AmphoraWS – webová služba pro vyhledávání ve strukturovaných dokumentech*

Marek ANDRT[1], Michal KRÁTKÝ[1], Vojtěch SVÁTEK[2], Václav SNÁŠEL[1]

[1]*Katedra informatiky, VŠB – Technická universita Ostrava*
17. listopadu 15, 708 33 Ostrava-Poruba
{marek.andrt,michal.kratky,vaclav.snasel}@vsb.cz
[2]*Katedra informačního a znalostního inženýrství , VŠE Praha*
nám. W. Churchilla 4, 130 67 Praha 3
svatek@vse.cz

Abstrakt. Tento příspěvek popisuje webovou službu, která umožňuje indexovat WWW stránky a vyhledávat informace v zaindexovaných dokumentech pomocí XML dotazovacího jazyka. Vyhledávací stroj využívá dříve publikovaný vícerozměrný přístup pro indexování XML dat, který transformuje XML dokument na body vícerozměrných prostorů. XML dotazy jsou pak vykonávány pomocí dotazů vícerozměrných datových struktur. Předpokládaným využitím popisované webové služby, kromě obecného dotazování strukturovaných dokumentů, je i zefektivnění extrakce informací znalostními metodami.

Klíčová slova: indexování XML, indexování webu, webová služba, extrakce informací z textu

1 Úvod

S rostoucím objemem informací na WWW se vyhledávání ve webových stránkách stává stále náročnější. V současné době je podstatná část informací uložena v HTML dokumentech, které mají povahu *slabě strukturovaných* dat. Jazyk *XML* [18] se stal de facto standardem pro popis slabě strukturovaných dat, s množstvím vyvinutých dotazovacích jazyků jako je např. *XPath*. Abychom mohli využít XML dotazovacího jazyka pro dotazování HTML dokumentů, musíme je převést na odpovídající XML dokumenty pomocí tzv. doznačkování. Poté můžeme s příslušným webovým sídlem pracovat jako s množinou XML dokumentů, která je však značně rozsáhlá a pro její efektivní dotazování je nutné ji předzpracovat - *indexovat*. Takto vzniklý index lze také využít, kromě obecného dotazování strukturovaných dokumentů, pro potřeby *extrakce informací z textu*. V tomto příspěvku popisujeme *webovou službu* umožňující indexování a dotazování webových stránek. Dnes existuje řada webových služeb pro indexování webových dokumentů, jako např. Google. Tyto služby ovšem indexují strukturované stránky jako čistý text a neberou v potaz vnitřní strukturu HTML dokumentů.

* Práce je částečně podporována grantem GAČR 201/03/1318.

V kapitole 2 je stručně zmíněna problematika indexování XML dat, webových služeb a extrakce informací z textu. V kapitole 3 je popsána samotná webová služba pro indexování a dotazování strukturovaných dokumentů. V závěru je shrnut obsah článku a nastíněny možnosti budoucí práce.

2 Stručný přehled problematiky

2.1 Indexování XML dat

Slovy databázové technologie je XML jazykem pro modelování dat [15]. *Správně strukturovaný* (well-formed) XML dokument nebo množina dokumentů je XML databáze a příslušné DTD (popř. schéma) jejím schématem. Implementace systémů vhodných pro efektivní uložení a dotazování XML dokumentů (tzv. nativní XML databáze) vyžaduje vývoj nových technik a je dnes jednou z klíčových otázek světa informačních technologií.

XML dokument je obvykle modelován jako graf, jehož uzly odpovídají příslušným elementům a atributům. Tento graf je nejčastěji strom (tzv. *XML strom*). Pro získání dat z XML databáze byly vyvinuty různé dotazovací jazyky, např. XPath [19] nebo XQuery [20]. Společným rysem těchto jazyků je použití regulárních výrazů pro vyjádření cesty grafem, kde cesta je sekvence názvů elementů (nebo atributů) od kořenového elementu k listovému. Uživatel pak v XML dokumentu naviguje pomocí různě dlouhých cest vyjádřených regulárním výrazem.

