Integrace dat a sémantický web

doktorand:
ING. ZDEŇKA LINKOVÁ
Katedra matematiky
Fakulta jaderného a fyzikálního inženýrství
ČVUT
Praha
zdena@centrum.cz

školitel:
ING. JÚLIUS ŠTULLER, CSC.
Ústav informatiky AV ČR
Pod Vodárenskou věží 2
182 07 Praha 8
stuller@cs.cas.cz

obor studia:
Matematické inženýrství
číselné označení: 39-10-9

Abstrakt

World Wide Web obsahuje data, která jsou pro počítačové programy nesrozumitelná. Následkem toho je na něm obtížné některé věci zautomatizovat. Nedostatky současného webu by měl odstranit sémantický web, ve kterém data budou mít přesně popsaný význam. Zlepšení může přinést také v oblasti integrace, která je v případě dat pocházejících z webu velmi obtížná. Tento článek[1] se zabývá integrací webových dat. Zaměřuje se na relační data ve formátu XML a navrhuje postupy základních integračních operací.

1. Úvod

World Wide Web (WWW) obsahuje obrovské množství informací vytvořených mnoha různými organizacemi, společnostmi i jednotlivci z mnoha různých důvodů. Současný web je však více přátelský k lidskému uživateli než k počítačovým programům. Vše na něm je pro počítačové aplikace sice čitelné, avšak nesrozumitelné. Následkem velkého množství různorodých dat je obtížné udržování, aktualizace a vyhledávání informací. To ztěžuje snahu věci na webu zautomatizovat. Přitom je hodně způsobů, jak by programy mohly obsahu webu využívat, jen kdyby mu rozuměly.

Web může plně dosáhnout svých možností pouze tehdy, jestliže se stane místem, kde mohou být data sdílena a zpracovávána automatizovanými nástroji stejně jako lidmi. Toho chce dosáhnout sémantický web. Je založen na myšlence mít data na webu nejen uložena, ale také definována a spojena takovým způsobem, aby mohla být programy užívána nejen k zobrazení, ale též pro navigaci, integraci, automatizaci a opakované použití napříč různými aplikacemi.

2. Sémantický web

Sémantický web [1], [2] je zamýšlen jako rozšíření současného World Wide Webu sestávající z počítačově čitelných, srozumitelných a smysluplně zpracovávatelných dat. Základem je zavedení sémantiky – spolu s uloženými daty bude k dispozici i jejich popis. To znamená, že data budou mít definovány svůj význam.

Ačkoli v současnosti již některé črty existují, je sémantický web zatím pouze vize. Je to cíl, ke kterému se směřuje. Založen by měl být na standardech, které jsou průběžně definovány. O definici těchto standardů usiluje W3C (WWW Consorcium [3]). Ovšem stejně jako jiné oblasti výpočetní techniky se i tato neustále vyvíjí a dle potřeb mohou vznikat standardy nové. Toto pojetí rozšíření WWW je tedy ve fázi neustálého vývoje.

[1] Práce byla částečně podpořena projektem 1ET100300419 programu Informační společnost (Tématického programu II Národního programu výzkumu v ČR): Inteligentní modely, algoritmy, metody a nástroje pro vytváření sémantického webu.

Principy sémantického webu jsou implementovány v jednotlivých vrstvách webových technologií a standardů [4]. Vrstvy jsou zobrazeny na následujícím obrázku.

Obrázek 1: Vrstvy sémantického webu

Vrstva infrastruktury poskytuje možnost identifikace, lokalizace a transformace zdrojů. *Vrstva strukturizace*, *vrstva metadat* a *vrstva logiky a ontologie* jsou nezbytné k vyjádření obsahu webu, který ze zdrojů získáváme. *Vrstva věrohodnosti* je již záležitostí konkrétních aplikací. Týká se ověřování a důvěryhodnosti získané informace - ne vše umístěné na webu totiž musí být pravdivé. Aplikace musí rozhodnout, zda se na informaci spolehne, na základě nějakého podaného důkazu věrohodnosti zdroje.

