industry.png" type="ClassHierarchyFacet" propertyURI="jo:isActiveIn"
    indirectPropertyURI0="jo:isOfferedBy" rangeURI="c:IndustryClassification"/>
</facets>

<!-- Data types that can be accessed for processing, retrieval and presentation. -->
<instanceTypes>
    <!-- Literal types. -->
    <instanceType uri="xsd:string" literal="true"/>
    <instanceType uri="xsd:float" literal="true"/>
    <instanceType uri="xsd:dateTime" literal="true"/>

    <!-- +
    | Job Offer domain instance types.
    + -->
    <!-- Salary class. -->
    <instanceType uri="jo:Salary" literal="false">
        <property uri="jo:minAmount" range="xsd:float" provider="SesameProvider"/>
        <property uri="jo:maxAmount" range="xsd:float" provider="SesameProvider"/>
        <property uri="jo:text" range="xsd:string" provider="SesameProvider"/>
    </instanceType>

    <!-- Organization class. -->
    <instanceType uri="jo:Organization" literal="false">
        <property uri="jo:name" range="xsd:string" provider="SesameProvider"/>
    </instanceType>

    <!-- Region class. -->
    <instanceType uri="r:Region" literal="false">
        <property uri="rdfs:label" range="xsd:string" provider="SesameProvider"/>
    </instanceType>

    <!-- ApplyInformation class. -->
    <instanceType uri="jo:ApplyInformation" literal="false">
        <property uri="jo:text" range="xsd:string" provider="SesameProvider"/>
    </instanceType>

    <!-- Prerequisite class. -->
    <instanceType uri="jo:Prerequisite" literal="false">
        <property uri="jo:text" range="xsd:string" provider="SesameProvider"/>
    </instanceType>

    <!-- OfferSource class. -->
    <instanceType uri="ofr:OfferSource" literal="false">
        <property uri="ofr:acquisitionDate" range="xsd:dateTime" provider="SesameProvider"/>
    </instanceType>

    <!-- JobOffer class. -->
    <instanceType uri="jo:JobOffer" literal="false">

```

```

<!-- Sesame properties. -->
<property uri="ofr:name" range="xsd:string" provider="SesameProvider"/>
<property uri="jo:hasSalary" range="jo:Salary" provider="SesameProvider"/>
<property uri="jo:isOfferedBy" range="jo:Organization" provider="SesameProvider"/>
<property uri="jo:hasDutyLocation" range="r:Region" provider="SesameProvider"/>
<property uri="jo:hasApplyInformation" range="jo:ApplyInformation" provider="SesameProvider"/>
<property uri="jo:hasPrerequisite" range="jo:Prerequisite" provider="SesameProvider"/>
<property uri="jo:hasSource" range="ofr:OfferSource" provider="SesameProvider"/>

<!-- HSQL properties. -->
<property uri="jop:user" range="xsd:string" provider="HSQLProvider"/>
<property uri="jop:date" range="xsd:string" provider="HSQLProvider"/>
<property uri="jop:active" range="xsd:string" provider="HSQLProvider"/>
</instanceType>
</instanceTypes>
</factic>

```

Ukážka prepisovania URL adries pre elementy *img* (obrázky) – skopírovanie všetkých atribútov, doplnenie atribútu *src* o prefix k umiestneniu obrázkov, doplnenie internacionalizácie opisných textov (z XSL transformácie).

