Doktorandský den ’04

Ústav informatiky

Akademie věd České republiky

Paseky nad Jizerou, 29. září – 1. říjen 2004

Obsah

Rekonstrukce databázového modelu na základě dat (studie proveditelnosti)

doktorand:
ING. MARTIN ŘIMNÁČ
Katedra kybernetiky, FEL ČVUT Praha
rimnacm@cs.cas.cz

školitel:
ING. JÚLIUS ŠTULLER CSC.
Ústav informatiky AV ČR Praha
stuller@cs.cas.cz

obor studia:
Databázové systémy
číselné označení: I

Abstrakt

Příspěvek popisuje provedenou studii proveditelnosti databázově orientované části systému zajišťujícím automatickou extrakci dat z webových zdrojů (formáty XHTML, XML, CSV). Úkolem této části je transformace dat do automaticky vygenerovaného relačního modelu, který může být následně užit pro realizaci myšlenek sémantického webu.

V úvodní části je uvedena motivace pro implementaci takového nástroje. Součástí příspěvku je i částečné ohlédnutí za již implementovanými metodami, které autor v současné době zpracovává. V poslední části je nastíněna fuzzyfikace problematiky.

1. Motivace

Tak jako kola větrných mlýnů se nebudou točit bez větru, tak koncepce sémantického webu nebude přijata širokou veřejností bez relevantně využitelných informací v takovém rozsahu, jaký dnes nabízejí webové servery jak v podobě XHTML stránek, tak v podobě stažitelných dokumentů rozličných aplikací nebo různých webových služeb. Z toho důvodu je vhodné se zabývat nástrojem, který by pokud možno automaticky data z webových serverů získával a konvertoval je do strojově dále zpracovatelné podoby (např. relační databáze, XML, RDF). Součástí získávání dat může být i zahrnutí jejich dostupné sémantiky, zpravidla v rámci stránky vyjádřené pomocí formátování dokumentu.

Tato práce navazuje na diplomovou práci doktoranda [1], která se zabývala mapováním obecných webových prezentací. Základní mapovací jednotkou je webová stránka, výstupem algoritmu pak uspořádání stránek do stromové struktury. Jediným předpokladem tohoto algoritmu je strukturovanost webové prezentace. Současné úsilí hledá odpověď na otázku, zda-li lze efektivně mapovat i na nižší úrovni, než-li je webová stránka, tedy na úrovni strukturovaného obsahu stránky. Současné vyhledávací služby vracejí odkazy na stránky, které hledanou informaci obsahují. Je ale možné najít požadovanou informaci samu? Tím se

dostáváme zpět k sémantickému webu. Lze tedy implementovat automatický nástroj, který by dokázal data najít, extrahovat a dále je prezentovat v kontextu jiných informací?

Praktickou motivací pro tuto úlohu je sledování časově proměnných veličin, např. cen různých počítačových komponent nebo vývoj devizových kurzů měn. Na základě takto získaných informací se můžeme ptát, který prodejce má nejvýhodnější služby, jaké jsou alternativy výrobků, jaké jsou trendy. A to bez striktní podmínky publikování informací jejich poskytovatelem ve formátu podporujícím paradigma sémantického webu. Nástroje využívající myšlenek sémantického webu navrhovanou metodikou mohou získat informace a dokáží prezentovat svoje přednosti. To může vést k všeobecnému přijetí této koncepce a budoucí moderní webové prezentace již budou "samozřejmě" zahrnovat i sémantiku prezentovaných dat.

2. Současný stav problematiky

Tento příspěvek se zaměřuje na tu část problematiky, která se zabývá rekonstrukcí databázového modelu na základě vstupních dat. Tato úloha je v různých souvislostech řešena od zavedení relačních databází, první dílčí výsledky jsou publikovány od roku 1975 [2].

Poměrně velká pozornost byla v počátcích věnována sledování dotazů (příp. transakcí) [2, 3]. Na základě množin atributů, ke kterým bylo přistupováno v rámci jedné operace, byla statisticky vyhodnocována příbuznost atributů. Podle různých kritériích na hodnotu vzájemné příbuznosti atributů pak byly generovány množiny atributů, které byly sdružovány do relací. Tento způsob nezaručuje žádnou z normálních forem, spíše je využitelný pro fyzický návrh databáze a předzpracování (předpřipravení) dotazů.

Pro logický návrh jsou vhodnější metody analyzující závislosti mezi atributy. Jednotlivé závislosti mezi atributy mohou být znázorňovány pomocí hypergrafů [4]. Úkolem algoritmu je rozdělit relaci obsahující všechny atributy schématu do subrelací tak, aby tyto subrelace byly v požadované normální formě nebo splňovaly jiná kritéria. Metody lze rozdělit podle přístupu, buď přistupují shora dolů nebo zdola nahoru.