Pro efektivní uložení a dotazování XML dat není možné využít existující databázové modely. Při vykonávání dotazu daného regulárním výrazem cesty, je nutné procházet XML stromem. V tomto případě konvenční přístupy jako například SQL nebo OQL selhávají nebo nejsou příliš efektivní. Pro indexování a efektivní dotazování XML, nebo obecně slabě strukturovaných dat byla proto vyvinuta celá řada přístupů. Některé z nich jsou založeny na konvenčních *relačních technologiích* (např. Lore [14] a XISS [13]), jiné využívají speciální datové struktury pro reprezentaci XML dat, jako *trie* (např. Index Fabric [6]) nebo *vícerozměrné datové struktury* (např. XPath Accelerator [7]). Přehled přístupů pro indexování XML dat najdeme např. v [4].

2.2 Webové služby

Webové služby (Web Services - WS) [21] poskytují prostředky pro spolupráci mezi různými softwarovými aplikacemi, jenž mohou být provozovány na odlišných platformách v síťovém prostředí. WS představují v podstatě distribuovanou technologii, jakými jsou například RPC nebo CORBA. Architektura webových služeb nespecifikuje jakým způsobem jsou implementovány, ani neurčuje způsob jejich provázání. Účel WS je vykonání určité služby *poskytovatelem* (provider), jenž poskytuje příslušného agenta implementujícího danou službu a umožňuje tak *žadateli* (requester) tuto službu využívat.

Způsob výměny zpráv mezi agenty žadatele a poskytovatele je dokumentován v popisu webové služby (Web Service Description - WSD). WSD formálně popisuje

rozhraní WS pomocí jazyka WSDL (Web Service Description Language) [22]. WSDL tedy slouží k popisu formátu zpráv, datových typů, přenosového protokolu, specifikaci URL poskytovatelova agenta a jména služby. WSDL popisuje i chování služby, a to především odpověď na zprávu zaslanou této službě. V podstatě jde o dohodu mezi žadatelem a poskytovatelem určující záměr a výsledek interakce. Agenti žadatele a poskytovatele mezi sebou komunikují prostřednictvím zpráv. Nejčastěji je použit protokol SOAP (Service Oriented Architecture Protocol) [23], který je založen na XML. Požadavek také může být specifikován jako požadavek HTTP GET. V tomto případě, ale není možné využít rozšířených funkcí webových služeb. Samotné zprávy protokolu SOAP mohou být přenášeny pomocí protokolu HTTP, nebo i jiných protokolů, jako například SMTP a FTP.

2.3 Extrakce informací z textu

Práce s vnitřní strukturou HTML dokumentů má smysl zejména v případě, kdy chceme tuto strukturu přímo využívat pro odlišení významné a nevýznamné informace v daném kontextu, nebo dokonce pro rozlišení různých sémantických kategorií textu. Tato úloha je klíčová v tzv. *extrakci informací z textu* (information extraction - IE), kde jsou vybrané textové pasáže přiřazovány k sémantickým kategoriím, spojovány do záznamů, případně jsou z nich vytvářeny instance tříd a relací doménové ontologie. Obvyklým přístupem používaným v IE je zpracování *celého dokumentu*, což vychází z původní orientace na prosté textové dokumenty.

V případě webových dokumentů ovšem může být podstatným vodítkem pro určení třídy dané informace struktura kódu HTML. Tato struktura může být *globální*, platná pro celý dokument nebo jeho významnou část; s touto eventualitou počítají přístupy k extrakci založené na tzv. *wrapperech* [9]: dokument je jako celek převáděn do databázové podoby, výhradně nebo převážně na základě struktury elementů HTML, případně typů dat v nich obsažených. Struktura HTML ovšem může být také spíše *lokální* – pro určitou informaci může být významné např. umístění v určitém místě tabulky nebo v sousedství hypertextového odkazu; v takovém případě ovšem nelze abstrahovat jednoduchý wrapper pro celý dokument. Jediná možnost, jak automaticky získat z dokumentu užitečné informace, je aplikovat techniky na *vybraná místa* dokumentu.

Při omezení rozsahu extrakce se totiž mohou uplatnit i sofistikované a tudíž časově náročné extrakční techniky. Jednou z možností je plná *syntaktická analýza* přirozeného jazyka, která se právě kvůli své náročnosti při analýze webových stránek jinak zpravidla nepoužívá – v řadě případů však může mít smysl, např. u biografií osob nebo profilů firem psaných volným textem. Ještě užitečnější může často být (rovněž časově poměrně náročná) heuristická analýza vycházející z modelu *vizuální prezentace* informací pomocí kódu HTML [5,16], aplikovatelná na částečně strukturované textové informace jako jsou kontaktní adresy, produktové katalogy nebo biografie strukturované do kvazi–záznamů.