```

<!-- Transform URL for images. -->
<xsl:template match="img">
  <img>
    <xsl:copy-of select="@*"/>
    <xsl:attribute name="src">
      <!-- Use the context path attribute to access local images in tool path. -->
      <!--<xsl:value-of select="$contextPath"/>-->
      <!-- Use the images directory for images in the skin path. -->
      <xsl:text>images/</xsl:text>
      <xsl:value-of select="@src"/>
    </xsl:attribute>
    <xsl:attribute name="i18n:attr">
      <xsl:text>alt</xsl:text>
    </xsl:attribute>
    <xsl:apply-templates/>
  </img>
</xsl:template>

```

D.2.3 Nasadenie nástroja SemanticLog

Pre nasadenie nástroja *SemanticLog* je potrebné vykonať nasledovnú postupnosť krokov:

- 1) Inicializovať MySQL databázu pomocou skriptu *createTables.txt*.
- 2) Nasadiť webovú službu *SemanticLog* pomocou deployment deskriptoru *SemanticLog.wsdd* do kontajnera Axis.

- 3) Skopírovať knižnice nástroja *SemanticLog* do Axisu a Cocoonu.
 - a) Jednotlivé knižnice sú v *SemanticLog/lib* a *SemanticLog/out*.
- 4) Nakonfigurovať *SemanticLog.properties* v Axisu i Cocooone.
- 5) Nasadiť nástroj *LogAnalyzer*, ktorý *SemanticLog* volá po zaznamenaní udalosti na jej spracovanie. Inicializovať MySQL cez skript *createTables_LogAnalyzer.txt*.

D.3 Opis návrhu nástrojov

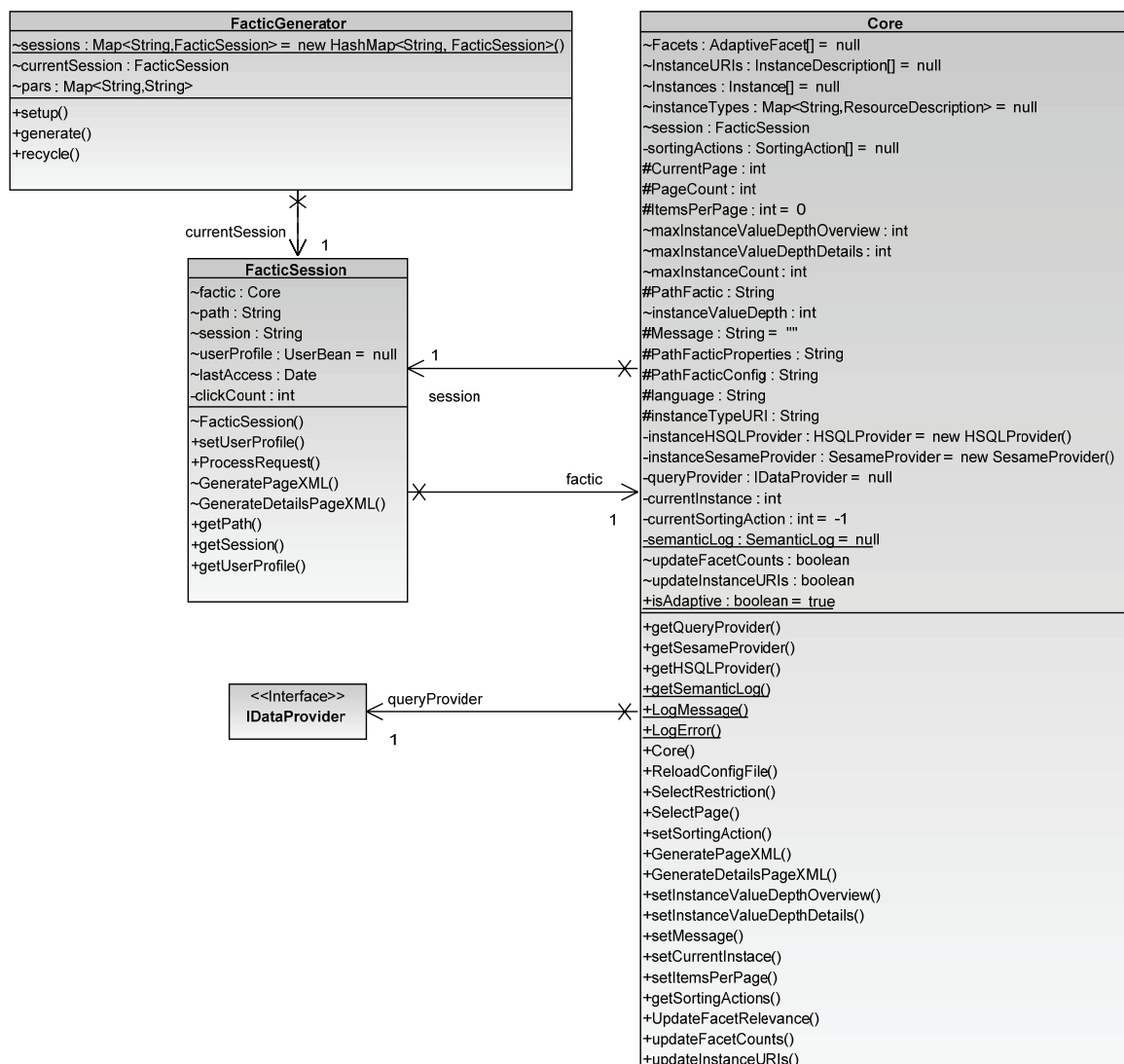
V tlačenej prílohe uvádzame prehľad požiadaviek, významných častí technického návrhu a implementácie nástrojov *Factic* a *SemanticLog*. Podrobný opis jednotlivých balíkov, tried a metód pre oba nástroje (javadoc) sa nachádza na priloženom elektronickom médiu.

D.3.1 Opis návrhu nástroja Factic

Nástroj *Factic* je navrhnutý ako generátor pre prezentačný server Apache Cocoon. *Factic* má modulárny návrh, jeho implementáciu sme rozdelili do týchto balíkov:

- *sk.fiit.nazou.factic* – hlavný balík, ktorý obsahuje jadro nástroja *Core*, riadenie sedení používateľa *FacticSession* a generátor výstupu pre Cocoon *FacticGenerator*.
- *sk.fiit.nazou.factic.DataProvider* – dátová vrstva nástroja, ktorá obsahuje rozhrania pre dátové zdroje *IDataProvider* a fazety *IFacetProvider*.
- *sk.fiit.nazou.factic.DataProvider.HSQL* – implementácia dátového zdroja pre HSQL.
- *sk.fiit.nazou.factic.DataProvider.Sesame* – implementácia dátových zdrojov pre Sesame.
- *sk.fiit.nazou.factic.Facet* – obsahuje triedy realizujúce funkcionality jednotlivých typov faziet, ich adaptáciu ako aj ohraničenia vo fazetách.
- *sk.fiit.nazou.factic.Resource* – obsahuje triedy pre načítanie zdrojov z ontológie a generovanie výstupu z nich.

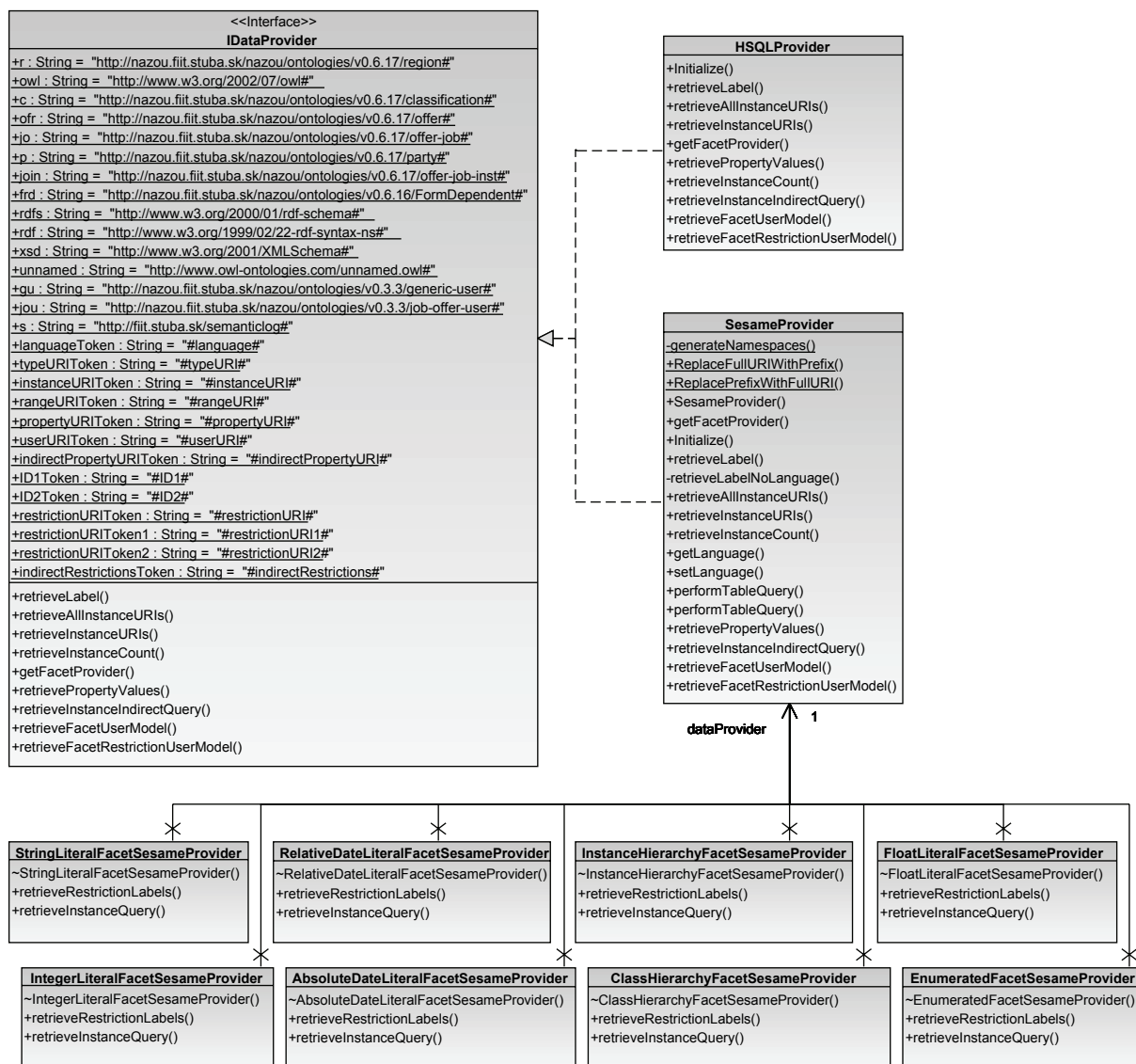
Obrázok D-3 obsahuje diagram tried balíka *sk.fiit.nazou.factic*. Generátor výstupu pre Cocoon implementuje trieda *FacticGenerator*, ktorá spracúva požiadavky prichádzajúce od používateľa. Riadenie sedení používateľa zabezpečuje trieda *FacticSession*, pričom pre každé sedenie používateľa generátor vytvorí nový objekt tejto triedy.



Obrázok D-3: Diagram tried balíka *sk.fiit.nazou.factic*.

Jadro spracovania fazetového prehliadača realizuje trieda *Core*, ktorá obsahuje stav prehliadania informačného priestoru používateľom a zabezpečuje formuláciu a spracovanie dopytov do Sesame ako aj prezentáciu výsledkov a adaptáciu používateľského rozhrania.

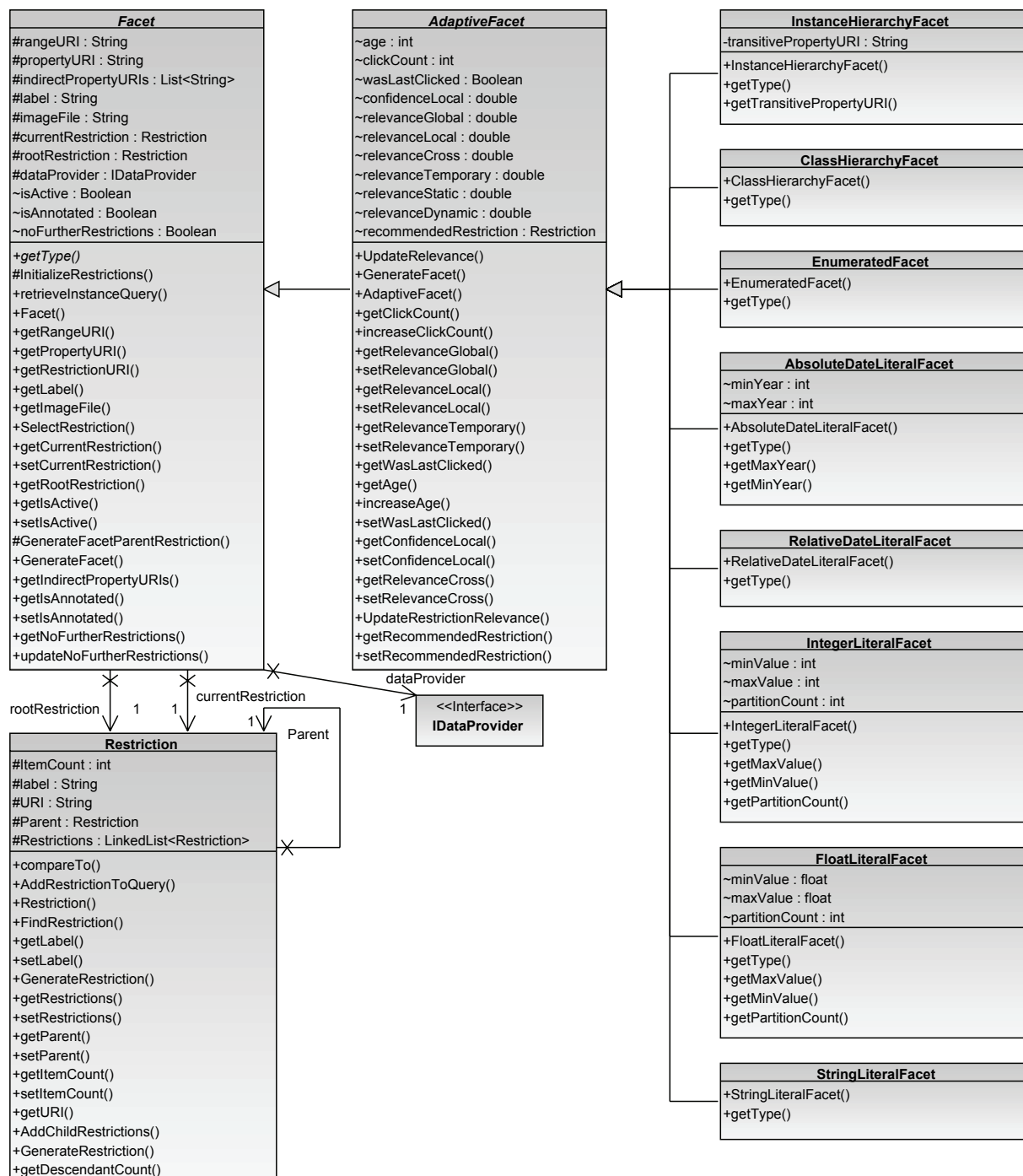
