

Tvorba nezávislého rozhrania pre ontologickú organizačnú pamäť

René Pázman

Softec s.r.o.

rene.pazman@softec.sk

<http://www.softec.sk/>, <http://rene.pazman.googlepages.com/>

Abstrakt V rámci projektu na vývoj nástrojov na spracovanie informácií a znalostí z heterogénnych zdrojov používame organizačnú pamäť ako centrálnu úložisko dát. V príspevku opisujeme náš prístup pri tvorbe rozhrania na ontologickú časť organizačnej pamäte a použitie návrhových vzorov pri jeho implementácii. Zameriavame sa na vytvorenie vrstvy pre taký prístup k ontológii, ktorý je nezávislý od implementácie ontologického úložiska.

Kľúčové slová RDF, ontológia, ontologické úložisko, Sesame, návrhové vzory

1 Úvod

V projekte NAZOU ([4]) riešime vývoj nástrojov pre získavanie, organizovanie a udržiavanie znalostí v prostredí heterogénnych informačných zdrojov. Ide o štátnu úlohu výskumu a vývoja, ktorá je približne v dvoch tretinách plánovaného trvania.

Cieľom projektu je navrhnúť metódy na získavanie, organizovanie, udržiavanie a poskytovanie informácií z internetu a tieto metódy implementovať formou nástrojov. Nástroje sú v projekte testované a vyhodnocované v pilotných aplikáciách.

Spracovávané informácie sa týkajú ohraničenej informačnej domény. Pre otestovanie nástrojov v rámci pilotných aplikácií sme zvolili doménu pracovných ponúk (zbieranie pracovných ponúk na internete a poskytovanie informácií o nich potenciálnym uchádzačom o prácu).

Informácie získané z vybraných zdrojov sú uložené v systéme. Pre tento účel systém (pilotná aplikácia) obsahuje úložisko dát a informácií. Informácie pre používateľov sú poskytované z tohto úložiska. Cieľová forma reprezentácie informácií sú štruktúrované dáta — ontológia.

2 Organizačná pamäť v projekte

Pôvodný návrh architektúry pilotnej aplikácie predpokladal tesnú funkčnú spoluprácu nástrojov realizovaných pomocou priamych funkčných prepojení. Pre návrh

takejto architektúry sa zvolila forma viacerých pohľadov na aplikáciu. Boli navrhnuté jej podsystémy, prípady použitia a komponenty. Informačná doména sa zovšeobecnenila na doménu ponúk s archetypmi aktérov a dát — ponuka, producent a konzument ([5]). Návrh predpokladal doménovo špecifický prístup k údajom, teda niektoré funkčné rozhrania boli závislé od zvolenej (aj keď zovšeobecnenej) domény.

Takýto návrh architektúry sa ukázal v našom projekte priamo nepoužiteľný. Dôvodov bolo viacero, najmä geografická distribuovanosť vývoja, nehomogénnosť znalostí členov tímu a ich časovej alokácie v projekte, neexistencia detailného návrhu cieľovej funkčnosti aplikácie a potreba opakovaného použitia nástrojov v iných doménach a v inej vzájomnej zostave.

Aktuálna pilotná aplikácia je vytvorená na voľnejšom základe. Ide o architektúru štýlu tabuľa (blackboard), kde nástroje sú samostatné jednotky, pracujúce nad organizačnou pamäťou a komunikujúce najmä cez organizačnú pamäť. Takéto uvoľnenie väzieb medzi nástrojmi umožnilo o.i. aj lepšiu distribuovateľnosť prác v projekte a ťažisko prác sa mohlo presunúť z riešenia integračných konfliktov na návrh a overovanie metód spracovania informácií. Organizačná pamäť sa týmto stala dôležitým centrálnym prvkom.

Organizačná pamäť sa v projekte NAZOU skladá z troch vrstiev: interakčnej vrstvy na vzdialený prístup k pamäti, manipulačnej vrstvy s programátorskými rozhraniami pre nástroje a fyzickej vrstvy, v ktorej sú jednotlivé úložiská. Fyzická vrstva je realizovaná úložiskami troch typov — súborovým, relačným a ontologickým.

Pre realizáciu ontologického úložiska bol zvolený systém Sesame ([1]). V ďalšom texte opisujeme náš prístup pri návrhu rozhrania na ontologické úložisko.

3 Ontologické rozhranie

Pri návrhu rozhrania na ontologické úložisko sa brali do úvahy viaceré požadované vlastnosti:

1. *Nezávislosť od informačnej domény.* Aj keď v pilotnej aplikácii používame doménu pracovných ponúk, jadro nástrojov je implementované nezávisle od informačnej domény — z toho dôvodu, aby sa nástroj dal použiť aj v inej doméne. Takto sa navrhla aj architektúra nástroja, keď sa upustilo od pôvodnej architektúry systému.
2. *Nezávislosť od úložiska.* Nástroje nemajú byť pevne viazané na zvolené úložisko Sesame, ale mali by byť bez väčších zmien použiteľné aj s inými úložiskami (napr. Jena).
3. *Jednoduchosť.* Rozhranie by malo byť jednoduché a zrozumiteľné pre autorov nástrojov.
4. *Zachovanie výkonnosti.* Rozhranie nesmie zhoršiť výkonnosť úložiska, teda jeho použitie nesmie znížiť odozvu úložiska oproti použitiu natívneho rozhrania úložiska.