Lze předpokládat, že k identifikaci míst vhodných pro znalostně–intenzivní analýzu (v rámci rozsáhlejší kolekce HTML dokumentů) lze efektivně využít právě *XML index* ve spojení se *slovním indexem*. Zatímco ve slovním indexu se rychle vyhledají „slibná“ klíčová slova, XML index následně poskytne znalostním komponentám jejich přiměřeně rozsáhlé okolí.

3 AmphoraWS – webová služba pro vyhledávání ve strukturovaných dokumentech

3.1 Popis webové služby

Webová služba *AmphoraWS* [1] poskytuje několik metod umožňující uložení a dotazování webových dokumentů. Metoda *Index* umožňuje zaindexování webového sídla. Parametrem metody je URL kořenové stránky a výsledkem je řetězec s jednoznačným číslem databáze. Příkladem může být URL *http://www.abclinux.cz/*. HTML stránky jsou převedeny na XML a poté indexovány vícerozměrným přístupem (viz kapitola 3.2). Metoda *DatabaseList* vrací řetězec s jedinečnými čísly zaindexovaných webových sídel. Metoda *ResourceList* má jeden parametr *dbId* specifikující jedinečné číslo zaindexovaného webového sídla, a vrací seznam URL zaindexovaných stránek. Metoda *Query* umožňuje dotazování nad specifikovanou databází. Jejím prvním parametrem je *dbId* databáze a druhým dotaz. V budoucnu se počítá s využitím vhodně zvolené podmnožiny dotazovacího jazyka XPath.

Webová služba je naprogramována v jazyce C# nad platformou .NET, přístup pro indexování XML dokumentů je implementován v C++ a využívá knihovnu tříd ATOM (Amphora Tree Object Model). Tato knihovna, která umožňuje implementaci perzistentních datových struktur, je vyvíjena výzkumnou skupinou ARG [2].

3.2 Vícerozměrný přístup pro dotazování XML dat založené na klíčových slovech

Webová služba AmphoraWS využívá *vícerozměrný přístup* pro indexovaní XML dat [10,11]. Tento přístup umožňuje efektivní dotazování indexovaných dat pomocí podmnožiny jazyka XPath. V [10] je popsáno rozšíření tohoto přístupu o dotazování založené na klíčových slovech – termech textového obsahu elementů.

3.2.1 Vícerozměrný přístup pro indexování XML dat

Vícerozměrný přístup je založen na myšlence modelovat XML strom jako množinu cest od kořene ke všem listovým uzlům. Každý element je ohodnocen jedinečným číslem id_N, které je zvyšováno při průchodu stromem do hloubky. Tímto způsobem je zachováno *uspořádání dokumentu* (document order). Všem názvům elementů a atributů a jejich hodnotám s_i jsou přiřazena jedinečná čísla $id_T(s_i)$. *Index termů* pak obsahuje všechny tyto termy a jejich id_T. Každý typ cesty (tzv. *značkovaná cesta*) lp_i je převeden na bod vícerozměrného prostoru a je mu přiděleno jedinečné číslo $id_{LP}(lp_i)$. Tyto body jsou pak uloženy v *indexu značkovaných cest*. *Cesty* jsou také převedeny na body vícerozměrného prostoru a uloženy v *indexu cest*. Bod je tvořen jedinečným číslem příslušné značkované cesty id_{LP}, čísly elementů id_N, a číslem listové hodnoty id_T. Výsledná *n*-tice $(id_{LP},id_{N0},\ldots,id_{Nl},id_T)$ představuje bod v *n*-rozměrném prostoru a množinu bodů získaných z XML dokumentu můžeme indexovat pomocí *vícerozměrných datových struktur*, např. R-stromem [8]. Takto zaindexovaná XML data jsou dotazována pomocí dotazů vícerozměrných datových struktur – dotazů bodových a rozsahových [8].

```
<!DOCTYPE books [
  <!ELEMENT books(book)>
  <!ELEMENT book(title,author)>
  <!ATTLIST book id CDATA #REQUIRED>
  <!ELEMENT title(#PCDATA)>
  <!ELEMENT author(#PCDATA)>
]>
```

