

Reprezentácia ontológií relačným modelom a realizácia dopytovania

Richard Veselý, Mária Bieliková

Ústav informatiky a softvérového inžinierstva,
Fakulta informatiky a informačných technológií,
Slovenská Technická Univerzita, Ilkovičova 3, SK 842 16 Bratislava
{vesely,bielik}@fiit.stuba.sk

Abstrakt. V príspevku sa venujeme návrhu nového prístupu k reprezentácii ontológií vo fyzickom úložisku postavenom na relačnej databáze. Návrh kombinuje niekoľko známych prístupov a má za cieľ umožniť škálovateľnosť s veľkým množstvom dát ako aj zefektívniť prístup k dátam a ich manipuláciu. Súčasťou návrhu je snaha nájsť všeobecnejšiu jazykovú reprezentáciu umožňujúcu dopytovanie nad dátami obsiahnutých v úložisku s expresívnosťou dostatočnou na obsiahnutie vyjadrovacej schopnosti existujúcich ontologických dopytovacích jazykov. Účelom tejto reprezentácie je zvýšiť mieru abstrakcie medzi zdrojovou reprezentáciou (existujúce jazyky) a cieľovou reprezentáciou (jazyk fyzického úložiska) a poskytnúť dostatočnú flexibilitu a možnosť objasneného prekladu medzi jednotlivými reprezentáciami.

Kľúčové slová: ontológia, úložisko, relačná databáza, dopytovanie, preklad

1 Úvod

V súčasnosti sa výskum čoraz viac zameriava na oblasť efektívnej práce s informáciami, najmä vyhľadávanie a odporúčanie v rozsiahlych a meniacich sa informačných priestoroch akým je aj Internet a jeho služba World Wide Web. Pod efektívnou prácou máme na mysli najmä efektívnosť z pohľadu človeka – používateľa, ktorá sa vzhľadom na rozsah informačných priestorov dá dosiahnuť automatizáciou činností spojených so získavaním informácií, ich analýzou, organizovaním a prezentáciou či odporúčaním. Aby stroje mohli pracovať s informáciami potrebujeme k dátam (informačnému obsahu) priradiť sémantiku v špecifickom kontexte. Na opisanej myšlienke je postavený princíp webu so sémantikou, v rámci ktorého v súčasnosti prebieha aktívny výskum, vznikajú webové aplikácie, ktoré pracujú s metadátami reprezentovanými v ontológiách. V tejto súvislosti sa do popredia dostáva otázka ako ukladať a efektívne pristupovať k čoraz väčšiemu množstvu metadát.

Pre reprezentáciu v ontológiách vzniklo niekoľko špecifických jazykov založených na XML, pričom medzi najčastejšie používané patria RDF a OWL založené na štandardoch a odporúčaní¹. Tieto všeobecné textové reprezentácie v XML sú však neškálovateľné a neefektívne, pri dopytovaní je spravidla nutné

¹ World Wide Web Consortium (W3C). Resource Description Framework (RDF) - www.w3.org/RDF/; OWL Web Ontology Language Reference - www.w3.org/TR/owl-ref/

prechádzať všetky dáta sekvenčne a nie je možné takéto reprezentácie zdieľať pre prístup viac používateľov.

Opísané problémy s reprezentáciou ontológií sa prejavujú najmä s rastúcim počtom metadát. Preto sa začali hľadať spôsoby ako zvýšiť efektívnosť prístupu a manipulácie s dátami a zabezpečiť dostatočnú škálovateľnosť aj pre veľké objemy dát a metadát.

V tomto príspevku opisujeme návrh nového prístupu k reprezentácii ontológií vo fyzickom úložisku postavenom na relačnej databáze. Inovácia spočíva v modelovaní reprezentácie ontológie podľa jazyka OWL a navrhuje univerzálnejší dopytovací mechanizmus.

2 Súčasné ontologické úložiská

V súčasnosti už existuje niekoľko otvorených aj komerčných produktov, ktoré realizujú ontologické úložiská, napr. Sesame, Jena, RDFSuite, Kowari [1], O3F [2], InKling, rdfDB, RDFStore, EOR, Redland, RDF Gateway, TRIPLE, KAON, Cerebra, Empolis K42, Ontopia Knowledge Suite a iné ([3], [4]). Väčšina týchto riešení poskytuje úložisko pre ontologické dáta, najčastejšie pamäťové alebo perzistentné, pričom umožňuje s dátami pracovať pomocou špeciálne navrhnutých nástrojov alebo aplikačných programovacích rozhraní. Stručne opíšeme niekoľko najznámejších.

