Integrace dat na sémantickém webu

Linková, Zdeňka
2008

Integrace dat na sémantickém webu

doktorand:
ING. ZDEŇKA LINKOVÁ

Ústav informatiky AV ČR, v. v. i.
Pod Vodárenskou věží 2
182 07 Praha 8

Katedra matematiky
FJFI ČVUT
Trojanova 13
120 00 Praha 2

linkova@cs.cas.cz

školitel:
ING. JÚLIUS ŠTULLER, CSC.

Ústav informatiky AV ČR, v. v. i.
Pod Vodárenskou věží 2
182 07 Praha 8

stuller@cs.cas.cz

obor studia:
Matematické inženýrství

Práce byla podpořena projektem 1ET100300419 programu Informační společnost (Tématického programu II Národního programu výzkumu v ČR: “Inteligentní modely, algoritmy, metody a nástroje pro vytváření sémantického webu”) a výzkumným záměrem AV0Z10300504 “Informatika pro informační společnost: Modely, algoritmy, aplikace”.

Abstrakt

V tomto příspěvku je popsán přístup k virtuální integraci dat využívající současných principů, metod a nástrojů sémantického webu. Přístup pracuje s daty ve formátu RDF a předpokládá dostupnost ontologií, které je popisují. Ontologie jsou základem pro všechny kroky prezentovaného integračního procesu. Jsou využity jak k určení vztahů mezi daty a poskytovaným integrovaným pohledem, tak i k zápisu nalezených korespondencí. Ty jsou dále použity při zpracování dotazů kladených na integrovaná data.

1. Úvod

Úloha zpracování dat z různých (i distribuovaných) datových zdrojů je známa více než 40 let. Tato úloha je označována jako integrace dat a je předmětem mnoha výzkumných prací a projektů zabývajících se celou škálou typů dat - od dat relačních databází přes obecná (heterogenní) data. Současným velmi rozšířeným tématem je integrace dat pocházejících z webu, případně dat sémantického webu.

V případě webových dat je obvykle používána tzv. virtuální integrace dat [18]. Tento přístup je někdy také označován jako integrace pomocí pohledů či pomocí mediátorů. Je založený na tom, že se na data poskytne globální integrovaný pohled (který je ovšem virtuální), místo aby byla úloha řešena vytvořením nového materializovaného zdroje. Definovaný pohled zprostředkovává přístup k datům, která zůstávají fyzicky uložena v původních zdrojích, nicméně díky němu je možné původní data zpracovávat takovým způsobem, jako kdyby byla uložena na jednom místě, v jednom zdroji, v jednom prostředí, se stejným schématem atd.

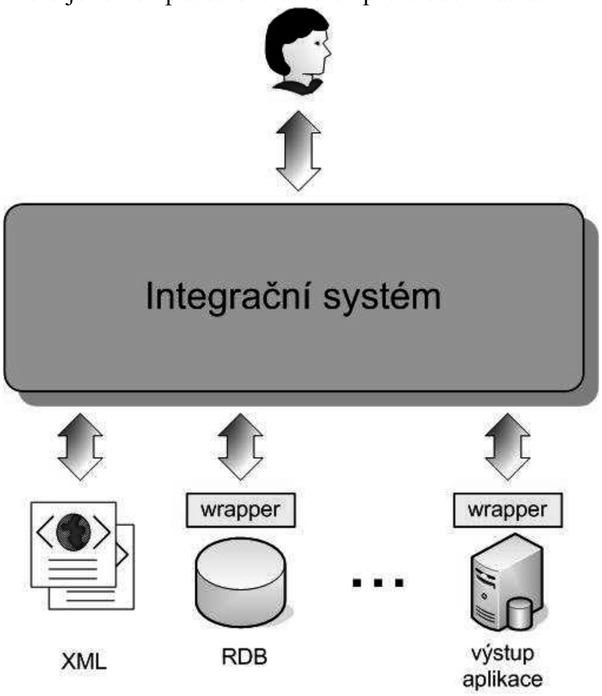
Abychom více omezili obecný typ dat, která chceme integrovat, zaměříme se na data sémantického webu. Integrace takovýchto dat může vycházet z toho, že na sémantickém webu by měla být počítačově zpracovávatelná data. Současnými prostředky a technikami, které jsou využívány k podpoře této myšlenky je jazyk XML, model RDF a OWL ontologie. Na základě hlavní motivace sémantického webu - umožnit zpracování dat bez nutnosti lidského zásahu, mohou tedy přístupy řešení integrace založené na těchto principech očekávat lepší zautomatizování řešené úlohy.