(a)

```
<?xml version="1.0" ?>
<books>
  <book id="003-04312">
    <title>The Two Towers</title>
    <author>J.R.R. Tolkien</author>
  </book>
  <book id="001-00863">
    <title>The Return of the King</title>
    <author>J.R.R. Tolkien</author>
  </book>
  <book id="045-00012">
    <title>Catch 22</title>
    <author>Joseph Heller</author>
  </book>
</books>
```

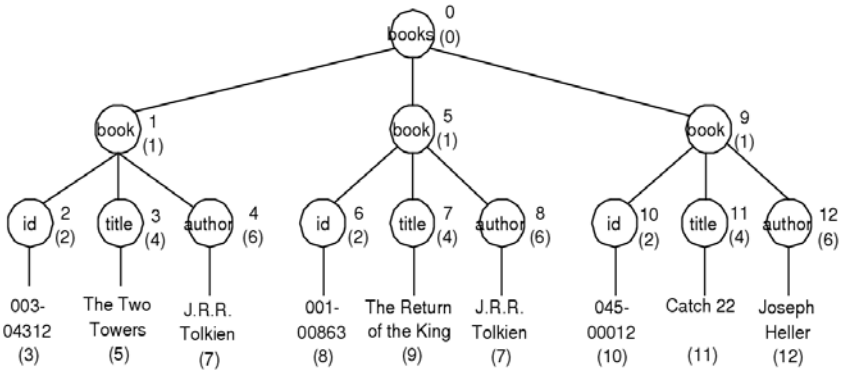
(b)

Obr. 1. (a) DTD dokumentů obsahující informace o knihách a jejich autorech
(b) Správně strukturovaný XML dokument platný k tomuto DTD.

Nyní uvedeme příklad tvorby indexů. Vezměme XML dokument z obrázku 1 jehož XML strom je uveden na obrázku 2. Tento XML dokument obsahuje cesty:

- 0,1,2,'003-04312'; 0,5,6,'001-00863' a 0,9,10,'045-00012' náležící ke značkované cestě books,book,author.
- 0,1,3,'The Two Towers'; 0,5,7,'The Return of the King' a 0,9,11,'Catch 22' náležící ke značkované cestě books,book,title.
- 0,1,4,'J.R.R. Tolkien'; 0,5,8,'J.R.R. Tolkien' a 0,9,12,'Joseph Heller' náležící ke značkované cestě books,book,author.

V tabulce 1 je reprezentován obsah indexů pro tento XML dokument.



Obr. 2. Strom XML dokumentu z obrázku 1 s jednoznačnými čísly $id_N(u_i)$ elementů a atributů u_i a jednoznačnými čísly $id_T(s_i)$ názvů elementů a atributů a jejich hodnot s_i (hodnoty v závorkách).

Tabulka 1. Obsah indexů pro XML dokument z obrázku 1.

index termů		index značkovaných cest		index cest
s_i	id_T	$(id_{T0},id_{T1},id_{T2})$	id_{LP}	$(id_{LP},id_{N0},id_{N1},id_{N2},id_T)$
books	0	(0,1,2)	0	(0,0,1,2,3)
book	1	(0,1,4)	1	(1,0,1,3,5)
id	2	(0,1,6)	2	(2,0,1,4,7)
003-04312	3			(0,0,5,6,6)
…				(1,0,5,7,9)
Joseph Heller	12			…
				(2,0,9,12,12)

Nyní popíšeme implementaci XPath dotazu /books/book[author = `J.R.R. Tolkien`], který je proveden ve třech navazujících fázích.