Sesame. Sesame, riešenie postavené na RDFS repozitári umožňujúcom dopytovanie bol pôvodne vyvinutý firmou Administrator Nederland bv ako jeden z kľúčových výstupov pre šiesty rámcový program Európskej Únie v rámci projektu On-To-Knowledge.

Systém pozostáva z repozitáru, dopytovacieho rámca a administratívneho modulu pre pridávanie a mazanie RDF dát a informácií týkajúcich sa schémy. Podporuje expresívne dopytovanie nad RDF dátami a schémou pomocou upraveného jazyka RQL (SeRQL), nepodporuje však všetky vlastnosti RQL definované v špecifikácii jazyka. Taktiež podporuje základné odvodzovanie ako tranzitívnosť podtried a podvlastností. Súčasná verzia 2.0-rc2 (30.11.2007) už podporuje kompletné dopytovanie jazykom SPARQL.

Jena. Vyvinutá firmou Hewlett-Packard, Jena predstavuje kolekciu RDF nástrojov napísaných v Jave, ktoré zahŕňajú: Java API pre modely a grafy, RDF parser (podporujúci filter pre RDF/XML, N3 a N-Triples), dopytovanie pomocou RDQL, ARQ (jazyk SPARQL rozšírený o počítanie, GROUP BY, podpora návrhu SPARQL/Update a ďalšie), podpora tried pre DAML+OIL ontológie a pamäťové alebo perzistentné úložisko postavené na BerkeleyDB alebo inej implementácii.

Vďaka abstrakcii nad úložiskom umožňuje Jena integrovať nové úložiskové subsystémy. Aktuálna implementácia nepodporuje odvodzovanie, pretože ho neumožňuje použitý dopytovací jazyk RDQL. Staršia verzia nepodporovala odvodzovanie, aktuálna implementácia však zahŕňa bohaté odvodzovanie pomocou všeobecného pravidlového odvodzovača, ale aj jednoduché tranzitívne odvodzovanie a vstavané pravidlá pre RDFS a časť OWL/Lite.

RDF Suite. Vyvinutý v rámci šiesteho rámcového programu Európskej Únie, čiastočne podporovaný projektami C-Web a MesMuses, predstavuje zostavu nástrojov pre správu RDF metadát a sústreďuje sa na spracovanie RDF metadát pre webové aplikácie veľkého rozsahu. Pozostáva z nástrojov na parsovanie, validáciu, ukladanie a dopytovanie sa nad RDF dátami, konkrétne sa jedná o nástroje Validating RDF Parser (VRP), RDF Schema Specific DataBase (RSSDB) a RDF Query Language (RQL).

Hlavným cieľom RSSDB je separácia informácií RDF schémy od informácií dát a tiež rozlišovanie medzi unárnymi a binárnymi reláciami, ktoré majú inšancie tried a vlastností. Dopytovanie nad uloženými RDF dátami je zabezpečené dopytovacím modulom, ktorý implementuje jazyk RQL. Z výkonnostných dôvodov sa modul snaží delegovať čo najviac dopytov na fyzické úložisko. RQL modul je jednoducho integrovateľný s webovými aplikačnými servermi a komerčnými ORDBMS. Projekt nebol už niekoľko rokov aktualizovaný a pravdepodobne je neaktívny.

Redland. Redland je knižnica vyvinutá na Bristolskej univerzite, ktorá poskytuje vysoko-úrovňové rozhranie pre ukladanie, dopytovanie a manipuláciu s RDF modelmi. Každý koncept RDF modelu zapúzdruje Redland vo vlastnej triede a poskytuje preto objektovo-orientované API. Niektoré triedy poskytujúce parsery, mechanizmy na ukladanie a iné elementy sú postavené ako moduly, ktoré sa dajú podľa potreby pridávať alebo odoberať.

Redland poskytuje modulárnu objektovo-orientovanú knižnicu napísanú v C, Perle, Pythóne a Tcl a tiež Javovské rozhranie pre manipuláciu s RDF modelom a jeho časťami, poskytuje tiež parser na čítanie RDF/XML a inú syntax (DAML+OIL a N-Triplety) a úložiskové mechanizmy pre modely v pamäti aj na disku prostredníctvom Sleepycat/Berkeley DB.