Dátová vrstva nástroja je realizovaná balíkom *sk.fiit.nazou.factic.DataProvider* (Obrázok D-4), ktorý obsahuje definíciu rozhraní určených na prístup k dátovým úložiskám. Rozhranie *IDataProvider* slúži na „vysoko úrovňový“ prístup k dátam, rozhranie *IFacetProvider* slúži na tvorbu dopytov do dátového úložiska odvodených od rôznych typov faziet.



Obrázok D-4: Diagram tried balíka *sk.fiit.nazou.factic.DataProvider*.

Služby dátovej vrstvy využíva trieda *Core* na zabezpečenie tvorby a vykonania dopytov pomocou faziet viacerých typov. Triedy zabezpečujúce funkcionality faziet sa nachádzajú v balíku *sk.fiit.nazou.factic.Facet* (Obrázok D-5). Základnú funkcionality fazety, ako sú generovanie výstupných dát, tvorba dopytov a správa ohraňení vo fazete, realizuje abstraktná trieda *Facet*. Táto definuje aj virtuálne metódy, ktoré sú preťažené v odvodených triedach a implementujú tak vlastnosti rôznych typov faziet.

Jednotlivé ohraňenia vo fazetách sú implementované pomocou triedy *Restriction*, ktorá zabezpečuje generovanie výstupných dát pre ohraňenia a udržiava informácie o hierarchii ohraňení vo fazete, počte inštancií vyhovujúcich ohraňeniam a mapovaní ohraňení na vlastnosti inštancií v ontológii.



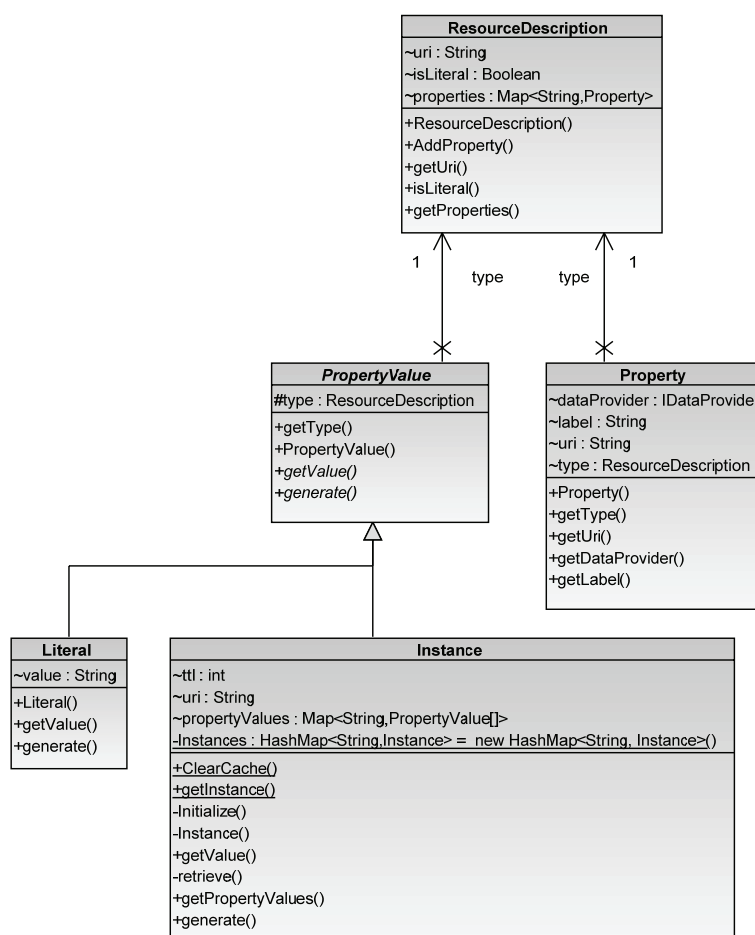
Obrázok D-5: Diagram tried balíka *sk.fiit.nazou.factic.Facet*.

Prispôsobovanie faziet je realizované triedou *AdaptiveFacet*, ktorá obsahuje atribúty a metódy potrebné k práci s relevanciou a spoľahlivosťou faziet získaných z modelu používateľa. Od triedy *AdaptiveFacet* sú odvodené jednotlivé typy faziet:

- *InstanceHierarchyFacet* – realizuje fazetu založenú na hierarchii inštancií z ontológie.
- *ClassHierarchyFacet* – realizuje fazetu založenú na hierarchii tried z ontológie.
- *EnumeratedFacet* – realizuje fazetu založenú na enumerácii inštancií z ontológie.

- *AbsoluteDateLiteralFacet* – realizuje fazetu založenú na literále typu dátum.
- *RelativeDateLiteralFacet* – realizuje fazetu založenú na literále typu dátum.
- *FloatLiteralFacet* – realizuje fazetu založenú na literále typu reálne číslo.
- *IntegerLiteralFacet* – realizuje fazetu založenú na literále typu celé číslo.
- *StringLiteralFacet* – realizuje fazetu založenú na literále typu textový reťazec.

Výsledky ohraničenia informačného priestoru tvoria jeho inštanície. Informácie o zobrazovaných inštanciách nástroj načítava z ontologického úložiska pomocou dátovej vrstvy cez rozhranie *IDataProvider*. Načítané informácie o inštanciách a ich objektových a dátových vlastnostiach sú uložené do objektového modelu, ktorý je implementovaný v balíku *sk.fiit.nazou.factic.Resource* (Obrázok D-6). Opis zdrojov – mapovanie konceptov a ich vlastností na ontológiu realizuje trieda *ResourceDescription*. Prácu s hodnotami konceptov implementuje trieda *PropertyValue*, od ktorej je odvodená trieda *Literal* pre dátové vlastnosti a trieda *Instance* pre objektové vlastnosti konceptov. Trieda *Property* implementuje prácu s prepojeniami konceptov pomocou vlastností v ontológii.



Obrázok D-6: Diagram tried balíka *sk.fiit.nazou.factic.Resource*.

Ukážka zdrojového kódu – spracovanie požiadavky na zobrazenie prehľadu inštancií. Spracovanie parametrov volania z URL – obnova interného stavu na základe vykonanej akcie používateľa, aktualizácia stavu adaptácie fazetového prehliadača, príprava záznamu o akcii

používateľa. Aktualizácia relevancie fazieta a ohraničení z modelu používateľa, inicializácia nového stavu a vygenerovanie výstupného XML dokumentu. Finálny záznam akcie používateľa do MySQL databázy pomocou volania nástroja SemanticLog.