Současné projekty v této oblasti se zaměřují hlavně na využití ontologií. Ontologie mohou být použity v mnoha krocích integračního procesu. Nejčastěji jsou ovšem využity ve fázi hledání korespondencí mezi integrovanými daty. Tento článek popisuje přístup, ve kterém jsou ontologie kromě výše uvedeného použity také k definování nalezených korespondencí. Součástí popisu přístupu je nejen jak získat potřebné korespondence a jakým způsobem je v ontologii zapsat, ale také jak je poté využít při zpracování dotazů.

Článek je organizován následovně: Část 2 poskytuje základní popis obecného přístupu virtuální integrace dat, v podrobnostech se pak dále orientuje na přístup založený na ontologiích a prezentuje ideu využití ontologie jako prostředku k popisu vztahů mezi jednotlivými elementy zdrojů. Část 3 pak popsaného přístupu využívá při zpracování dotazů. Srovnání s jinými ontologicky zaměřenými přístupy je předmětem části 4. Celý článek shrnuje část 5.

2. Integrace dat s využitím ontologií

Běžným způsobem jak kombinovat data pocházející z velkého množství zdrojů nebo ze zdrojů s relativně často se měnícím obsahem je virtuální integrace dat. V takovém přístupu řešení úlohy integrace zůstávají data uložena v původních zdrojích a přístup k nim je umožněn prostřednictvím integrovaného pohledu nebo pomocí rozhraní integračního systému, který takový pohled poskytuje. Z této myšlenky vyplývá hlavní výhoda přístupu: nevytváří se kopie dat v novém materializovaném zdroji - není třeba se zabývat aktuálností dat a nemusí být řešeny paměťové nároky. Proto je tento způsob často volen pro webová data.



Obrázek 1: Virtuální integrace dat

Základem přístupu na Obr. 1. jsou datové zdroje. Vyšší vrstva je reprezentována komponentami označovanými jako *wrappery* - ty přísluší k lokálním zdrojům. Každý wrapper poskytuje přístup ke zdroji a plní funkci rozhranní mezi lokálním prostředím zdroje a prostředím integračního systému.

Vlastní jádro integrace představuje integrační systém, který použije uživatel, chce-li přistupovat k integrovaným datům. Uživatel formuluje své dotazy v prostředí globálního pohledu prezentovaného systémem. Protože však dotaz musí být vyhodnocen nad daty ve zdrojích, jejichž prostředí může být naprosto odlišné, musí systém dotaz nějakým způsobem zpracovat, než jej může vyhodnotit nad zdroji, aby mohl vrátit odpověď uživateli. K umožnění požadované funkcionality jsou definovány korespondence mezi globálním a jednotlivými lokálními prostředími.

Integrační proces je možné vidět jako kolekci úloh, které spolu zajistí žádaný výsledek. Základními kroky ve virtuálně řešené integraci jsou:

- *matching* - úloha hledání korespondencí mezi daty
- *mapování* - způsob, jak zaznamenat nalezené korespondence
- *dotazování* - úloha vyhodnocení dotazů za pomoci informací uložených v mapování

V prezentovaném přístupu jsou uvažována data pocházející ze sémantického webu. Proto jsou předpokládány zdroje obsahující RDF data vyjádřená pomocí syntaxe XML. Dalším důležitým předpokladem jsou OWL ontologie popisující integrované zdroje. Presentovaný přístup těží z dostupných informací obsažených v ontologiích, proto je jejich dostupnost klíčovým předpokladem tohoto způsobu řešení integrace dat.

2.1. Korespondence mezi daty

Při hledání vztahů mezi daty obsaženými v různých datových zdrojích lze nalézt různé typy vzájemných korespondencí. V obecném případě může jeden element jednoho zdroje korespondovat s jedním nebo více jinými elementy (i jiných zdrojů), může korespondovat s kombinací elementů, nebo nemusí korespondovat s žádným jiným elementem. V této souvislosti se obvykle při hledání korespondencí používá pojem *kardinalita*, která pro určitou korespondenci vyjadřuje, kolik elementů mapovaných schémat do vztahu vstupuje. Kardinalita korespondence může být 1:1, 1:N, N:1, N:M. Většina existujících přístupů využívá kardinalit 1:1 nebo 1:N.