Vlastnosti jednotlivých ontologických úložísk sú zhrnuté v tabuľke 1.

Tabuľka 1. Porovnanie vlastností vybraných ontologických úložísk

Názov	Dopytovací jazyk	Typ úložiska	Odvodzovanie	Udržia vanie	Exportný formát
Sesame	RQL (SeRQL ²)	ORDBMS	áno	áno	RDF
Jena	RDQL	Pamäťové alebo RDBMS	nie	áno	Trojice v ASCII
RDF Suite	RQL	ORDBMS	áno	áno	RDF
Redland	Triple-matching	Pamäťové (Sleepycat), perzistencia (Berkeley DB)	nie	áno	Trojice

Základným problémom spomínaných riešení je rýchlosť a škálovateľnosť. Dôvodom návrhu nového prístupu bola najmä snaha zlepšenia výkonnosti riešenia pre veľké

² openRDF.org. Sesame RQL Query Language - www.openrdf.org/doc/sesame/users/ch06.html

množstvo poznatkov v ontológii. Druhotným problémom je pomerne veľký počet komunitou používaných dopytovacích jazykov a preto bolo tiež snahou zabezpečiť univerzálnejší mechanizmus dopytovania.

Dopytovanie a preklad

Dopytovanie sa nad ontológiami patrí medzi štandardnú funkcionálnu existujúcich riešení a pre tento účel sa vyvinulo niekoľko dopytovacích jazykov, napríklad RQL, SeRQL, SPARQL, RDQL, RDFQL, SquishQL, RDFPath, Versa, Triple, DAML+OIL, TMQL, Tolog. Samotný preklad funguje ako most medzi vrstvami úložisk, pričom transformuje nejakú zdrojovú reprezentáciu na cieľovú, najčastejšie ontologický dopytovací jazyk na dopytovací jazyk fyzického úložiska.

Pri preklade možno klásť dôraz na zdrojovú reprezentáciu, napr. formulácia dopytov v prirodzenom jazyku alebo možno klásť dôraz na cieľovú reprezentáciu, kedy potrebujeme preklad na konkrétnu fyzickú reprezentáciu, najčastejšie SQL.

S prekladaním a transformáciou reprezentácií je spojených niekoľko problémov. Základným problémom je odlišná vyjadrovacia sila jazykov, ktorá sa môže prejaviť napríklad pri preklade zložitejších jazykových konštrukcií (napr. OPTIONAL blok v SPARQL [5], problémy s touto konštrukciou diskutujeme ďalej). Problematické je tiež prekladanie jazykov s diametrálne odlišnou formou reprezentácie (grafy/stromy voči relačnej algebre).

Analýza jazykových konštrukcií

Transformáciou ontologických dopytovacích jazykov na relačné sa zaoberalo viacero autorov. Ako najčastejší problém sa uvádza preklad grafového vzoru „optional“ (umožňuje vykonanie nepovinnnej vetvy dopytu podmienené schopnosťou naviazať vo vetve použité premenné), ktorý sa vyskytuje napr. v jazykoch SPARQL, SeRQL a RDFQL. Považa problém spočíva v zložitosti, ktorá zahŕňa zdieľané dynamicky viazané premenné a možnosť použitia paralelného a vnoreného vzoru definovaných v špecifikácii jazykov.

Existuje viacero problémov tejto transformácie [5]:

- Základná sémantika vzoru OPTIONAL – vykonanie bloku OPTIONAL nemusí byť nutne úspešné a v prípade zlyhania sa v dopyte SELECT vrátia nenaviazané prázdne premenné.
- Sémantika zdieľaných premenných – vo všeobecnosti musia mať zdieľané premenné naviazané rovnaké hodnoty, premenné môžu byť zdieľané medzi subjektami, predikátmi, objektami a ľubovoľne navzájom.
- Sémantika paralelných vzorov OPTIONAL – zatiaľ čo zlyhanie jedného OPTIONAL bloku nezabraňuje vykonaniu ďalšieho, v prípade úspešného vyhodnotenia dôjde k naviazaniu premenných, čím sa pri vyhodnotení ďalšieho OPTIONAL bloku vynúti naviazanie rovnakých hodnôt.
- Sémantika vnorených vzorov OPTIONAL – predtým, než sa vyhodnotí blok OPTIONAL, musia byť úspešne vyhodnotené všetky základné grafové vzory a ďalšie OPTIONAL bloky, ktoré obsahuje.