Přístup shora dolů spočívá v `dekompozici schématu`. V principu se algoritmus inicializuje jedinou relací obsahující všechny atributy schématu a tuto relaci testuje na podmínky specifikované normální formy [5] nebo na množiny různých druhů závislostí [6]. Pokud relace těmto podmínkám nevyhovuje, je rozdělena. Na dekomponované schéma jsou kladeny různé požadavky jako minimální redundance, reprezentativnost a separace [7].

Naopak přístup zdola nahoru vychází z funkčních závislostí a postupně `odstraňuje redundantní závislosti` vznikající díky jejich tranzitivitě (popsána dále). Odstraňování může být provedeno na základě analýz uzávěrů množiny atributů [8, 9] nebo při uvažování prvků těchto uzávěrů jako vzájemných podmnožin atributů [10].

3. Navrhovaná metodika

V současné době autor příspěvku analyzuje již navržené algoritmy v chronologickém pořadí a konfrontuje je s vlastní intuitivně navrženou metodikou, ke které byla provedena níže popsaná studie proveditelnosti. Cílem je najít algoritmus s přístupem zdola nahoru, který by bylo možné fuzzyfikovat a při rekonstrukci modelu uvažovat fuzzy-závislosti místo klasických závislostí.

Jako nevýhodu všech výše popsaných algoritmů můžeme označit fakt, že pracují se striktní definicí funkčních (příp. i jiných) závislostí, kterou některé závislosti v obecném případě na reálných datech nemusí splňovat. Předpokládejme tedy, že malé procento záznamů těchto dat danou funkční závislost nevykazuje. Výše uvedené algoritmy používají nefuzzy vstupy, tedy toto procento záznamů ignorují (čímž prakticky provedou defuzzyfikaci hned na svém vstupu) nebo docházejí k situaci neodpovídajícím schématům (uvažuje se pouze podmnožina skutečných závislostí). Alternativní přístup, kterým se autor hodlá zabývat, uvažuje fuzzy závislosti po celou dobu dekompozice a defuzzyfikace je provedena až na výsledném dekomponovaném schématu.

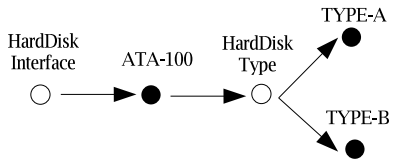
Podobně jako většina výše uvedených algoritmů omezme vstupní informace následovně:

- Data budou interně uložena formou stromu atributů a jejich hodnot, která umožňuje uložit v databází data, jejichž strukturu apriori neznáme. Tato reprezentace dat bude sloužit jako zdroj informací pro vygenerování relačního schématu.
- Data ve svém relačním schématu neobsahují cykly. Vylučujeme tak relace mezi stejnými entitními typy, např. relaci potomek. Tato podmínka vede na zjednodušení úlohy, některé úlohy vykazují pouze polynomiální složitost při acyklicitě [4].
- Žádné další informace nejsou k dispozici.
- Hodnoty atributů pro jednoduchost předpokládejme diskrétní.
- Jednoatributové primární klíče každé subrelace.

4. Integrace dat

Pro účely studie proveditelnosti byla použita podstatně zjednodušená verze grafového modelu sloužícího původně k integraci dat XML dokumentů [11].

- Uzly stromu jsou dvojího druhu, buď představují jméno atributu $attr_i$ nebo jeho hodnotu val_{ij}.
- Dvojice uzlů $(attr_i, val_{ij})$ je propojena orientovanou hranou.
- Všechny takové dvojice jednoho záznamu jsou hierarchicky propojeny tak, aby graf vykazoval stromovou strukturu.



Obrázek 1: Příklad struktury integrovaných dat

Kvalita integrace je dána počtem hran grafu vztaženou na počet uložených záznamů. Poměrně snadno lze ukázat, že počet hran je minimální, pokud posloupnost atributů $\{A_k\}$ je hierarchicky uspořádána tak, že

$$|D(A_i)| < |D(A_j)| \Rightarrow i < j\,. \tag{1}$$

Symbol $|D(A_k)|$ označuje počet prvků (diskrétních hodnot) domény k–tého atributu.

Takto provedené uspořádání atributů však nic neříká o vztazích mezi atributy, příp. o dekompozici atributů do databázového schématu a je tudíž pro získávání sémantických informací na základě dat nepoužitelné.

5. Závislosti atributů

Pro dekompozici relací mezi atributy použijeme definici funkční závislosti a využijeme některé vlastnosti těchto závislostí. Je užito značení podle [12]. Pro stanovení funkční závislosti používáme intenzivního přístupu.