5. *Univerzálnosť*. Rozhranie má poskytovať všetky potrebné možnosti úložiska, najmä na vyhľadávanie pomocou špecializovaných dopytovacích jazykov, ale má tiež umožňovať prácu s individuami, RDF grafmi a pod.

Rozhranie na ontológiu tvoria tri vrstvy. V spodnej vrstve je natívne rozhranie úložiska, v našom prípade Sesame API, ktoré je tvorené Java rozhraniami (interfaces) a triedami a je súčasťou úložiska Sesame.

Vrchná vrstva je definíciou nezávislého rozhrania, ktoré tvorí fasádu ontologického úložiska pre jeho používateľov (nástroje). Táto vrstva je tvorená výhradne Java rozhraniami. Nástroje pre svoju prácu poznajú iba túto vrstvu.

Stredná vrstva spája tieto dve vrstvy — adaptuje natívne rozhranie úložiska (spodnú vrstvu) na nezávislé rozhranie pre nástroje (vrchnú vrstvu). Je tvorené Java triedami, ktoré realizujú Java rozhrania z vrchnej vrstvy a používajú Java rozhrania a triedy zo spodnej vrstvy.

V prípade potreby použitia iného ontologického úložiska (ako napr. Jena) sa spodná vrstva vymení za natívne rozhranie úložiska (Jena API) a vytvorí sa stredná vrstva pre nové úložisko — tú je treba implementovať nanovo. Vrchná vrstva sa nezmení, čo umožní zmenu úložiska bez zmeny implementácie nástrojov.

Vzhľadom na túto situáciu sme pre implementáciu rozhrania použili najmä návrhový vzor *Adapter* ([3]), známy aj ako *Wrapper*, obalovač. Pre realizáciu dopytovania pomocou špecializovaných dotazovacích jazykov (ako napr. SeRQL a RDQL) sa použil tento návrhový vzor 4-krát. Jedno použitie vzoru umožnilo obaliť spojenie na ontologické úložisko a ďalšie tri použitia zabezpečujú prístup k výsledkom dopytov (pre výslednú tabuľku, riadok výslednej tabuľky a bunku takéhoto riadku).

Okrem tohto základného vzoru sa použili aj ďalšie vzory:

- *Abstract Factory* ([3]). Vzor je použitý pri vytváraní spojenia na ontologické úložisko. Použitie vzoru umožňuje dynamickú zmenu implementácie rozhrania ontologického úložiska (umožňuje prechod medzi Sesame a Jena). Výber implementácie rozhrania je riadený nastavením v konfiguračnom súbore.
- *Singleton* ([3]). Vzor sa používa pri vytváraní spojenia na ontologické úložisko pri testovaní.
- *Dependency Injection* ([2]). Tento vzor sa využíva v projekte pre takú vzájomnú integráciu nástrojov, ktorá umožňuje voľné a konfigurovateľné väzby medzi nimi. Vzor je možné použiť aj pre vytvorenie spojenia na ontologické úložisko.

4 Záver

Navrhnuté a implementované rozhranie spĺňa stanovené požiadavky — je nezávislé od úložiska a informačnej domény, jednoduché a prehľadné a efektívne (napr. netransformuje zbytočne výsledky vyhľadávania). Poskytuje potrebné funkcie na dopytovanie ontológie a na prácu s výrokmi (statements), RDF grafmi a inštanciami (individuami).

Rozhranie je v projekte realizované pre Sesame, ale dá sa implementovať aj pre iné úložisko, napr. Jena. Rozhranie v súčasnosti poskytuje základné funkcie pre prácu s ontológiou. Plánujeme ho rozširovať o ďalšie funkcie, ktoré Sesame poskytuje.

Uvažujeme tiež s jeho rozširovaním o funkcie na udržiavanie dát a propagovanie zmien v údajoch medzi nástrojmi. Tieto funkcie by mali byť doménovo nezávislé.

Okrem toho zvažujeme vytvoriť nad ním doménovo špecifickú nadstavbu, ktorá by umožňovala jednotnú prácu s doménovými objektmi. V tomto prípade by však doménová závislosť mala ostať na úrovni pojmov znovupoužiteľných v iných príbuzných doménach, ako sú napr. pojmy ponuka, poskytovateľ ponuky, záujemca o ponuku, všeobecné preferencie záujemcu o ponuku a pod.

Referencie

1. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: An Architecture for Storing and Querying RDF Data and Schema Information. In: Fensel, D., Hendler, J., Lieberman, H., Wahlster, W. (Eds): *Semantics for the WWW*, MIT Press. Available at <http://www.cs.vu.nl/frankh/postscript/MIT01.pdf>. (2001)
2. Fowler, M.: Inversion of Control Containers and the Dependency Injection pattern. Available at <http://www.martinfowler.com/articles/injection.html>. (2004)
3. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Professional Computing Series. (1995)
4. Návrat, P., Bieliková, M., Rozinajová, V.: Methods and Tools for Acquiring and Presenting Information and Knowledge in the Web. In: Rachev, B., Smrikarov, A. (Eds.): *CompSysTech 2005*, Varna, Bulgaria. pp. IIB.7.1–IIB.7.6. (2005)
5. Vranič, V., Marko, V.: Developing a Product-Line Based Architecture in a Domain Under Research. In: Návrat, P., Bartoš, P., Bieliková, M., Hluchý, L., Vojtáš, P. (Eds.): *Tools for Acquisition, Organisation and Presenting of Information and Knowledge, Proceedings in Informatics and Information Technologies, Research Project Workshop, Bystrá dolina, Nízke Tatry, Slovakia*. pp. 211–222. (2006)