```
void GeneratePageXML(ContentHandler contentHandler, Map<String, String> pars) throws SAXException
{
    String event = "http://fiit.stuba.sk/semanticlog#Event_ShowOverview";

    LinkedList<String> list1 = new LinkedList<String>();
    LinkedList<String> list2 = new LinkedList<String>();
    HashMap<String, Object[]> map = new HashMap<String, Object[]>();

    try
    {
        factic.ReloadConfigFile();
        factic.setInstanceValueDepthOverview();

        // Update facet adaptation state.
        if (pars.containsKey("facet"))
        {
            clickCount++;
            for (int i = 0; i < factic.Facets.length; i++)
            {
                factic.Facets[i].increaseAge();
            }
        }

        // Apply state changes.
        if (pars.containsKey("itemsperpage"))
        {
            int itemsPerPage = Integer.valueOf(pars.get("itemsperpage"));

            Core.LogMessage("Select Items per Page: " + pars.get("itemsperpage"));
            factic.setItemsPerPage(itemsPerPage);

            // Prepare event log changes.
            event = "http://fiit.stuba.sk/semanticlog#Event_SelectItemsPerPage";
            list1.add(SesameProvider.ReplacePrefixWithFullURI(pars.get("itemsperpage")));
            map.put("http://fiit.stuba.sk/semanticlog#Attribute_Count", list1.toArray());
        }
        if (pars.containsKey("facet") && pars.containsKey("enable"))
        {
            int facet = Integer.valueOf(pars.get("facet"));
            boolean enable = Boolean.valueOf(pars.get("enable"));

            // Update adaptation values.
            for (int i = 0; i < factic.Facets.length; i++)
                factic.Facets[i].setWasLastClicked(false);
            factic.Facets[facet].increaseClickCount();
        }
    }
}
```