1. Nalezení jednoznačných čísel id_T termů dotazu v indexu termů. V tomto případě termů books, book, author a J.R.R. Tolkien s id_T 0,1,6 a 7.
2. Nalezení jedinečných čísel značkovaných cest id_{LP} v indexu značkovaných cest. Značkovaná cesta books,book,author s id_{LP} 2 je nalezena bodovým dotazem 0,1,6.
3. Nalezení odpovídajících bodů v indexu cest, jenž je realizováno rozsahovým dotazem ve vícerozměrné datové struktuře. Dotazem jsou hledány cesty s definovanou značkovanou cestou a termem, čemuž odpovídá rozsahový dotaz $(id_{LP},*,...,*,id_T)$. Poznamenejme, že rozsahový dotaz $(id_{LP},min(D),...,min(D),id_T)$: $(id_{LP},max(D),..., max(D),id_T)$, specifikovaný nad doménou $D \in \Omega$ vícerozměrného prostoru $\Omega=D^n$, je možné zkráceně zapsat $(id_{LP},*,...,*,id_T)$. V tomto případě dotaz (2,*,*,*,7) vyhledá body (2,0,1,4,7) a (2,0,5,8,7), které reprezentují cesty 0,1,4,'J.R.R. Tolkien' a 0,5,8,'J.R.R. Tolkien'.

3.2.2 Rozšíření o dotazování založené na klíčových slovech

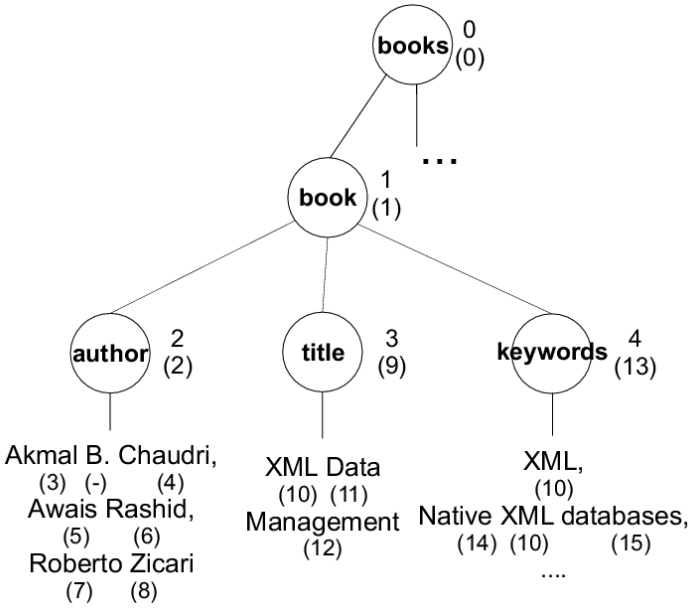
V [10] je uveden *index cest-značkovaných cest-termů* (*Path-Labelled path-Term* (*PLT*)), který umožňuje efektivní dotazování XML dat založené na klíčových slovech získaných z textového obsahu elementů. Je definován operátor ~=, který je splněn pokud daný element obsahuje specifikovaný term. K řešení tohoto problému je nutné indexovat každý term obsahu elementů. Rovněž musí být zachována informace o příslušnosti termů k jejich cestě a odpovídající značkované cestě. Výše uvedené požadavky splňuje *index cest-značkovaných cest-termů*, jenž indexuje prostor dimenze 3 a obsahuje body $(id_P(p_i),id_{LP}(lp_i),id_T(s_i))$. První souřadnice tohoto bodu obsahuje jedinečné číslo $id_P(p_i)$ cesty p_i. Stejné $id_P(p_i)$ je nutné uložit v první souřadnici bodu reprezentujícího cestu p_i v indexu cest. Druhá souřadnice představuje jedinečné číslo $id_{LP}(lp_i)$ značkované cesty lp_i uložené v indexu značkovaných cest. Poslední souřadnice obsahuje jedinečné číslo $id_T(s_i)$ indexovaného termu, které je získáno z indexu termů.

Obrázek 3 ukazuje část dokumentově orientovaného XML dokumentu, pro který bude popsána tvorba indexu cest-značkovaných cest-termů.

```
<books>
  <book>
    <autor>Akmal B. Chaudri, Awais Rashid, Roberto Zicari</autor>
    <title>XML Data Management</title>
    <keywords>XML, Native XML databases, …</keywords>
  </book>
</books>
```

Obr. 3. Část dokumentově orientovaného XML dokumentu.