2.1. Infrastruktura

Sémantický web bude sestávat z propojených zdrojů – bude obsahovat zdroje a odkazy mezi nimi. Objekt bude možné identifikovat (stejně jako na současném webu) užíváním identifikátorů: přímý odkaz vznikne vytvořením a přiřazení URI (Universal Resource Identifier) danému objektu. Sémantický web bude také samozřejmě decentralizovaný, jak jen to bude možné. Decentralizace však vyžaduje kompromisy, třeba v nutnosti tolerovat neúplnou či chybějící informaci v podobě odkazu na neexistující zdroj.

Sémantický web bude obsahovat nejen klasické (mediální) zdroje (stránky, texty, obrázky, audio klipy), ale mnohem více – bude obsahovat zdroje představující lidi, místa, organizace a události. Navíc přinese také možnost specifikace typů zdrojů i typů odkazů. Bude obsahovat mnoho různých druhů vztahů mezi různými typy zdrojů. Díky tomu budou moci aplikace zjistit druh vztahu mezi daty.

2.2. Vyjádření datového obsahu

Důležitým požadavkem počítačově zpracovatelné informace je *strukturov ání* dat. Na webu je hlavní strukturovací technikou značkování dokumentů pomocí tzv. tagů, což je určitá část textu, která obsahuje informace udávající role a vlastnosti obsahu dokumentu. V současné době je standardním mechanizmem ke strukturování dat jazyk XML (eXtensible Markup Language) [5]. Tento jazyk poskytuje datový formát pro strukturované dokumenty a umožňuje běžnou syntaxi pro počítačově čitelná data.

Samo XML ovšem k popisu dat nestačí. Pomocí tagů lze vytvářet strukturu, ale jejich použití neříká nic o tom, co daná struktura znamená. Technologií k určení druhu a významu informace je základ pro zpracování *metadat* – RDF (Resource Description Framework) [6], který je mechanismem, jak říci něco o datech. Představuje jednoduchý mechanizmus reprezentace znalostí pro webové zdroje. Datový model RDF poskytuje abstraktní, konceptuální rámec pro definici a použití metadat. Pro účely vytváření a výměny těchto metadat je však třeba konkrétní syntaxe. RDF k tomuto účelu používá kódování pomocí XML [7].

Prostředkem k definici termínů použitých k vyjádření metadat jsou ontologie, které tak zajišťují prostředek pro sdílení termínů při spolupráci aplikací. Myšlenka sémantického webu též zahrnuje přidání logiky na web - ve smyslu používání pravidel k tvoření závěrů a podobně. Ontologie [8] označuje ucelenou kolekci termínů, vztahů a vyvozovacích pravidel. V kontextu webových technologií je ontologií dokument nebo soubor, který formálně definuje vztahy mezi termíny. Slovník ontologie lze chápat jako jakýsi výkladový slovník pojmů. Pomocí odvozovacích pravidel můžeme nad pojmy činit různé závěry a slovník se tak může dále vyvíjet.

2.3. Provozování aplikací

Skutečná síla sémantického webu se projeví, jestliže lidé vytvoří mnoho programů, které budou shromažďovat webový obsah z různých zdrojů, zpracovávat informace a vyměňovat si výsledky s ostatními

programy. Efektivita takovýchto tzv. *softwarových agentů* [9] se zvýší tím více, čím více bude obsah webu srozumitelnější pro počítače a čím přístupnější budou automatizované služby (zahrnující ostatní agenty).

Sémantický web bude moci poskytovat základ a strukturu k realizovatelnosti dalších technologií – kromě něj budou někteří agenti využívat umělé inteligence, například při automatickém vytváření složitých kolekcí hodnot, kdy se na výsledku podílí celý soubor specializovaných agentů.

3. Integrace dat pocházejících z webu

I když lze z webu získat mnoho informací, nejsou všechny informace poskytovány jediným zdrojem – jsou roztroušeny. K uspokojení konkrétního požadavku je často třeba pracovat s daty z více zdrojů, které jednotlivé dílčí části nabízejí. Výsledkem je pak ovšem více oddělených částí a nikoli požadovaná kompletní informace. Data je proto potřeba integrovat, tzn. z několika původních zdrojových informací vytvořit jediný informační zdroj, ať už materializovaný, či virtuální.