```

Core.LogMessage("Activate Facet: " + pars.get("facet") + " Activation: " +
    pars.get("enable"));
factic.Facets[facet].setIsActive(enable);

// Prepare event log changes.
if (enable)
    event = "http://fiit.stuba.sk/semanticlog#Event_FacetEnable";
else
    event = "http://fiit.stuba.sk/semanticlog#Event_FacetDisable";
list1.add(SesameProvider.ReplacePrefixWithFullURI(
    factic.Facets[facet].getPropertyURI()));
map.put("http://fiit.stuba.sk/semanticlog#Attribute_Property", list1.toArray());
}
if (pars.containsKey("facet") && pars.containsKey("annotate"))
{
    int facet = Integer.valueOf(pars.get("facet"));
    boolean annotate = Boolean.valueOf(pars.get("annotate"));

    // Update adaptation values.
    for (int i = 0; i < factic.Facets.length; i++)
        factic.Facets[i].setWasLastClicked(false);
    factic.Facets[facet].increaseClickCount();

    Core.LogMessage("Annotate Facet: " + pars.get("facet") + " Annotate: " +
        pars.get("annotate"));
    factic.Facets[facet].setIsAnnotated(annotate);
}
if (pars.containsKey("facet") && pars.containsKey("restriction"))
{
    int facet = new Integer(pars.get("facet")).intValue();

    // Update adaptation values.
    for (int i = 0; i < factic.Facets.length; i++)
        factic.Facets[i].setWasLastClicked(false);
    factic.Facets[facet].increaseClickCount();

    Core.LogMessage("SelectFacetRestriction: " + pars.get("facet") + " Restriction: " +
        pars.get("restriction"));
    factic.SelectRestriction(facet, pars.get("restriction"));
    factic.updateInstanceURIs();
    factic.updateFacetCounts();

    // Prepare event log changes.
    event = "http://fiit.stuba.sk/semanticlog#Event_SelectRestriction";

list1.add(SesameProvider.ReplacePrefixWithFullURI(factic.Facets[facet].getPropertyURI()));
map.put("http://fiit.stuba.sk/semanticlog#Attribute_Property", list1.toArray());

```

```

list2.add(SesameProvider.ReplacePrefixWithFullURI(factic.Facets[facet].getRootRestriction().
    FindRestriction(pars.get("restriction")).getURI());
    map.put("http://fiit.stuba.sk/semanticlog#Attribute_Restriction", list2.toArray());
}
if (pars.containsKey("selectpage"))
{
    factic.SelectPage(new Integer(pars.get("selectpage")).intValue());

    // Prepare event log changes.
    event = "http://fiit.stuba.sk/semanticlog#Event_PageSelect";
    list1.add(pars.get("selectpage"));
    map.put("http://fiit.stuba.sk/semanticlog#Attribute_Page", list1.toArray());
}
if (pars.containsKey("prevpage"))
{
    factic.SelectPage(new Integer(pars.get("prevpage")).intValue()+1);

    // Prepare event log changes.
    event = "http://fiit.stuba.sk/semanticlog#Event_PagePrevious";
    list1.add(pars.get("prevpage"));
    map.put("http://fiit.stuba.sk/semanticlog#Attribute_Page", list1.toArray());
}
if (pars.containsKey("nextpage"))
{
    factic.SelectPage(new Integer(pars.get("nextpage")).intValue());

    // Prepare event log changes.
    event = "http://fiit.stuba.sk/semanticlog#Event_PageNext";
    list1.add(pars.get("nextpage"));
    map.put("http://fiit.stuba.sk/semanticlog#Attribute_Page", list1.toArray());
}
if (pars.containsKey("nextsortparam") && pars.containsKey("nextsortascending"))
{
    factic.setSortingAction(Integer.valueOf(pars.get("nextsortparam")),
        Boolean.valueOf(pars.get("nextsortascending")));

    // Prepare event log changes.
    event = "http://fiit.stuba.sk/semanticlog#Event_SelectSortingOrder";

    List<String> properties =

factic.getSortingActions()[Integer.valueOf(pars.get("nextsortparam"))].getPropertyURIs();
list1.add(properties.get(properties.size()-1));
if (Boolean.valueOf(pars.get("nextsortascending")))
    list2.add("http://fiit.stuba.sk/semanticlog#SortingOrder_Ascending");
else
    list2.add("http://fiit.stuba.sk/semanticlog#SortingOrder_Descending");
map.put("http://fiit.stuba.sk/semanticlog#Attribute_Property", list1.toArray());
map.put("http://fiit.stuba.sk/semanticlog#Attribute_SortingOrder", list2.toArray());

```

```

    }

    // Update facet adaptation state.
    if (Core.isAdaptive)
        factic.UpdateFacetRelevance(clickCount);

    factic.InitializeInstances();

    factic.GeneratePageXML(contentHandler, pars);
} catch (Exception e)
{
    // TODO Auto-generated catch block
    e.printStackTrace();
}