Prezentovaný přístup uvažuje vztahy následujících kardinalit:

- **1:1** - při vzájemném porovnávání dvou schémat. Tento případ vyjadřuje, že element jednoho schématu je ve vztahu s jedním elementem druhého schématu.
- **1:N** - při porovnávání jednoho schématu s více dalšími schématy. Tento případ je možné vidět jako množinu korespondencí kardinalit 1:1.

Uvažovaným vztahem mezi daty jsou následující *druhy korespondencí*:

- **Is-a** hierarchický vztah (tj. jeden element je obecnější než druhý, nebo naopak) - tento druh je označen jako $\subseteq$, resp. $\supseteq$.
- **Ekvivalence** mezi elementy - tento druh je označen jako =.
- **Disjunktnost** - tj. mezi elementy není žádná souvislost.

Výsledek úlohy hledání vzájemných vztahů mezi schématy, tedy nalezené korespondence, se často označuje jako *mapování*. Obecně může mapování představovat libovolná struktura. Kromě například používání mapovacích pravidel jako tvrzení pro elementy globálních a lokálních schémat (ať už ve formě 1-1 pravidel či pohledů), které jsou orientovány na konkrétní řešenou úlohu, je možné využít složitější a dokonce standardizovanou strukturu, jenž by pokrývala všechna mapování. K popisu mapování mezi elementy schématu globálního pohledu a schémat lokálních zdrojů bude použita *ontologie OWL*.

K popisu mapování bude v závislosti na typu vztahu využit odpovídající konstrukt. Abstraktním mechanismem pro seskupování popisovaných zdrojů v OWL je třída (class). Zdrojem na webu je jakákoli identifikovatelná entita. Proto bude pojetí `owl:Class` použito pro korespondenci elementů:

- **Is-a** hierarchický vztah, tj. element1 $\subseteq$ element2, lze vyjádřit pomocí podtříd. Příslušným rysem OWL je `rdfs:subClassOf`, který umožňuje vyjádřit, že extenze jedné třídy je podmnožinou extenze jiné třídy.
- Vztah **ekvivalence**, tj. element1 = element2, lze v OWL vyjádřit s `owl:equivalentClass`. `owl:equivalentClass` umožňuje vyjádřit, že dvě třídy mají stejnou extenzi. V tomto případě může být také použit `rdfs:subClassOf` tak, že definujeme element1 jako podtřídu třídy element2 a současně element2 jako podtřídu třídy element1.
- **Disjunktnost** (neboli tvrzení, že extenze jedné třídy nemá žádné společné prvky s extenzí jiné třídy) lze vyjádřit pomocí `owl:disjointWith`.

2.2. Hledání korespondencí v případě sdílené ontologie

Důležitým předpokladem prezentovaného přístupu je dostupnost ontologií, které popisují integrovaná data. Ke každému uvažovanému zdroji je tedy předpokládána existence nějaké popisující ontologie. Situace přitom nemusí být taková, že jeden zdroj je popsán právě jednou ontologií, ale zdroj může být popsán více ontologiemi, přičemž každá z nich jej popisuje pouze částečně, nebo naopak jediná ontologie může popisovat data více zdrojů současně.

V nejjednodušším případě je popis všech zdrojů dostupný v jediné ontologii. Tato ontologie je lokálními zdroji sdílena a pokrývá popis všech lokálních dat. Vztahy mezi elementy není třeba hledat - mohou být nalezeny přímo v této ontologii.

Uvažujeme-li dříve zmíněné typy korespondencí, je možné přístup založit na is-a hierarchii definované sdílenou ontologií. Některé vztahy nemusí být v ontologii vyjádřeny přímo, ale je možné je z ontologie získat využitím tranzitivity is-a vztahu. Je-li například použit přístup k ontologii jako grafu s třídami popisujícími jednotlivé pojmy jako uzly a s orientovanými hranami vyjadřujícími existenci is-a vztahu, korespondenci nepopisuje pouze existující hrana, ale také ohodnocená cesta v grafu.

V případě, že jsou elementy disjunktní, znamená to, že v is-a hierarchii neexistuje žádná cesta a není tedy nutné nějaký vztah hledat. V praxi vede tato situace ke stejnému efektu, jako když je vztah hledán, ale žádný není nalezen. Ovšem je vhodné tuto informaci o disjunktnosti dále uchovávat, protože může být dále využita při rozšiřování přístupu například o další usuzování apod.