Integrace informací ze současného WWW je však velice obtížná. Na webu vznikají jakési ostrovy souvisejících dat, každý pochází od jiného poskytovatele. Poskytovatelé publikují data nezávisle, což vede k odlišnému užívání termínů a k používání odlišných nebo dokonce žádných schémat. Jednou z motivací tvorby sémantického webu je i usnadnění takové operace, jako je integrace.

3.1. Relační databáze a XML

V rámci diplomové práce[10] jsem pracovala na vývoji systému integrujícího zadané zdroje z webu. Integrační postup je založen na vstupním formátu XML, který se na webu stává de facto standardem. Formát zpracovávaných dat byl dále omezen na strukturu, kterou lze reprezentovat jako tabulku v relační databázi (viz následující obrázek). Jedním z důvodů je fakt, že integrace je dlouho uznávaným problémem databázových systémů a existují nástroje pro integraci databázových tabulek, je tedy možné oba přístupy porovnávat.

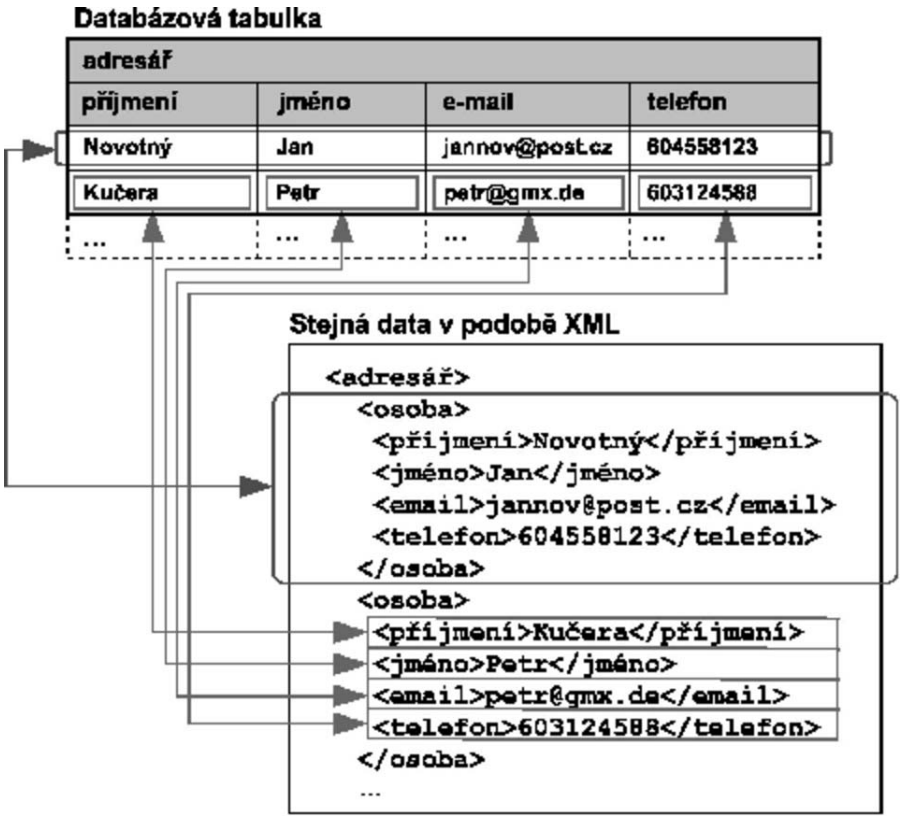
3.2. Model integrace

Při návrhu integračního systému byl použit přístup pomocí modelu stromu. Hierarchii struktury vnořených elementů v XML dokumentu lze skutečně nahlížet jako stromovou strukturu. Ze stromové struktury vychází také DOM – standardní API pro přístup k obsahu dokumentu XML.