// Log event.
factic.LogEvent(event, map);
}

```

D.3.2 Opis návrhu nástroja SemanticLog

Nástroj *SemanticLog* predstavuje webovú službu, resp. API určené na zaznamenanie údajov akciách používateľa do MySQL databázy. Implementáciu nástroja predstavuje trieda *SemanticLog* v balíku *sk.fiit.nazou.semanticlog* (Obrázok D-7), ktorá umožňuje zaznamenať udalosti, ich atribúty a stav zobrazenia v nástroji *Factic* (resp. inom) pomocou svojej verejnej metódy *LogEvent*.

Funkcionalitu nástroja možno rozdeliť na:

- Prijímanie požiadaviek nástrojov o zaznamenanie udalostí.
- Sledovanie sedení používateľa a vytváranie súvislých postupností udalostí pre jednotlivých používateľov (pre každé sedenie zvlášť).
- Vytvorenie záznamu o udalosti so sémantikou a jeho zapísanie do databázy.

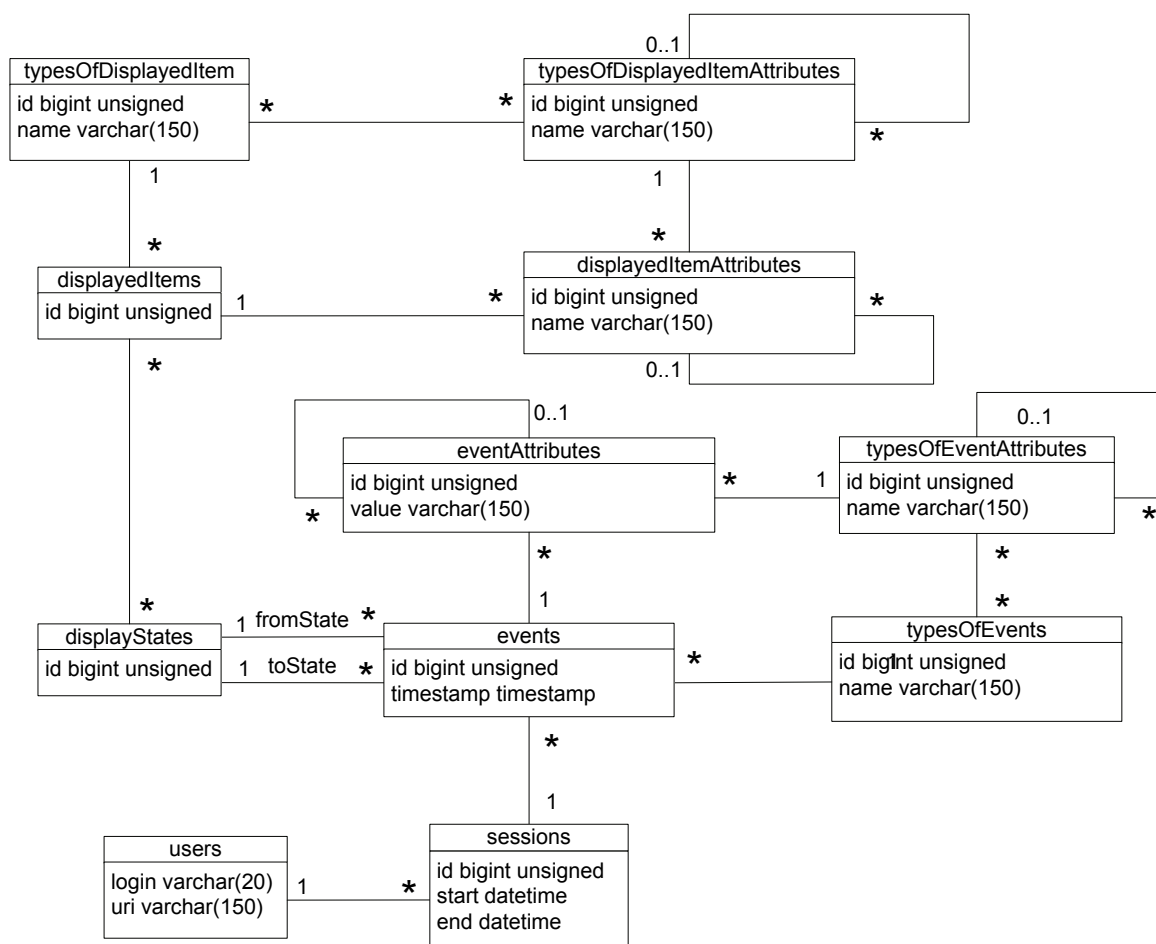
Počas svojej činnosti *SemanticLog* čaká na požiadavky iných nástrojov o zaznamenanie udalostí. Po príchode požiadavky vytvorí z údajov o udalosti záznam, ktorý priradí príslušnému používateľovi a príslušnému sedeniu. Súčasne naviaže záznam na predchádzajúci stav sedenia a zaznamená ho do relačnej databázy.

Zaznamenanie udalosti nástrojom *SemanticLog* predstavuje inicializácia objektového modelu záznamu a jeho následné uloženie do MySQL databázy pomocou objektovo-relačného mapovača Hibernate. Objektový model databázy je definovaný v knižnici *UserLogs.jar*, ktorú *SemanticLog* zdieľa s nástrojom *LogAnalyzer*.



Obrázok D-7: Diagram tried balíka *sk.fiit.nazou.semanticlog*.

Logický dátový model relačnej databázy, do ktorej *SemanticLog* ukladá záznamy o udalostiach v podobe ontológie udalostí znázorňuje **Error! Not a valid bookmark self-reference..** Jednotlivé udalosti sú zaznamenávané do tabuľky *events*, pričom ku každej je jednoznačne priradený typ udalosti (tabuľka *typesOfEvents*), používateľ (tabuľka *users*) a sedenie (tabuľka *sessions*) v ktorom udalosť nastala. Ku každej udalosti je tiež priradená dvojica stavov zobrazenia (tabuľka *displayStates*), ktoré zachytávajú stav prezentácie pred udalosťou a po nej.



Obrázok D-8|D}-9: Dátový model používaný na uloženia záznamov o udalostiach.

Zaznamenanie stavu zobrazenia pred a po udalosti je dôležité pre určenie charakteristík používateľa, pretože používateľ sa rozhoduje na základe informácií, ktoré sú mu prezentované systémom. Pre každý stav zobrazenia sa zaznamenávajú zobrazované objekty (tabuľka *displayedItems*), ich typy (tabuľka *typesOfDisplayedItems*), atribúty (tabuľka *displayedItemAttributes*) a typy atribútov (tabuľka *typesOfDisplayedItemAttributes*). Podobne možno zaznamenať atribúty aj pre jednotlivé udalosti (tabuľka *eventAttributes*) a ich typy (tabuľka *typesOfEventAttributes*).

Ako identifikátor používateľa je použité je prihlasovacie meno – login. Identifikátory konceptov a ich inštancií z ontológií – URI sú ukladané ako reťazce s max. dĺžkou 150 znakov.

Ukážka kódu metódy na záznam udalosti do databázy. Otvorenie sedenia Hibernate, načítanie poslednej udalosti pre aktuálne sedenie používateľa, aktualizácia sedenia používateľa, vytvorenie novej udalosti, príslušných atribútov a stavu zobrazenia. Finálne uloženie vytvorených objektov prostredníctvom sedenia Hibernate do MySQL databázy.

```

void LogEventGeneric(String sessionID, String eventURI, Map<String, Object[]> eventAttributes, Map<String,
Object[]> displayState)
{
    Session hibernateSession = HibernateUtil.getSessionFactory().getCurrentSession();
    hibernateSession.beginTransaction();

    // Retrieve last event for the user session.
    Event lastEvent = (Event) hibernateSession.get(Event.class, sessions.get(sessionID));
    if (lastEvent == null)
    {
        LogError("No event with the given ID exists! - " + sessions.get(sessionID).toString());
        logger.error("No event with the given ID exists!");
        lastEvent = (Event) hibernateSession.get(Event.class, sessions.get(sessionID));
        if (lastEvent == null)
        {
            LogError("No event with the given ID exists AGAIN! - " +
                sessions.get(sessionID).toString());
            Iterator i = hibernateSession.createQuery("from Event").list().iterator();
            while (i.hasNext())
            {
                Event e = (Event) i.next();
                LogMessage(new Long(e.getId()).toString());
            }
            return;
        }
    }

    // Update the end time of the session.
    UserSession userSession = lastEvent.getUserSession();
    userSession.setEnd(new Date());

    // Create and assign a new event.
    Event event = new Event();
    event.setTypeOfEvent(getEventType(eventURI));
    event.setFromState(lastEvent.getToState());
    event.setUserSession(userSession);