Stratégie prekladu

Štandardný postup pri preklade z ontologického dopytovacieho jazyka na relačný je použitie gramatiky (ideálne atribútovej) na spracovanie a transformáciu vstupu na žiadanú výstupnú podobu. V prípade zložitejších jazykových konštrukcií je nutné aplikovať prekladovú stratégiu, ktorá najčastejšie predstavuje špecifický algoritmus.

Použitie prekladovej stratégie vyžadujú najmä komplexné grafové vzory, akým je blok OPTIONAL. Pre tento prípad existuje niekoľko riešení ([5], [6], [7], [8]), no väčšina neumožňuje preklad vnorených blokov. Chebotko a kol. navrhli údajne prvú stratégiu, schopnú prekladu ľubovoľne komplexných OPTIONAL blokov [5]. Ich riešenie sa vzťahuje na jazyk SPARQL a poskytujú dve implementácie – BGPtoSQL pre základné grafové vzory a SPARQLtoSQL pre komplexné grafové vzory.

3 Návrh reprezentácie

Navrhnutá reprezentácia ontológie predstavuje kombinovaný hybridno-špecifický prístup. Logický model reflektuje základné entity jazyka OWL a zachytáva entity a relácie špecifikácie OWL Lite a podmnožinu špecifikácie OWL DL³. Ukazuje sa, že najrozšírenejší prístup reprezentácie pomocou vertikálnej tabuľky je neadekvátny pre objemy ontologických dát rádovo miliónov trojíc. Hybridno-špecifický prístup bol zvolený z výkonnostných dôvodov a predstavuje kompromis medzi všeobecnosťou reprezentácie a efektívnosťou prístupu k dátam v nej obsiahnutých. Model je zároveň navrhnutý tak, aby optimalizoval prístup k dátam, ktorých sa týkajú najčastejšie vykonávané dopyty. Navrhnutá reprezentácia separuje schému ontológie a inštančné dáta, zároveň delí inštančné dáta na tri základné skupiny – metadáta spojené s inštanciou, jej objektové vlastnosti a dátové vlastnosti.

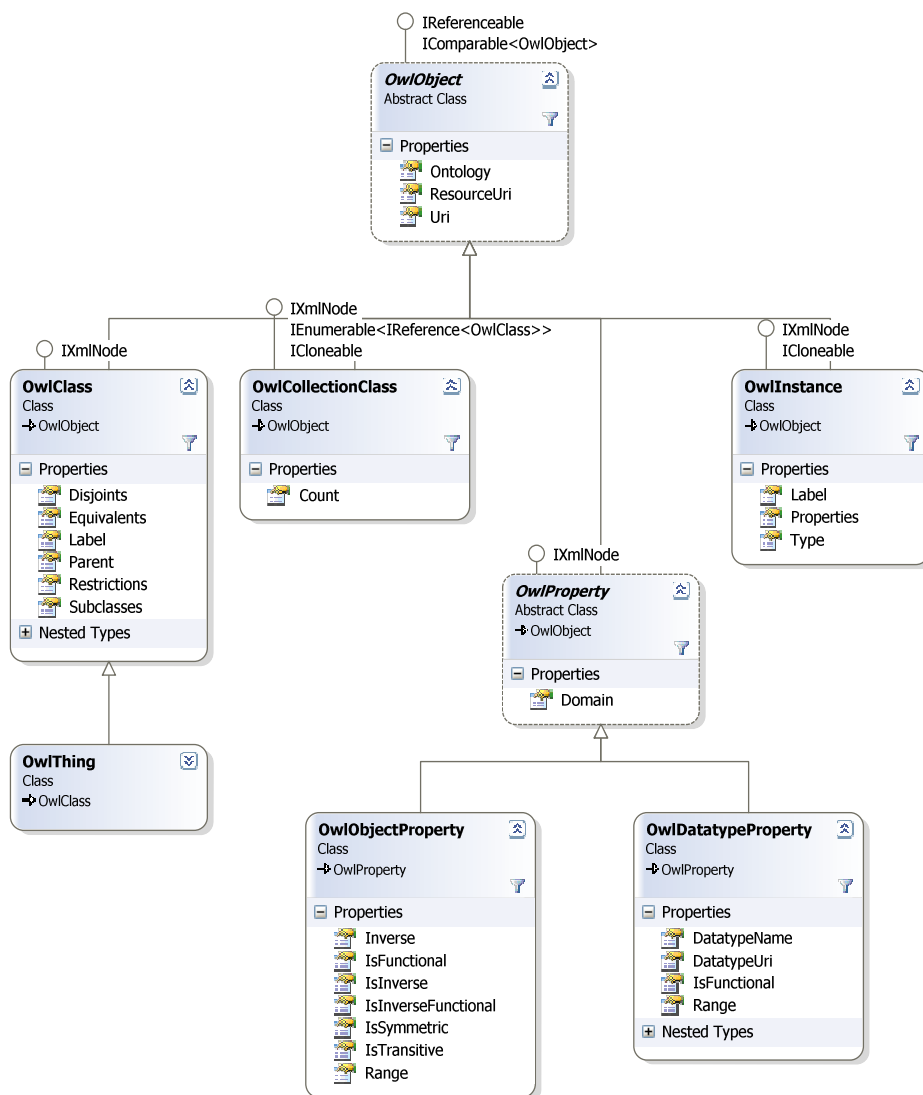
Základnou entitou navrhnutého logického modelu dát je entita *Type*, ktorá predstavuje entitu *owl:Class* jazyka OWL. Táto obsahuje informáciu o svojom predkovi, vďaka čomu je možné rýchlo zostrojiť strom dedičnosti, keďže je tabuľka tried oddelená od ostatných dát v ontológii. Medzi entity opisujúce schému ontológie patria aj *ObjectProperty* a *DatatypeProperty*, ktoré predstavujú objektové vlastnosti a dátové vlastnosti triedy, ku ktorým sa viaže niekoľko relácií, pre objektové vlastnosti symetričnosť, tranzitívnosť, funkcionálnosť (vlastnosť môže nadobúdať iba jednu hodnotu) a inverzná funkcionálnosť, zatiaľ čo dátové vlastnosti majú definovanú iba funkcionálnosť. Obe majú definované domény, ktorých sa týkajú, reprezentované entitami *ObjectDomain* a *DatatypeDomain* a rozsah platných hodnôt v podobe *ObjectRange* a *DatatypeRange*.