DOM (Document Object Model) [11] je objektový model dokumentu. Dokument je prezentován jako stromová hierarchická struktura. Každému elementu (textový element, komentář, instrukce pro zpracování atd.) odpovídá jeden uzel stromu. DOM tvoří mnoho rozhraní obsahujících funkce, které umožňují celý strom dokumentu procházet a modifikovat jednotlivé uzly.

V návrhu je také použito označení inspirované názvoslovím modelu stromu:

- Zdroje jsou očíslovány a označeny: *zdroj1* a *zdroj2*.
- Uzel, který analogicky odpovídá řádku tabulky, je označen *řádek*.
  Uzel, který analogicky odpovídá sloupci tabulky, je označen *sloupec*.
  Jméno uzlu označuje *jméno*.
- Počet uzlů *řádek* ve *zdroj1* označuje *počet_řádků1* a *počet_řádků2* označuje počet uzlů *řádek* ve *zdroj2*.
  Počet uzlů *sloupec* ve *zdroj1* označuje *počet_sloupců1* a *počet_sloupců2* označuje počet uzlů *sloupec ve zdroj2*.
  Jednotlivé uzly popořadě jsou očíslovány a je použito označení *řádek(1), řádek(2), ...*, resp. *sloupec(1), sloupec(2), ...*.
- Uzel, který je následovníkem uzlu *sloupec* (je tedy typu text a obsahuje hodnotu), označuje *text*.



Obrázek 2: Formát XML a databázová data

- V přístupu je použita tečková notace.
- Je-li dán jakýkoli konkrétní uzel, je tím míněn uzel včetně svých potomků, tedy celý podstrom.

3.3. Postup integrace

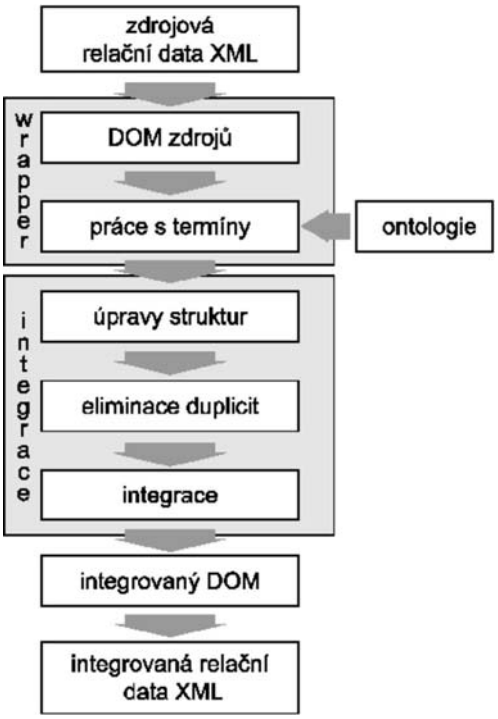
Schéma architektury navrhovaného systému a postupu úpravy a integrace datových zdrojů je patrné z obrázku na následující straně. Systém by měl mít dvě hlavní navazující části vykonávající jednak přípravu dat (moduly wrapperů [12] pro každý zdroj), jednak vlastní integraci. Příprava dat zahrnuje převod zdrojů na DOM reprezentaci a práci s použitými termíny.

Mnoho aplikací vychází z předpokladu, že rozdílná jména označují rozdílné věci. Na webu ovšem takový předpoklad učinit nelze. Je totiž například možné na jeden datový zdroj odkazovat několika různými způsoby. V některých případech je možné mezi rozdílnými pojmy použitými v datových zdrojích najít určitý vztah, dokonce i ekvivalenci. Používání ekvivalentních pojmů jako pojmů, které spolu nesouvisí, přitom vede v operaci integrace ke špatným výsledkům. Je proto výhodné před vlastní integrací s termíny pracovat.

Celý postup vlastní integrace vychází z integrace tzv. základní situace (viz dále). Její použití je z důvodů požadavků na vstupní data dosti omezeno. Proto jsou nejprve se zdroji provedeny některé úpravy: změna pořadí, odstranění či doplnění jednotlivých částí struktur. Aby výsledek integrace neobsahoval duplicitní informace, je nutné výskyt duplicit ošetřit.