    // Store event attributes.
    Set<Entry<String, Object[]>> attributes = eventAttributes.entrySet();

    Iterator<Entry<String, Object[]>> i = attributes.iterator();
    while (i.hasNext())
    {
        Entry<String, Object[]> entry = i.next();
        LogEventGenericAttribute(event, entry.getKey(), entry.getValue());
    }

    // Create the new display state.
    DisplayState state = CreateDisplayState(displayState);
    event.setToState(state);
}

```

```

// Store displayed items.
Iterator<DisplayedItem> displayedItemIterator = state.getDisplayedItems().iterator();
while (displayedItemIterator.hasNext())
{
    DisplayedItem item = displayedItemIterator.next();
    item.save(hibernateSession);
    Iterator<DisplayedItemAttribute> attributeIterator =
        item.getDisplayedItemAttributes().iterator();
    while (attributeIterator.hasNext())
    {
        DisplayedItemAttribute att = attributeIterator.next();
        att.save(hibernateSession);
        SaveChildAttributes(att, hibernateSession);
    }
}

// Save the display state, session and event.
state.save(hibernateSession);
userSession.save(hibernateSession);
event.save(hibernateSession);

// Save attributes.
Iterator<EventAttribute> iterator = event.getEventAttributes().iterator();
while (iterator.hasNext())
{
    EventAttribute att = iterator.next();
    att.save(hibernateSession);
    SaveChildAttributes(att, hibernateSession);
}
hibernateSession.getTransaction().commit();

// Save sessionID to UserSession mapping.
sessions.put(sessionID, event.getId());