2.3. Obecný případ hledání korespondencí založený na ontologiích

Obecně nemusí být ontologie, která by popisovala všechna zpracovávaná data, dostupná. Některé zdroje mohou sdílet některé pojmy, avšak sdílení všech pojmů všemi zdroji nelze předpokládat. Je třeba pracovat obecně s více ontologiemi. Sloučením všech ontologií, které popisují integrované datové zdroje, získáme "novou" sdílenou ontologii, a tak je tento obecný případ převeden na předchozí.

Slučováním ontologií se zabývá řada výzkumů v oblastech ontology alignment a ontology merging [5] a je tedy možné využít některou ze známých metod. V souvislosti s ontologemi, pojmy alignment a merging spolu úzce souvisí. Pro oba jsou také relevantní úlohy hledání korespondencí (matching) a mapování (mapping). *Ontology alignment* obvykle označuje stanovení binárních vztahů mezi dvěma ontologiemi. To umožňuje definovat způsob, jak tyto ontologie sloučit. Výsledkem *ontology merging* je nová integrovaná ontologie.

Metodami pro ontology merging, jež je možné při hledání sdílené ontologie použít, se zabývá mnoho výzkumných projektů, například Chimaera [7], PROMPT [12], FCA-MERGE [16], HCONE [6]. V této fázi integračního procesu je možné využít některý z již vytvořených nástrojů. To je výhoda, která vyplývá z faktu, že k zachycení potřebných vztahů využíváme standardizovaný nástroj.

3. Dotazování nad integrovanými daty

Vytvoření mapování uvedené v předchozí kapitole je stěžejní úloha, jejíž výsledek hraje důležitou roli při přístupu k datům pomocí dotazů. Dotazy jsou tvořené nad poskytovaným pohledem (využívají jeho jazyk, schéma apod.). Pro vyhodnocení dotazu nad daty uloženými v lokálních datových zdrojích je třeba původní dotaz nějakým způsobem zpracovat.

Zpracováním dotazu [13] se zabývají dva základní přístupy. Prvním je *přepisování dotazů* (query rewriting) - dotaz je dekomponován na části odpovídající lokálním zdrojům. Ty jsou dále přepsány tak, aby byly vyjádřeny v prostředí příslušného lokálního zdroje. Nad zdrojem jsou pak vzniklé lokální dotazy vyhodnoceny a ze získaných lokálních odpovědí je následně sestavena globální odpověď, která je vrácena jako odpověď na původní (uživatelův) dotaz.

Druhou možností je *odpovídání dotazů* (query answering), která nijak nespecifikuje, jak má být daný dotaz zpracován. Jejím cílem je využít všechny dostupné informace k získání odpovědi na dotaz. Příkladem může být hledání takových dat, u nichž lze dle dostupných znalostí usuzovat, že jsou hledaným výsledkem.

V konkrétní situaci, kterou se zabývá tento článek, jsou uvažována RDF/XML data. RDF/XML data jsou obsažena v původních zdrojích a jsou také prezentována jako data integrovaného pohledu. V obou případech - na lokální i globální úrovni - je tedy jako dotazovací prostředek využíván jazyk SPARQL. Úlohou je globálně vyjádřený kladený dotaz vyjádřit v takové formě, aby bylo možné dotaz vyhodnotit nad zdroji.

K přepsání globálního dotazu do příslušných lokálních subdotazů je využito mapování zachycené v ontologii. Z této ontologie jsou patrné uvažované vztahy mezi pojmy použitými v dotaze a pojmy, které používají lokální zdroje. Přirovnáme-li ontologii ke grafu, ve kterém jsou pojmy zobrazeny jako uzly a vztahy mezi nimi jako ohodnocené hrany, lze přepsání pojmu, který byl v dotaze použit, získat z ontologie následujícím způsobem: všechny pojmy, do kterých vede z daného pojmu cesta ohodnocená uvažovanými vztahy korespondence (např. ekvivalence nebo hierarchie) jsou relevantní a použitelné při přepsání dotazu. Každého kandidáta na přepsání tedy získáme průchodem grafu ontologie od daného pojmu přes hrany korespondencí.

Není nutné využívat pouze hrany vyjadřující ekvivalenci. Například při uvažování hierarchie pojmů lze využít také is-a vztah. Jde přitom o pravidlo, jehož princip je dobře znám například v objektově orientovaném programování: potomek může zastoupit svého předka. Chceme-li uvažovat bohatší škálu korespondencí, je třeba přepisovací mechanismus doplnit o adekvátní mechanismy, aby bylo možné vztahů v přepisování využít.