Obrázek 4 reprezentuje strom části XML dokumentu uvedeného na obrázku 3 spolu s unikátními čísly uzlů a termů (hodnoty v závorkách). V tabulce 2 je zobrazen obsah všech čtyř indexů pro tento XML dokument. Znak „-“ na místě jedinečného čísla termu (viz obrázek 4) značí, že určité termy není nutné indexovat v indexu termů. Termy jako spojky, předložky apod. mohou být odstraněny pomocí *stop listu* známého z oblasti *vyhledávání dat* (Information Retrieval) [3].



Obr. 4. Strom části XML dokumentu z obrázku 3.

Tabulka 2. Obsah indexů pro XML dokument uvedený na obrázku 3.

index termů		**index značkovaných cest**		**index cest**	**PLT index**
s_i	id_T	$(id_{T0},id_{T1},id_{T2})$	id_{LP}	$(id_P,id_{LP},id_{N0},id_{N1},id_{N2})$	(id_P,id_{LP},id_T)
books	0	(0,1,2,0)	0	(0,0,0,1,2)	(0,0,3)
book	1	(0,1,9,1)	1	(1,1,0,1,3)	(0,0,4)
author	2	(0,1,13,2)	2	(2,2,0,1,4)	(0,0,5)
Akmal	3				(0,0,6)
…					(0,0,7)
XML	10				(0,0,8)
Data	11				(1,1,10)
Management	12				(1,1,11)
keywords	13				(1,1,12)
Native	14				(2,2,10)
databases	15				(2,2,14)
					(2,2,15)

Nyní popíšeme vykonání dotazu /books/book[keywords~='XML']/title nad XML dokumentem uvedeném na obrázku 3.

1. Nalezení $id^1_{LP} = id_{LP}(\text{'books,book,keywords'})$ a $id^1_T = id_T(\text{'XML'})$.
2. Vykonání úzkého rozsahového dotazu $(*,id^1_{LP},id^1_T)$ v indexu PLT. Výsledkem je k jedinečných čísel $id(p_1),...,id(p_k)$ relevantních cest $p_1,..., p_k$.
3. Vykonání komplexního rozsahového dotazu $(id_P(p_1),id^1_{LP},*,...,*),...,(id_P(p_k),id^1_{LP},*,...,*)$ v indexu cest. Výsledkem jsou body reprezentující relevantní cesty.
4. Nalezení $id^2_{LP} = id_{LP}(\text{'books,book,title'})$.
5. Vykonání osy *child* jazyka XPath s id^2_{LP} v indexu cest. Výsledkem je m cest $p^f_1, ...,p^f_m$. Osa child jazyka XPath je realizována pomocí sekvence rozsahových dotazů (viz [12]).
6. Vykonání komplexního rozsahového dotazu $(id_P(p^f_1),id^2_{LP},*),...,(id_P(p^f_m),id^2_{LP},*)$. Výstupem je kolekce jedinečných čísel termů $id_T(s_i)$. Příslušné řetězce s_i jsou získány z indexu termů a tvoří výsledek dotazu.

4 Závěr

V tomto příspěvku je popsána vyvíjená webová služba pro indexování WWW stránek. Webové stránky jsou převedeny na XML dokumenty a poté indexovány pomocí vícerozměrného přístupu. V budoucnu bychom chtěli umožnit dotazování zaindexovaných stránek pomocí vhodně zvolené podmnožiny XML dotazovacího jazyka (jako je např. XPath). AmphoraWS bude využívána nejen k samotnému indexování a vyhledávání v XML dokumentech, ale i pro potřeby extrakce informací z textu. V tomto směru nyní probíhá práce na využití zde popsané služby v projektu Rainbow [17].