Inštančné dáta sú separované od zvyšku dát v ontológii v podobe niekoľkých entít. Hlavnou je entita *Instance*, ktorá obsahuje svoj typ a ďalšie inštančné metadáta. Hodnoty jej objektových a dátových vlastností sú zastúpené entitami *ObjectPropertyReference* a *DatatypePropertyReference*.

³ World Wide Web Consortium (W3C). OWL Web Ontology Language Reference - www.w3.org/TR/owl-ref/

Každá entita v modeli má svoj globálny unikátny identifikátor, ktorý je použitý aj pri zachovaní dátovej integrity modelu.

Súčasťou návrhu riešenia je aplikačné programovacie rozhranie, ktoré poskytuje knižnicu s objektovým modelom reprezentujúcim ontológiu, navrhnutým podľa vyššie spomenutého opisu reprezentácie – reflektuje entity jazyka OWL. Prehľad základných tried modelu je znázornený na obrázku 1.

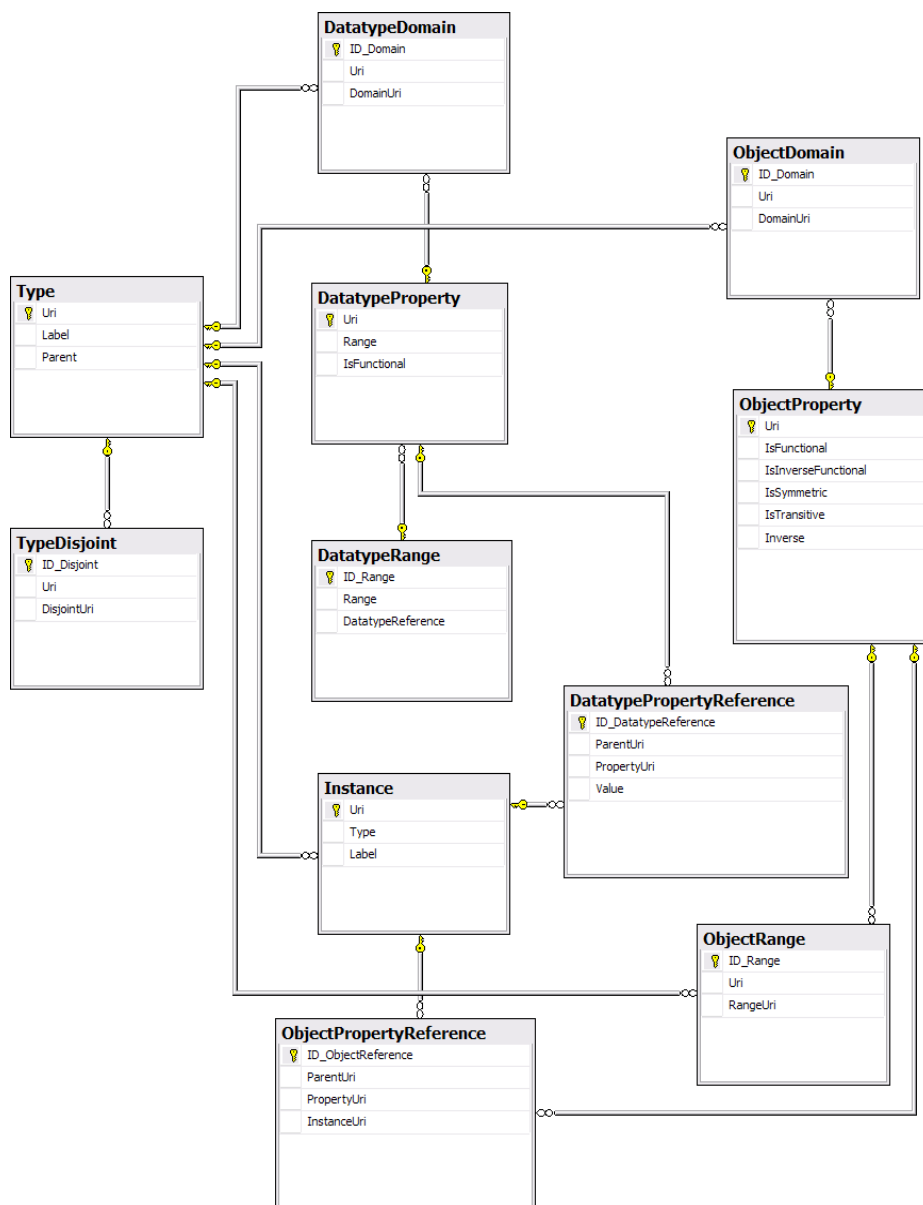


Obrázok 1. Prehľad základných tried objektového modelu ontologie

Každá ontológia v modeli môže mať určené vlastné úložisko, pričom štandardne, sú všetky dáta uložené v pamäti. Objektový model definuje pre úložisko rozhranie, preto

je možné implementovať ľubovoľné fyzické alebo virtuálne úložisko. Štandardná implementácia predstavuje navrhované fyzické úložisko v relačnej databáze.

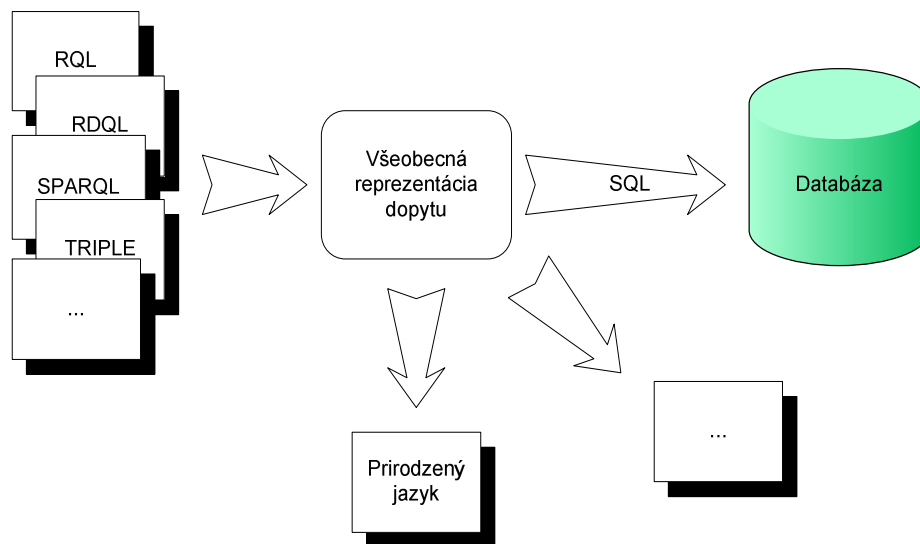
Logický dátový model úložiska (pozri obrázok 2) sme zvolili v súlade s navrhnutým prístupom k reprezentácii ontológie a rovnako ako objektový model obsahuje všetky základné entity jazyka OWL.