Zvolený způsob zpracování dotazů v prezentovaném přístupu je popsán následujícími přepisovacími algoritmy. Základní situací je tzv. *jednoduchý dotaz*, tj. dotaz obsahující pouze jednoduchou podmínku na požadovaná data trojice RDF, RDF trojice v dotaze nijak nekombinujeme. Dotaz tedy není třeba rozkládat a získané odpovědi není třeba kombinovat. Globální odpověď získáme přepsáním lokálních odpovědí do globálního prostředí.

Algoritmus 1 *Přepsání jednoduchého dotazu I*

vstupy: globální dotaz, mapovací ontologie
výstupy: lokální dotazy, lokální odpovědi, globální odpověď

*- pro každý pojem t generuj **množinu všech možných přepsání pojmu r(t)***
*- použitím všech r(t) generuj **množinu všech možných přepsání dotazu**, tj. množinu všech lokálních dotazů*
*- všechny **lokální dotazy** vyhodnoť nad všemi lokálními zdroji a získej **lokální odpovědi***
*- využitím reversního přepsání vrať odpovědi v globálním prostředí, tj. **globální odpověď***

Základní případ nemusí nutně vést k situaci, že by odpovědí musela být jediná trojice RDF. Hledaná data mohou být obsažena ve více zdrojích. Dá-li každý takový zdroj odpověď, jsou všechny tyto získané RDF trojice součástí výsledku, který získáme jejich sjednocením. Může následovat další zpracování výsledku, například odstranění duplicit. V této fázi je též možné, že odhalíme nekonzistenci v datech zdrojů.

Uvedený algoritmus je možné (a je to dokonce žádoucí) dále zefektivňovat. Ptáme-li se všech zdrojů s využitím všech možných přepsání, je jednak u některých kombinací zdrojů a dotazů předem očekávána prázdná množina s odpovědí a jednak narůstá počet všech možných přepsání dotazu. V případě jednoduchého dotazu s podmínkou na jedinou trojici jde o zanedbatelný fakt, ovšem ve složitějších případech při kombinování trojic či kladení složitějších podmínek objem lokálních dotazů neúnosně narůstá.

V optimalizované formě postupu přepisování je proto zohledněn fakt, zda je daný pojem zdrojem podporován či nikoliv, tedy dotaz je přepisován přímo do formy pro konkrétní datový zdroj. Využity jsou tedy pouze podporované pojmy neboli relevantní k danému zdroji. Takovou informaci je možné získat přímo z ontologie zdroje, schémata zdroje, nebo také předzpracováním zdroje, pokud je požadováno tuto množinu co nejvíce omezit. To je velmi efektivní v případech, kdy je podporovaná ontologie mnohem rozsáhlejší vzhledem ke zdroji, schéma obsahuje velké množství nepovinných prvků a podobně.

Algoritmus 2 *Přepsání jednoduchého dotazu II*

vstupy: globální dotaz, mapovací ontologie, množiny podporovaných pojmů pro každý zdroj
výstupy: lokální dotazy, lokální odpovědi, globální odpověď

*- pro každý pojem t generuj **množinu všech relevantních přepsání pojmu r(t)***
*- použitím všech r(t) generuj **množinu všech relevantních přepsání dotazu**, tj. množinu všech lokálních dotazů*
*- všechny **lokální dotazy** vyhodnoť nad všemi lokálními zdroji a získej **lokální odpovědi***
*- využitím reversního přepsání vrať odpovědi v globálním prostředí, tj. **globální odpověď***

V případě, že globální dotaz obsahuje složenou podmínku, například při kombinaci více RDF trojic, je nutné složený dotaz nejprve rozdělit do více jednoduchých dotazů s jednoduchými podmínkami. Získané jednoduché odpovědi je nutné před vrácením odpovědi adekvátním způsobem opět složit. Rozklad dotazu na jednoduché dotazy je určen strukturou podmínek na data RDF. Obecně jde například o kombinaci sjednocením či průnikem, adekvátní složení je tedy průnik odpovědí, či jejich sjednocení.

Při rozkladu složeného dotazu však nejde pouze o podmínku specifikovanou v dotaze. Ovlivněn bude také požadovaný výstup - jde-li v dotaze o kombinaci trojic, je nutné, aby v jednoduché odpovědi byly obsaženy prvky, přes které je pak skládána globální složená odpověď. Před vlastním rozkladem dotazu je proto nutné tyto výstupy (pokud nejsou uvedeny) doplnit. Při rozkladu dotazu pak není rozdělena jen vlastní podmínka, ale také výstupy tak, aby každý jednoduchý dotaz obsahoval pouze vzájemně relevantní části.