Literatura

1. ARG: *AmphoraWS*. http://pckp311a.vsb.cz/amphoraws/amphoraws.asmx, 2004.
2. ARG: *Amphora Research Group*. http://www.cs.vsb.cz/arg/, 2004.
3. R. Baeza-Yates and B. Ribiero-Neto. *Modern Information Retrieval*. Addison Wesley, New York, 1999.
4. D. Barashev, M. Krátký, and T. Skopal: *Modern Approaches to Indexing XML Data*. In TRANSACTIONS of VŠB-Technical University of Ostrava, Computer Science and Mathematics Series, Ostrava, Czech Republic, Volume 2, ISBN 80-248-0455-7, ISSN 1213-4279, 2003.
5. R. Burget: Information Extraction from HTML Documents Based on Logical Document Structure. Doktorská disertační práce, VUT Brno 2004.
6. B. Cooper, N. Sample, M.J. Franklin, G.R. Hjaltason, and M. Shadmon: A Fast Index for Semistructured Data. In *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB'01)*, Rome, Italy, pages 341–350. Morgan Kaufmann, 2001.
7. T. Grust: Accelerating XPath Location Steps. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, Madison, USA. ACM Press, 2002.
8. A. Guttman. R-Trees: A Dynamic Index Structure for Spatial Searching. In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, Annual Meeting, Boston, USA, pages 47–57. ACM Press, June 1984.
9. C.A. Knoblock, S. Minton, J.L. Ambite, N. Ashish, P.J. Modi, I. Muslea, A.G. Philpot, and S. Tejada: Modeling Web Sources for Information Integration. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)*, Madison, WI, 1998.
10. M. Krátký and M. Andrt: On Efficient Part-match Querying of XML Data. In *Proceedings of the Annual International Workshop on Databases, Texts, Specifications and Objects (DATESO 2004)*. Desná–Černá Říčka, Czech Republic, ISBN 80-248-0457-3. Published on CEUR Workshop Proceedings, ISSN 1613–0073, http://CEUR-WS.org/Vol-98/, 2004.
11. M. Krátký, J. Pokorný, and V. Snášel: Implementation of XPath Axes in the Multi-dimensional Approach to Indexing XML Data. In *Current Trends in Database Technology, International Workshop on Database Technologies for Handling XML information on the Web, DataX, EDBT 2004*, Heraklion - Crete, Greece. volume 3268 of Lecture Notes in Computer Science, Springer-Verlag, 2004.
12. M. Krátký, T. Skopal, and V. Snášel. Multidimensional Term Indexing for Efficient Processing of Complex Queries. *Kybernetika, Journal of the Academy of Science of the Czech Republic*, 40(3):381–396, 2004.
13. Q. Li and B. Moon. Indexing and Querying XML Data for Regular Path Expressions. In *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB'01)*, Rome, Italy. Morgan Kaufmann, 2001.

14. J. McHugh, S. Abiteboul, R. Goldman, D. Quass, and J. Widom. Lore: a database management system for semistructured data. *ACM SIGMOD Record*, 26(3):54–66, 1997.
15. J. Pokorný: *XML: a challenge for databases?*. Chap. 13 In: Contemporary Trends in Systems Development. Ed. Maung K. Sein, Kluwer Academic Publishers, Boston, pp. 147-164, 2001.
16. V. Svátek, J. Bráza, V. Sklenák: Towards Triple-Based Information Extraction from Visually-Structured HTML Pages. In *Poster Track of the Twelfth International World Wide Web Conference (WWW'2003)*, Budapest, 2003.
17. V. Svátek a kol.: *Rainbow – Reusable Architecture for Intelligent Brokering of Web information access*. http://rainbow.vse.cz , 2004.
18. W3 Consortium: *Extensible Markup Language (XML) 1.0*. 1998, http://www.w3.org/TR/REC-xml.
19. W3 Consortium. *XML Path Language (XPath) Version 2.0*, W3C Working Draft, 15 November 2002, http://www.w3.org/TR/xpath20/.
20. W3 Consortium: *XQuery 1.0: An XML Query Language*, W3C Working Draft. 15 November 2002, http://www.w3.org/TR/xquery/.
21. W3 Consortium: *Web Services Architecture*, W3C Working Group Note 11 February 2004, http://www.w3c.org/TR/ws-arch/
22. W3 Consortium: Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, W3C Working Draft, 2004, http://www.w3c.org/TR/wsdl20/
23. W3 Consortium: *SOAP Version 1.2 Part 0: Primer*, W3C Recommendation 24 June 2003, http://www.w3c.org/TR/soap12-part0/

Annotation:

AmphoraWS – Web service for querying semi-structured data

In this article a Web service for indexing and querying Web pages is described. An XML query language is possible to use for querying the indexed Web sites. The search engine applies the previous published multi-dimensional approach to indexing XML data. The application of the Web service is a querying of semi-structured data as well as a making the information extraction more effective using knowledge methods.