Jednotlivé kroky následujícího ”základního” algoritmu integrace budou podrobněji rozvedeny v dalších odstavcích 3.4 - 3.7.

Algoritmus integrace:



Obrázek 3: Postup při integraci dat

```
Uprav {Aplikace zvoleného postupu úpravy struktur}
Seřad' {Změna pořadí ve strukturách}
Odstraň_duplicity {Eliminace duplicit}
Integruj {Integrace základní situace}
```

3.4. Integrace základní situace

Situace, kdy jsou schémata integrovaných XML dokumentů naprosto stejná, je považována za základní. Při analogii k databázové tabulce budou mají zdroje identické databázové schéma, tj. stejné sloupce – stejný počet sloupců, jejich názvy a pořadí. Integrační operací je sjednocení obou zdrojů. Celý obsah elementu dokumentu z prvního i z druhého zdroje je sloučen do jediného elementu dokumentu výsledku. Toto sjednocení lze přirovnat k operaci sjednocení dvou databázových tabulek.

Algoritmus integrace základní situace:

```
Vytvoř kořen_výsledku
for i=1,2,...,počet_řádků1
  Zkopíruj zdroj1.řádek(i) do výsledku a Napoj na kořen
for j=1,2,...,počet_řádků2
  Zkopíruj zdroj2.řádek(j) do výsledku a Napoj na kořen
```

3.5. Eliminace duplicit

Jestliže by bylo možné nějakou informaci získat z obou zdrojů, ve výsledku by se objevila duplicitně. To by nebyl dobrý výsledek integrace, a proto je nutné potencionální výskyt duplicit řešit. V návrhu je výskyt násobných informací eliminován ještě před integrační operací.

Nejprve je nutné zjistit, které informace vedou na nežádoucí redundanci. Jeden ze zdrojů je označen jako referenční. Pro každou informaci z druhého zdroje je pak nutné ověřit, jestli se nedá získat již ze

zdroje prvního – snahou vyhledat ji v referenčním zdroji. Skončí-li hledání úspěšně, násobná data budou odstraněna, a to z druhého (nereferenčního) zdroje.

V následujícím algoritmu je jako referenční zdroj *zdroj1*, duplicity jsou tedy odstraňovány ze *zdroj2*. Použitá metoda `Equals` je dvouhodnotová a porovnává dva stromy, zda jsou shodné.

Algoritmus odstraňování duplicit:

```
for i=1,2,...,počet_řádků1
    for j=1,2,...,počet_řádků2
        if zdroj1.řádek(i) Equals zdroj2.řádek(j)
             then Odstraň zdroj2.řádek(j)
```

3.6. Změna pořadí ve strukturách

Základní situaci integrace je možné aplikovat i na zdroje, jejichž struktury jsou shodné až na pořadí uzlů analogickým ke sloupcům tabulky. Příslušné pozice uzlů je pouze nutné změnit a vhodně seřadit. První zdroj je označen za referenční a dále se vychází z jeho struktury. Ve druhém zdroji je (podle referenčního) pořadí uzlů upraveno.

Algoritmus řazení sloupců:

```
for i=1,2,...,počet_sloupců1
  if zdroj1.řádek(1).sloupec(i).jméno
       ≠ zdroj2.řádek(1).sloupec(i).jméno
    then begin
      j:= Najdi_pozici zdroj1.řádek(1).sloupec(i) ve
                                              zdroj2.řádek(1)
      for k=1,2,...,počet_sloupců
           Odpoj zdroj2.řádek(k).sloupec(j)
           a Napoj ho na pozici i
    end
```

3.7. Úpravy struktur

Integraci základní situace není možné ihned použít v případě rozdílných struktur zdrojových dat. Jednoduchým případem neodpovídajících si schémat relačních XML zdrojů je, liší-li se struktury pouze v pořadí uzlů. Nabízí se pořadí uzlů vhodně upravit, to bylo ošetřeno v předchozí části. Spočívá-li ovšem rozdíl struktur zdrojů v odlišných sadách uzlů odpovídajících sloupcům tabulky, je třeba větších úprav. Volba vhodné operace závisí na okolnostech, na datech, která chceme integrovat, a na jejich významu.