Algoritmus 3 *Přepsání složeného dotazu*

vstupy: globální dotaz, mapovací ontologie, množiny podporovaných pojmů pro každý zdroj
výstupy: globální jednoduché dotazy, lokální jednoduché dotazy, lokální jednoduché odpovědi, globální odpověď

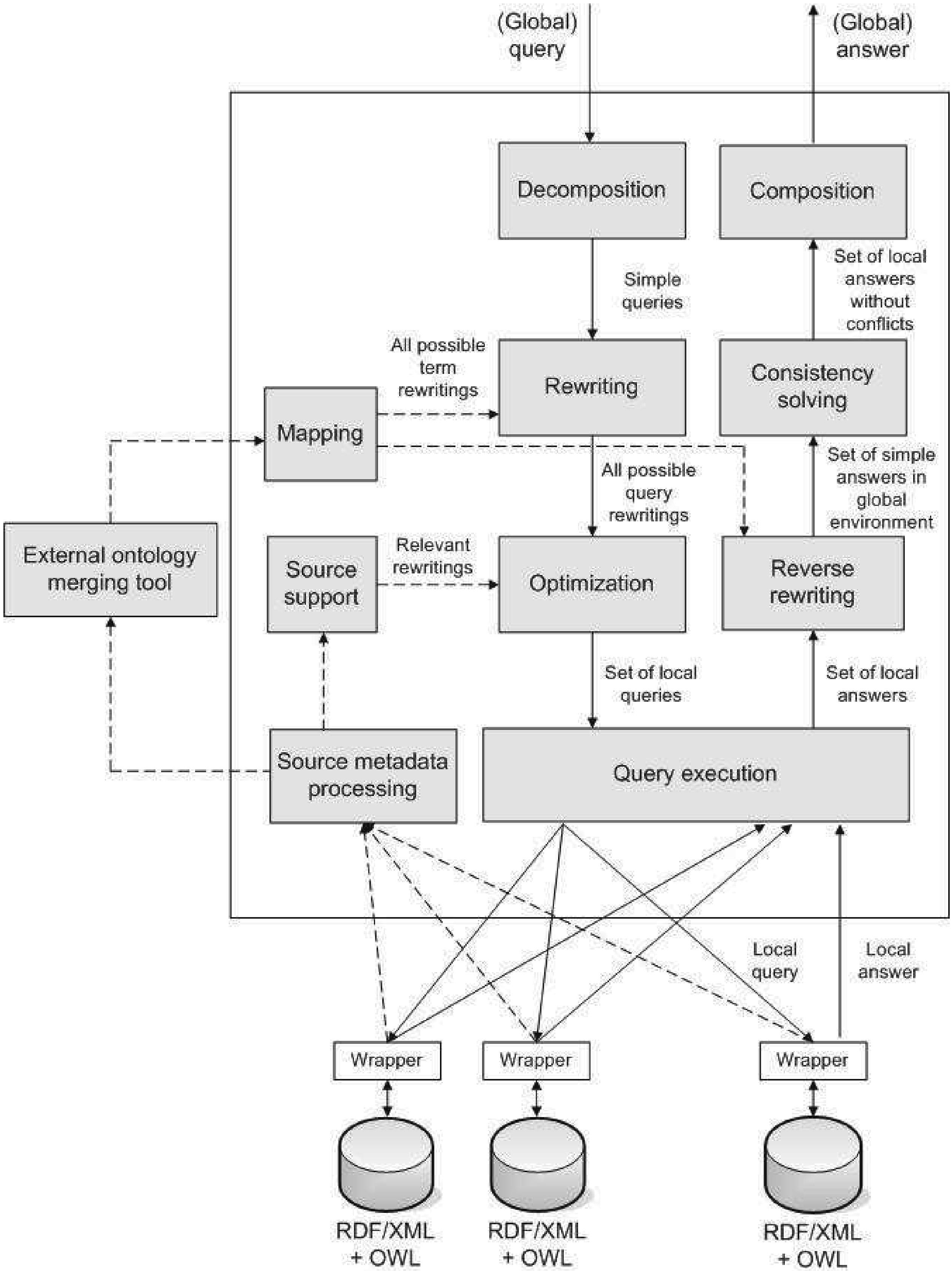
*- rozložením složených podmínek na jednoduché rozlož dotaz na **jednoduché dotazy***
*- pro každý jednoduchý dotaz přepisovacím algoritmem získej **jednoduché odpovědi***
*- jednoduchých odpovědí sestav **globální složenou odpověď***

Celý proces zpracování dotazu pomocí uvedených přepisovacích algoritmů, včetně zpracovávaných dat v jednotlivých fázích je znázorněn na Obr. 2.