Obrázok 2. Návrh logického dátového modelu reprezentácie ontológie

4 Preklad

Základom prekladu je proces transformácie zo zdrojovej reprezentácie na cieľovú, znázornený na obrázku 3.



Obrázok 3. Návrh prekladu zo zdrojovej reprezentácie do cieľovej s použitím všeobecnej reprezentácie dopytov

Transformácia

Prvý krok v procese prekladu je transformácia. Úlohou transformácie je previesť zdrojovú reprezentáciu na všeobecnú reprezentáciu dopytu. Túto úlohu plnia transformátory, vytvorené pre každú zdrojovú reprezentáciu, ktorú je želané previesť na všeobecnú.

Najčastejšie predstavuje zdrojová reprezentácia nejaký dopytovací jazyk a vtedy je jadrom transformátorov gramatika zdrojového jazyka. V prípade zložitejšej transformácie je nutné použiť transformačné stratégie, ktoré nad transformovanými uzlami vykonávajú dodatočnú logiku. Výsledkom transformačnej fázy je všeobecná reprezentácia dopytu.

Všeobecná reprezentácia dopytu

Všeobecná reprezentácia dopytu slúži ako prechodový formát medzi zdrojovou a cieľovou reprezentáciou. Uzly v grafe predstavujú abstraktné entity a hrany relácie medzi nimi, sémantika sa môže líšiť v závislosti od reprezentácie, na základe analýzy sa však dá súdiť, že bude graf neorientovaný a hrany pravdepodobne vyjadrovať sekvenčnú súslednosť entít.

Preklad

Preklad z všeobecnej reprezentácie na cieľovú reprezentáciu je opakom transformačnej fázy. Všeobecná reprezentácia dopytu v tomto prípade slúži na konštrukciu konkrétnej reprezentácie, vhodnej na špecifickú interpretáciu. V rámci overenia všeobecnosti reprezentácie a funkcionality prototypu sa zhotovia prekladače na cieľovú reprezentáciu vo forme relačného dopytovacieho jazyka a experimentálnu reprezentáciu vo forme prirodzeného jazyka, určeného na interpretáciu človekom.

5 Záver a ďalšia práca

Navrhnuté riešenie predstavuje nový prístup k reprezentácii ontológie pomocou relačnej databázy, flexibilný rozširovateľný objektový model zjednodušujúci prácu s ontológiou s transparentným použitím úložiska a experimentálny návrh všeobecnej stratégie transformácie zdrojových reprezentácií ontologických dopytovacích jazykov na všeobecnú jazykovú reprezentáciu, umožňujúcu ďalší preklad na požadovanú cieľovú reprezentáciu.

Hlavnou motiváciou pre naše riešenie je nedostatočný výkon súčasných ontologických úložísk, čo sa prejavilo pri práci s veľkým množstvom dát vo viacerých doménach. Jednak v doméne pracovných ponúk vo výskumnom projekte NÁZOU (<http://nazou.fiit.stuba.sk>), ale najmä v doméne vedeckých publikácií vo výskumnom projekte MAPEKUS [9], (Modeling and Acquisition, Processing and Employing Knowledge about User Activities in the Internet Hyperspace), ktorý sa zaoberá personalizovanou navigáciou vo veľkých informačných priestoroch v doméne vedeckých publikácií. Ide o vytvorenie personalizovanej prezentačnej vrstvy nad súčasnými digitálnymi knižnicami, ktorá využíva metadáta o publikáciách integrované z viacerých zdrojov a reprezentované v ontológii (RDF/OWL).

Pre potreby experimentov sme čerpali metadáta z viacerých publikačných portálov (ACM⁴, DBLP⁵ a Springer⁶) a následne uložili v ontológii. Súčasťou schémy napĺňanej ontológie je 390 tried, 14 objektových vlastností a 22 dátových vlastností. Výsledky jednotlivých meraní sú v tabuľkách 2, 3 a 4.