Nejjednodušší možností je úprava struktur tak, že do výsledku budou zahrnuty pouze ty uzly sloupců, které se vyskytují ve všech zdrojích zároveň. Integrace proběhne přes průnik uzlů. Jestliže bude každý ze zdrojů nejprve upraven tak, že v něm budou nadbytečné sloupce odstraněny, všechny zdroje získají stejnou strukturu, případně bude třeba upravit pořadí. Pak bude moci být aplikován základní algoritmus integrace.

Algoritmus úpravy struktur na průnik:

```
A:= průnik sloupců
for i=1,2,...,počet_řádků1
     for j=1,2,...,počet_sloupců1
     if zdroj1.řádek(i).sloupec(j) not in A
          then Odstraň zdroj1.řádek(i).sloupec(j)
for i=1,2,...,počet_řádků2
     for j=1,2,...,počet_sloupců2
     if zdroj2.řádek(i).sloupec(j) not in A
          then Odstraň zdroj2.řádek(i).sloupec(j)
```

Obohacení struktury dat bez další kombinace datového obsahu je další možností, jak zpracovat zdroje rozdílných struktur. Výsledek bude mít strukturu sestávající ze všech uzlů sloupců jež do integračního procesu vstoupily, bez ohledu na původní zdroj. Každý zdroj bude obohacen o příslušný nový sloupec, jehož textová hodnota ale nebude z ostatních dat nijak odvozena. Nové textové hodnoty zůstanou buď to prázdné, nebo bude vložena speciální hodnota značící nezadání (např. hodnota NULL).

Algoritmus obohacení struktury:

```
B:=sjednocení sloupců
∀ prvek z B
     for i=1,2,..., počet_řádků1
          if prvek not in zdroj1.řádek(i)
               Vytvoř zdroj1.řádek(i).prvek
               zdroj1.řádek(i).prvek.text:='NULL'
     for i=1,2,...,počet_řádků2
          if prvek not in zdroj2.řádek(i)
               Vytvoř zdroj2.řádek(i).prvek
               zdroj2.řádek(i).prvek.text:='NULL'
```

Při obohacení struktury s kombinací dat je obohacena struktura zdrojů o nové sloupce stejným způsobem, jako v předchozím případě. S datovým obsahem se ovšem dále pracuje a je snaha data vhodně zkombinovat. K určení, jak data vzájemně souvisejí je možné využít průnik struktur zdrojů. Související data mají v takto určených uzlech stejné hodnoty. Analogii operace v takové situaci, kdy je nakombinována jak struktura, tak samotná data, lze ve světě databází spatřovat v operaci JOIN, tj. ve spojení dvou tabulek.

K vytvoření kombinace jsou využity kopie příslušných podstromů, vlastní původní podstrom zůstane beze změny – tak ho lze využít k dalším případným kombinacím. Nakonec jsou všechny původní podstromy, které vedly ke vzniku kombinace odstraněny, neboť jsou nová data obsažena v kopiích. Jak je ovšem naloženo s podstromy, které nebylo možné s ničím skombinovat, záleží na tom, co by měl výsledek obsahovat. Do výsledku je možné zahrnout pouze data, která vznikla kombinací obsahů zdrojů. Takovýto postup vede na výsledek integrace plně odpovídající databázové operaci spojení tabulek (inner join). Při takové situaci jsou ovšem ztracena data, která v druhém zdroji neměla odpovídající doplnění. Je-li požadováno všechna data zachovat, lze obohatit data, která obohatit lze, a ve zbylých případech doplnit strukturu o sloupce s nezadanými hodnotami. Tato operace a následná integrace je analogická databázovému vnějšímu spojení (outer join).