LogEventXML(eventURI, eventAttributes);
}

```

Príloha E Obsah elektronického média

- \Papers – príspevky na konferencie a workshopy
- \Prototype – súbory súvisiace s implementovaným prototypom
 - \Domain Ontology – použitá doménová ontológia
 - \Install – aplikácie tretích strán použité k overeniu metódy
 - \Tools – súbory prototypu (zdrojové, binárne, dokumentácia)
 - \Factic – súbory nástroja Factic
 - \Bin – skompilované súbory
 - \Doc – javadoc dokumentácia
 - \Lib – potrebné knižnice
 - \Src – zdrojové súbory
 - \SemanticLog – súbory nástroja SemanticLog
 - \Bin – skompilované súbory
 - \Doc – javadoc dokumentácia
 - \Lib – potrebné knižnice
 - \Src – zdrojové súbory
 - \User Ontology – použitá ontológia používateľa
- \Report – elektronická verzia tohto dokumentu
- \Resources – literatúra použitá k vypracovaniu projektu

Použitá literatúra

- Adkisson, H. P. (2005): Use of Faceted Classification, <http://www.webdesignpractices.com/navigation/facets.html> (10.12.2006)
- AGIMO (2004): Better Practice Checklist – Website Navigation. Department of Finance and Administration, Australian Government Information Management Office, 2004.
- Andrejko, A. – Barla, M. – Bieliková, M. – Tvarožek, M. (2006a): Softvérové nástroje pre získavanie charakteristík používateľa. In: P. Vojtáš, T. Skopal (Eds.): Proceedings of DATAKON '06, 139-148, Czech Republic, Brno, 2006, ISBN 80-210-4102-1.
- Andrejko, A. – Barla, M. – Tvarožek, M. (2006b): Comparing Ontological Concepts to Evaluate Similarity. In: Pavol Návrat et. al (Eds.): Proceedings of Tools for Acquisition, Organisation and Presenting of Information and Knowledge. Research Project Workshop Bystrá dolina, Nízke Tatry, Slovakia, September 29-30, 2006. Bratislava, STU, 71-78, 2006, ISBN 80-227-2468-8.
- Barla, M. (2006a): Interception of User's Interests on the Web. In: V. Wade, H. Ashman, and B. Smyth (Eds.): AH 2006, LNCS 4018, 435–439, 2006, Springer-Verlag Berlin Heidelberg, 2006.
- Barla, M. – Tvarožek, M. (2006b): Automatic Acquisition of Comprehensive Semantic User Activity Logs. In: Pavol Návrat et. al (Eds.): Proceedings of Tools for Acquisition, Organisation and Presenting of Information and Knowledge. Research Project Workshop Bystrá dolina, Nízke Tatry, Slovakia, September 29-30, 2006. Bratislava, STU, 169-174, 2006, ISBN 80-227-2468-8.
- Bieliková, M. a kol. (2005a): Modelovanie a získavanie, spracovanie a využívanie znalostí o konaní používateľa v hyperpriestore Internetu. Priebežná správa o riešení projektu APVT-20-007104, FIIT STU Bratislava, Jún 2005, 25s.
- Bieliková, M. – Grlický, V. – Kuruc, J. (2005b): Framework for Ontology-Based Information Presentation. In: Proc. of ITAT 2005 – Workshop on Theory and Practice of Information Technologies, P. Vojtáš (Ed.), September 2005, Račkova dolina, 325-334, 2005.
- Brusilovsky, P. (1996): Methods and techniques of adaptive hypermedia. In: User Modeling and User-Adapted Interaction, 6 (2-3), 87-129, 1996.
- Brusilovsky, P. (2001): Adaptive Hypermedia. User Modeling and User-Adapted Interaction. Ten Year Anniversary Issue (Alfred Kobsa, ed.), 11 (1/2), 87-110, 2001.
- Brusilovsky, P. (2004): Adaptive navigation support: From adaptive hypermedia to the adaptive Web and beyond. In: Psychology 2 (1), 7-23, 2004.

- De Bra, P. – Houben, G.J. – Wu, H. (1999): AHAM: a Dexter-based reference model for adaptive hypermedia. In: Proc. of the 10th ACM Conf. on Hypertext and Hypermedia, Hypertext'99, New York, NY, USA, ACM Press, 147–156, 1999.
- De Bra, P. – Aerts, A. – Houben, G.J. – Wu, H. (2000): Making General-Purpose Adaptive Hypermedia Work. Proceedings of the WebNet Conference, pp. 117-123, 2000.
- De Bra, P. – Aerts, A. – Berden, B. – de Lange, B. – Rousseau, B. – Santic, T. – Smits, D. – Stash, N. (2003a): AHA! The adaptive hypermedia architecture. In Proceedings of the 14th ACM Conference on Hypertext and Hypermedia (Nottingham, UK, August 26 - 30, 2003). HYPERTEXT '03. ACM Press, New York, NY, 81-84, 2003.
- De Bra, P. – Santic, T. – Brusilovsky, P. (2003b): AHA! meets Interbook, and more... In Richards, G. (Ed.), Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2003. 57-64. Chesapeake, VA: AACE.
- Domingue, J. – Dzbor, M. – Motta, E. (2004): Magpie: Supporting Browsing and Navigation on the Semantic Web. Intelligent User Interfaces, 191-197, 2004.
- Eklund, J. – Brusilovsky, P. – Schwarz, E. (1997): Adaptive Textbooks on the WWW. In: H. Ashman, P. Thistewaite, R. Debreceeny and A. Ellis (eds.) Proceedings of AUSWEB97, The Third Australian Conference on the World Wide Web, Queensland, Australia, July 5-9, 1997, Southern Cross University Press, 186-192, 1997.
- Filkorn, R. – Barla, M. – Bartalos, P. – Sivák, P. – Szobi, K. – Tvarožek, M. (2006): Ontology as an Information Base for a Family of Domain Oriented Portal Solutions. In: Proc. of 15th Int. Conf. on Information Systems Development, ISD'06, Budapest University of Technology and Economics, August 31-September 2, 2006, 11 s., zborník vyjde v r. 2007.
- Grlický, V. (2006): Introduction to semantic web. 2006. PeWe Group. <http://www2.fiit.stuba.sk/~andrejko/pewe/data/2006-2007/semweb-grlicky.pps> (10.12.2006)
- Guha, R. – McCool, R. – Miller, E. (2003): Semantic Search. In Proc. of the 12th International conference on World Wide Web, 700-709, 2003.
- Gulli, A. – Signorini, A. (2005): The indexable web is more than 11.5 billion pages. In Special interest Tracks and Posters of the 14th international Conference on World Wide Web (Chiba, Japan, May 10-14, 2005). WWW '05. ACM Press, New York, 902-903, 2005.
- He, P. – Koh, S. – Lai, K. – Wang, S. (2003): Faceted Classification. (2003) <http://www.slais.ubc.ca/courses/libr517/02-03-wt2/projects/faceted/adv.htm> (6.5.2006)
- Horridge, M. – Knublauch, H. – Rector, A. – Stevens, R. – Wroe1, C. (2004): A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools.

- Hughes, S. – Brusilovsky, P. – Lewis, M. (2002): Adaptive Navigation Support in 3D E-Commerce Activities. In: Proceedings of Workshop on Recommendation and Personalization in eCommerce at the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002), 132-139, 2002.
- Instone, K. (2004): How user interfaces represent and benefit from a faceted classification system, SOASIST, 2004.
<http://user-experience.org/uefiles/facetedbrowse/KI-FB-SOASIST.pdf> (7.5.2006)
- Kostelník, R. – Bieliková, M. (2003): Web-Based Environment using Adapted Sequences of Programming Exercises. In Proc. of Information Systems Implementation and Modeling - ISIM 2003. M. Beneš (Ed.). MARQ Ostrava, Brno, April 28-30, 33-40, 2003, ISBN 80-85988-84-4.
- Kruk, S.R. (2005): Social Semantic Search and Browsing. DERI Research Seminar 2005 Galway, Ireland.
- Levene, M. – Wheeldon, R. (2003): Navigating the World-Wide-Web. 2003. In: M. Levene, A. Poulovassilis (Eds.): Web Dynamics, Springer, 2003.
- Návrát, P. – Bartoš, P. – Bieliková, M. – Hluchý, L. – Vojtáš, P. (2006): Tools for Acquisition, Organization and Presenting of Information and Knowledge, Research Project Workshop, Bystrá Dolina, Low Tatras, Slovakia, 256 pages, 2006.
- Shah, U. – Finin, T. – Joshi, A. – Cost, R. – Mayfield, J. (2002): Information retrieval on the semantic web. In Proc. of the 11th Int. Conference on Information and Knowledge Management (CIKM2002), 461-468, 2002.
- Studer, R. – Benjamins, V. – Fensel, D. (1998): Knowledge engineering: Principles and methods. IEEE Transactions on Data and Knowledge Engineering, 25:161-197, 1998, Elsevier Science B.V.
- The Knowledge Management Connection, (2006): Faceted Classification of Information
<http://www.kmconnection.com/DOC100100.htm> (10.12.2006)
- Timberlake, S. (2000): The Basics of Navigation,
<http://www.efuse.com/Design/navigation.html#StylesofNavigation> (10.12.2006)
- Tvarožek, M. (2006a): Personalized Navigation in the Semantic Web. In: Bieliková, M. (ed.): Proc. of IIT.SRC 2006: Student Research Conference, apríl 2006, Vydavateľstvo STU v Bratislave, 145-152, 2006, ISBN 80-227-2395-9.
- Tvarožek, M. (2006b): Personalized Navigation in the Semantic Web. In: V. Wade, H. Ashman, and B. Smyth (Eds.): AH 2006, LNCS 4018, 467–471, 2006. Springer-Verlag Berlin Heidelberg 2006.

- Tvarožek, M. (2006c): Personalized Navigation in the Semantic Web. In: Mannová, B. et al. (Eds.): Proc. of 4th Student Research Competition: ACM.SRC 2006, November 2006, Prague, 73-80, 2006, ISBN 80-01-03595-6.
- Tvarožek, M. (2006d): Personalized Navigation in the Semantic Web. In: Pavol Návrat et. al (Eds.): Proceedings of Tools for Acquisition, Organisation and Presenting of Information and Knowledge. Research Project Workshop Bystrá dolina, Nízke Tatry, Slovakia, September 29-30, 2006. Bratislava, STU, 182-192, 2006, ISBN 80-227-2468-8.
- Tvarožek, M. – Barla, M. – Bieliková, M. – Grlický, V. – Andrejko, A. – Filkorn, R. – Bartalos, P. (2006e): Presentation and Personalization of Information in the Semantic Web. In: Pavol Návrat et. al (Eds.): Proceedings of Tools for Acquisition, Organisation and Presenting of Information and Knowledge. Research Project Workshop Bystrá dolina, Nízke Tatry, Slovakia, September 29-30, 2006. Bratislava, STU, 201-207, 2006, ISBN 80-227-2468-8.
- Tvarožek, M. – Barla, M. – Bieliková, M. (2007): Personalized Presentation in Web-Based Information Systems. In: Jan van Leeuwen et al. (Eds.): SOFSEM 2007, LNCS 4362, pp. 796–807, 2007, Springer-Verlag Berlin Heidelberg 2007.
- Wikipedia (2006a): Semantic Web, http://en.wikipedia.org/wiki/Semantic_web (10.12.2006).
- Wikipedia (2006b): World Wide Web, http://en.wikipedia.org/wiki/World_Wide_Web (10.12.2006)
- Wynar, B. S. – Taylor, A. G. (1992): Introduction to cataloging and classification, 8th edition, 320-321, 1992, Libraries Unlimited Inc.
- Zhang, L. – Yu, Y. – Zhou, J. – Lin, CX. – Yang, Y. (2005): An Enhanced Model for Searching in Semantic Portals. In Proc. of WWW 2005, 453-462, 2005, ACM Press.