4. Srovnání přístupů

Integrace dat je složitá úloha, která zahrnuje celou sadu podúloh, které je třeba řešit, abychom v konečné fázi získali požadovaný výsledek. I jednotlivé fáze procesu integrace jsou značně obsáhlé a speciálně se jimi zabývá řada výzkumných článků.

Přístupy, které se věnují hledání korespondencí [10], [14], [15], se dají klasifikovat dle úrovně informací, kterou o datech využívají. Jedná se o metody pracující na úrovni instancí (korespondence mezi schématy zdrojů), na úrovni používaných pojmů (lingvisticky založené metody, zpracování slov jako řetězců znaků) nebo na úrovni struktury (grafové metody). Velmi častá je ovšem kombinace těchto přístupů a uplatňují se i funkce, vyjadřující podobnosti srovnávaných dat [11], [17], [19].



Obrázek 2: Zpracování dotazu

V tomto pohledu by se mohlo zdát, že přístup popsaný v tomto článku je značně odlišný. Podobné metody jako při hledání korespondencí se však uplatňují při slučování ontologií, jichž prezentovaný přístup využívá. Podobnosti lze tedy nalézt, jsou pouze řešeny na jiné úrovni. Toto převedení úlohy integrace dat na úlohu slučování ontologií [9] mimo jiné umožní využít výsledků jiných projektů (např. vytvořených nástrojů) a více zautomatizovat operace probíhající v procesu.

Na rozdíl od úlohy hledání korespondencí řešenou "tradičním" způsobem, kde je často nutná lidská interakce v konečné fázi při určení skutečně korespondujících dat, jsou všechny korespondence získané z ontologie s určitostí přijaty. Není na ně nahlíženo nejprve jako na kandidáty, neboť zde není žádný odhad korespondencí - všechny z nich jsou v dané ontologii definovány. Je však nutné poznamenat, že i v tomto případě je možné, že je určení korespondencí řešeno lidským zásahem, a to v případě využití externího nástroje při sloučení ontologií. Ačkoliv při odvozování vztahů schémat ze sdílené ontologie žádní kandidáti nevznikají a korespondence jsou přímo určeny, v obecném případě mohou vznikat právě při řešení podúlohy hledání sdílené ontologie pomocí existující metody, která s kandidáty pracuje.

K vyjádření mapování lze použít od jednoduchých 1-1 mapovacích pravidel vyjadřujících přímou korespondenci mezi elementy, přes mapování konceptu na dotaz nebo pohled [2], až po pomocné mapovací struktury. Různé projekty obvykle používají vlastní pojetí mapování, často je následován přístup definice mapování LAV (Local As View), GAV (Global As View), či jejich kombinace GLAV [8].

Zpracování dotazů je pak přímo ovlivněno volbou mapování. Podle složitosti jak uvažovaných dotazů, tak i mapování se odvíjí velmi individuálně konkrétní podoba přístupu k dotazům, například Inverse rule algorithm [3], Bucket algorithm a jeho vylepšení v systému MiniCon [13], či Styx [1].

Podobnost prezentovaného přístupu lze nalézt v případě algoritmu Styx, který také využívá vztahů předek - potomek při zpracování dotazů. Inspirován algoritmem Styx byl algoritmus použitý v systému VirGIS [4] integrujícím geografická data. V něm je udržováno mapování separátně pro každý zdroj a tak je dosaženo dotazování na relevantní pojmy. Na rozdíl od toho přístup prezentovaný v tomto článku pracuje s mapováním jako celkem a separátně udržuje pouze informace o podpoře částí pro každý zdroj. To, že celé mapování je obsaženo v jediné struktuře, umožňuje efektivní obohacování mapování při zjištění dalších korespondencí, při přidání nového zdroje do systému či při reakci na změnu některého ze zdrojů. Vše bez nutnosti přepracovat již zjištěné mapování nebo dokonce mapovat každý zdroj znovu.

5. Závěr

Článek popisuje přístup k řešení úlohy virtuální integrace dat pomocí ontologií. Ontologie je využita nejen ke získání informací při hledání souvislostí mezi daty, ale slouží i jako prostředek k zachycení nalezených korespondencí.