Algoritmus obohacení struktury a kombinace dat:

```
A := průnik sloupců
B := sjednocení sloupců
used1(1,2,...,počet_řádků1):=false
used2(1,2,...,počet_řádků2):=false
for i=1,2,...,počet_řádků1
  for j=1,2,...,počet_řádků2
    if ∀ prvek z A
         zdroj1.řádek(i).prvek.text =
                        zdroj2.řádek(j).prvek.text
              then souvisí:=true
              else souvisí:=false
    if souvisí = true then
         new:=Zkopíruj zdroj1.řádek(i) do zdroj1
         ∀ prvek z B
              if prvek not in new then
                        Zkopíruj zdroj2.řádek(j).prvek do new
```

```
        new:=Zkopíruj zdroj2.řádek(j) do zdroj2
        ∀ prvek z B
             if prvek not in new then
                  Zkopíruj zdroj1.řádek(i).prvek do new
for i=1,2,...,počet_řádků1
     if used1(i) then Odstraň zdroj1.řádek(i)
     else
             if inner_join
                  Odstraň zdroj1.řádek(i)
             if outer_join
                  ∀ prvek z B
                       if prvek not in zdroj1.řádek(i)
                         Vytvoř zdroj1.řádek(i).prvek
                         zdroj1.řádek(i).prvek.text:='NULL'
for i=1,2,...,počet_řádků2
     if used2(i) then Odstraň zdroj2.řádek(i)
     else
             if inner_join
                  Odstraň zdroj2.řádek(i)
             if outer_join
                  ∀ prvek z B
                       if prvek not in zdroj2.řádek(i)
                         Vytvoř zdroj2.řádek(i).prvek
                         zdroj2.řádek(i).prvek.text:='NULL'
```

4. Závěr

Integrace obecných dat pocházejících z webu je obtížná. Předpoklad XML formátu a relační struktury zdrojových dat však umožnil provést několik druhů integračních operací, z nich některé lze srovnávat s obdobnými operacemi prováděnými v oblasti relačních databází. Nicméně problematika je značně rozsáhlá - prezentovaný návrh pokrývá pouze část celého problému. Rozšiřování zpracovaného tématu je možné například v dalším využití existujících technik sémantického webu. Proto bych v tomto směru ve své práci ráda pokračovala.

Literatura

[1] T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web", *Scientific American*, vol. 284, 5, pp. 35–43, 2001.

[2] J. Euzenat, "Research Challenges and Perspectives of the Semantic Web", *Report of the EU-NSF Strategic Research Workshop*, Sophia-Antipolis, Francie, říjen, 2001.

[3] W3C (WWW Consorcium). `http://www.w3.org`.

[4] W3C: Semantic Web. `http://www.w3.org/2001/sw/`.

[5] N. Bradley, "XML kompletní průvodce", *Grada Publishing*, Praha, 2000, ISBN 80-7169-949-7.

[6] O. Lassila, R.R. Swick, "Resource Description Framework (RDF) Model and Syntax Specification", *W3C Recommendation*, únor, 1999, `http://www.w3.org/TR/1999/REC-rdf-syntax-19990222`.

[7] D. Beckett, "RDF/XML Syntax Specification (Revised)", *W3C Working Draft*, leden 2003, `http://www.w3.org/TR/2003/WD-rdf-syntax-grammar-20030123`.

[8] Ch. Welty, N. Guarino, "Supporting Ontological Analysis of Taxonomic Relationships", *Data & Knowledge Engineering*, vol. 39, pp. 51–74, 2001.

[9] J.H. Park, S.Ch. Park, “Agent-Based Merchandise Management in Business-to-Business Electronic Commerce”, *Decision Support Systems*, vol. 35, pp. 311–333, 2003.

[10] Z. Linková, “Integrace dat a sémantický web”, *Diplomová práce, FJFI ČVUT*, Praha, 2004.

[11] Document Object Model (DOM). `http://www.w3.org/DOM`.

[12] J.D. Ullman, “Information integration using logical views”, *Theoretical Computer Science* vol. 239, pp. 189–210, 2000.