Tabuľka 2. Merania pre testovaciu vzorku 10000 publikácií z digitálnej knižnice ACM

Metrika	Hodnota
Počet inštancii	238 930
Počet hodnôt dátových vlastností	96 918
Počet hodnôt objektových vlastností	682 215
Veľkosť súboru OWL	113 MB
Dĺžka importu	4.374 s
Dĺžka exportu	9.497 s

⁴ ACM – <http://portal.acm.org/>

⁵ DBLP – <http://www.informatik.uni-trier.de/~ley/db/>

⁶ Springer – <http://www.springer.com/>

Tabuľka 3. Merania pre testovaciu vzorku 50000 publikácií z DBLP

Metrika	Hodnota
Počet inštancii	127 116
Počet hodnôt dátových vlastností	203 087
Počet hodnôt objektových vlastností	632 442
Veľkosť súboru OWL	84.4 MB
Dĺžka importu	3.397 s
Dĺžka exportu	7.214 s

Tabuľka 4. Merania pre testovaciu vzorku 35000 publikácií z digitálnej knižnice Springer

Metrika	Hodnota
Počet inštancii	102 410
Počet hodnôt dátových vlastností	259 415
Počet hodnôt objektových vlastností	155 127
Veľkosť súboru OWL	75.7 MB
Dĺžka importu	2.038 s
Dĺžka exportu	5.634 s

V súčasnom stave umožňuje efektívnu prácu s dátami a je škálovateľné aj pre desiatky miliónov trojíc a viac. Prínosom riešenia je oddelenie schémy ontológie od inštančných dát, urýchlenie pomocou množstva optimalizácií ako sú vyrovnávacie pamäte, optimalizačné príznaky pre rôzne režimy práce s ontológiou, možnosť dávkových operácií s úložiskom a iné.

V budúcnosti je snahou zlepšiť návrh všeobecnej jazykovej reprezentácie a pokryť čo najviac zdrojových reprezentácií, aby bolo možné vykonávať dopyty pomocou najpoužívanejších ontologických dopytovacích jazykoch. Súčasťou ďalšej práce bude tiež rozšírenie podpory štandardu OWL, najmä OWL DL a prípadne rozšírenie o nové prvky štandardu OWL 1.1.

Podakovanie. Tento príspevok vznikol v súvislosti s riešením projektu Agentúry na podporu vedy a techniky na základe Zmluvy č. APVT-20-007104.

Referencie

1. Wood, D., Gearon, P., Adams, T. Kowari: A Platform for Semantic Web Storage and Analysis. In XTech, May, 2005
2. Mota, L., Botelho, L., Mendes, H., Lopes, A. O3F: an Object Oriented Ontology Framework. In International Conference on Autonomous Agents, 2003.
3. Magkanaraki, A. et al. Ontology storage and querying. Technical Report 308, ICS-FORTH, Heraklion, Crete, Greece, April 2002.

4. Parvatikar, P., Portwin, K. Scaling Jena in a commercial environment: The Ingenta MetaStore Project. In Proceedings of the 2006 Jena User Conference, Bristol, UK, May 2006.
5. Chebotko, A., Shiyong, L., Hasan, M. J., and Farshad, F. Semantics Preserving SPARQL-to-SQL Query Translation for Optional Graph Patterns. Technical Report TR-DB-052006-CLJF, Department of Computer Science, Wayne State University, May, 2006.
6. Cyganiak, R. A relational algebra for SPARQL. HPL-2005-170, Digital Media Systems Laboratory, HP Laboratories Bristol, September, 2005.
7. Harris, S. SPARQL query processing with conventional relational database systems. In International Workshop on Scalable Semantic Web Knowledge Base System, 2005.
8. Dokulil, J. Evaluation of SPARQL queries using relational databases. In Proc. of International Semantic Web Conference, Springer, LNCS 4273, 2006, pp. 972-973.
9. Bieliková, M., Návrat, P. Modeling and acquisition, processing and employment of knowledge on user behavior in hyperspace of the Internet. In *Proc. of Znalosti 2007*, Ostrava, pp. 368-371 (in Slovak).

Annotation:*Ontology representation and querying using relational model*

This paper discusses design of new approach to representation of ontology in physical storage based on relational database. Proposed design combines several known approaches and aims to allow scalability for huge amounts of data as well as efficient data access and manipulation. Part of proposal is a more general language representation that would enable querying on stored data with expressiveness equivalent to sum of all current ontology query languages. Purpose of this representation is to increase the degree of abstraction between source representation (existing languages) and target representation (language of physical storage) to allow for greater flexibility and bidirectional translation between individual representations.