Užití ontologie pro mapování umožňuje řešit změny a obohacování systému doplněním ontologie mapování bez nutnosti zasahovat do již existujících částí. Přináší také možnost znovupoužití i v jiných úlohách či situacích. Navíc, bude-li v budoucnu třeba zachytit i další typy vztahů mezi elementy, může být ontologie dále využita, neboť je schopna zachytit různé typy vztahů.

Mapování popsané v ontologii slouží dále jako klíčový zdroj ve fázi zpracování dotazů. Pro zodpovězení dotazů kladených na integrovaná data je v článku prezentován mechanismus, s nímž je daný dotaz z globální úrovně rozložen a přepsán tak, aby mohl být vyhodnocen nad fyzickými daty. Využitím představeného přístupu integrace je tak možné pracovat s daty na globální úrovni bez toho, aby uživatel musel řešit, ve kterém zdroji a v jaké podobě se dotazovaná data nachází.

Literatura

[1] Bernd A., Beeri C., Fundulaki I. a Scholl M., "Querying XML Sources Using an Ontology-Based Mediator", *On the Move to Meaningful Internet Systems, Confederated International Conferences DOA, CoopIS and ODBASE*, Springer-Verlag, pp. 429–448, 2002.

[2] Calvanese D., De Giacomo G. a Lenzerini M., "Ontology of integration and integration of ontologies", *Proceedings of DL 2001 - Description Logic Workshop*, 2001.

[3] Duschka O. M. a Genesereth M. R., "Answering recursive queries using views", *Proceedings of ACM PODS, ACM Press*, pp. 109–116, 1997.

[4] Essid M., Boucelma O., Lassoued Y. a Colonna, F.-M., "Query Processing in a Geographic Mediation System", *Proceedings of The 12th International Symposium of ACM GIS* Washingtin D.C., 2004.

[5] Kalfoglou Y. a Schorlemmer M., "Ontology

mapping: the state of the art", *The Knowledge Engineering Review 18*, 1, pp. 1–31, 2003.

[6] Kotis K. a Vouros G. A., "The HCONE Approach to Ontology Merging", *ESWS*, LNCS 3053, Springer, pp. 137–151, 2004.

[7] McGuinness D. L., Fikes R., Rice J. a Wilder S., "An Environment for Merging and Testing Large Ontologies", *Proceedings of the Seventh International Conference*, 2000.

[8] Lenzerini M., "Data Integration: A Theoretical Perspective", *Proceedings of the 21st ACM SIGMOD - SIGACT - SIGART symposium on Principles of database systems*, ACM Press, pp. 233–246, 2002.

[9] Linková Z., "Ontology-Based Schema Integration", *SOFSEM 2007. Theory and Practice of Computer Science*, Vol.: 2, Institute of Computer Science AS CR, Prague, pp. 71–80,2007.

[10] Mitra P., Wiederhold G. a Jannink J., "Semi-automatic integration of knowledge sources", *Proceeding of the 2nd Int. Conf. On Information FUSION'99*, 1999.

[11] Nottelmann H. a Straccia U., "Information retrieval and machine learning for probabilistic schema matching", *Inf. Process. Manage. 43*, 3, pp. 552–576, 2007.

[12] Noy F. N. a Musen M. A., "PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment", *AAAI/IAAI*, pp. 450–455, 2000.

[13] Pottinger R. a Levy A., "A Scalable Algorithm for Answering Queries Using Views", *Proceedings of the 26th VLDB Conference*, Cairo, Egypt, 2000.

[14] Rahm E. a Bernstein P. A., "A survey of approaches to automatic schema matching", *VLDB Journal: Very Large Data Bases 10*, 4, pp. 334–350, 2001.

[15] Shvaiko P. a Euzenat J., "A survey of schema-based matching approaches", *3730*, pp. 146–171, 2005.

[16] Stumme G. a Maedche A., "FCA-MERGE: Bottom-Up Merging of Ontologies", *IJCAI*, pp. 225-234, 2001.

[17] Su X. a Gulla J. A., "An information retrieval approach to ontology mapping", *Data & Knowledge Engineering 58*, 1, pp. 47–69, 2006.

[18] Ullman J. D., "Information integration using logical views", *Theoretical Computer Science 239*, pp. 189–210, 2000.

[19] Yi S., Huang B. a Chan W. T., "Xml application schema matching using similarity measure and relaxation labeling", *Inf. Sci. 169*, 1-2, pp. 27–46, 2005.