5.1. Výklad základních pojmů

Definujme `funkční závislost` dvou atributů X a Y téhož entitního typu E s instancemi $R = \{r_k\}$. Říkáme, že atribut Y je závislý na atributu X (značíme $X \rightarrow Y$) právě tehdy, když

$$\forall r_i, r_j \in R : y(r_i) \neq y(r_j) \Rightarrow x(r_i) \neq x(r_j)\,. \tag{2}$$

V zobecněném případě pak můžeme hovořit o závislostech množin atributů.

$$\forall r_i, r_j \in R : \overline{y}(r_i) \neq \overline{y}(r_j) \Rightarrow \overline{x}(r_i) \neq \overline{x}(r_j)\,, \tag{3}$$

což v atomickém zápisu znamená

$$\forall r_i, r_j \in R : y_0(r_i) \neq y_0(r_j) \wedge \cdots \wedge y_m(r_i) \neq y_m(r_j) \Rightarrow x_0(r_i) \neq x_0(r_j) \vee \cdots \vee x_n(r_i) \neq x_n(r_j)\,. \tag{4}$$

Naopak atributy označíme za nezávislé (značíme $X \nrightarrow Y$), pokud

$$\exists r_i, r_j \in R, i \neq j : y(r_i) \neq y(r_j) \wedge x(r_i) = x(r_j)\,. \tag{5}$$

Atributy X a Y označíme jako vzájemně závislé (značíme $X \leftrightarrow Y$), pokud

$$X \leftrightarrow Y \Leftrightarrow X \rightarrow Y \wedge Y \rightarrow X\,. \tag{6}$$

Pro naše účely doplňme k těmto definicím ještě následující 2 tvrzení:

Transivita Nechť X, Y, Z jsou atributy entitního typu E. Pak

$$X \rightarrow Z \wedge Z \rightarrow Y \Rightarrow X \rightarrow Y\,. \tag{7}$$

Hierarchie Nechť $\overline{X}, \overline{Y}, \overline{Z}$ jsou neprázdné množiny atributů. Pak

$$\overline{Z} \subset \overline{X} : \overline{Z} \rightarrow \overline{Y} \Rightarrow \overline{X} \rightarrow \overline{Y}\,. \tag{8}$$

5.2. Testování funkčních závislostí

Předpokládejme, že ve výše popsané stromové struktuře máme uloženo celkem (C) záznamů a testujeme funkční závislost $X \rightarrow Y$. Z této struktury extrahujme všechny hodnoty atributu Y a po větvích stromu k nim nalezneme odpovídající hodnoty atributu X. Není–li odpovídající hodnota atributu X nalezena, uvažujeme, že atribut nabývá hodnoty NULL. Seřaďme nyní extrahované atributy podle hodnot atributu X a sekundárně pomocí atributu Y.

Označme e_i extrahované dvojice atributů. Pak případ, kdy je porušena funkční závislost, lze detekovat pomocí

$$x(e_i) = x(e_{i-1}) \wedge y(e_i) \neq y(e_{i-1})\,. \tag{9}$$

Počet takových záznamů označíme jako c. Abychom získali nefuzzy funkční závislost, defuzzyfikujeme takto otestovanou závislost, např. pomocí prahování počtu c záznamů (např. ve významu maximální přípustné chyby f):

$$\begin{array}{l} \frac{c}{C} < f \Rightarrow X \rightarrow Y \\ \frac{c}{C} > f \Rightarrow X \nrightarrow Y \end{array} \qquad f \in <0,1>\ . \tag{10}$$

Diskutujme výpočetní složitost testu. Mějme N atributů. Přijmeme zjednodušující předpoklad, že všechny záznamy popisují jednu relaci (mají shodné atributy). Pak extrakce hodnot atributů je složitosti $o(NC)$. Efektivní složitost je nižší díky stromovému uspořádání.

Druhou složkou je seřazení hodnot, uvažujme $o(C \log(C))$.

Poslední složkou je samotný test spočívající v průchodu všech záznamů a porovnání se záznamem předcházejícím, tj. složitost $o(C)$. Efektivní složitost může být podstatně nižší díky možnosti agregace shodných záznamů.

Celková složitost je dána součtem dílčích složitostí:

$$o(NC) + o(C\log(C)) + o(C)\,. \tag{11}$$

Jak je patrno, nejsložitější operací je extrakce hodnot. Proto je vhodné provést extrakci pouze jednou ale pro všechny dvojice atributů. Složitost testu všech dvojic:

$$o(NC) + N(N-1)(o(C\log(C)) + o(C)) = o(NC) + o(N^2C\log(C)) + o(N^2C) = o(N^2C\log(C))\,. \tag{12}$$

5.3. Matice závislostí

Při provedené studii se ukázalo vhodné zavést pojem `matice závislostí`. Nechť model obsahuje N atributů. Pak matice závislostí prvního řádu

$$M^1 = \{m_{ij}\}, m_{ij} = \begin{cases} -1 & A_i \to A_j \\ 1 & A_j \to A_i \\ 0 & jinak \end{cases} \quad i,j = 1..N\,. \tag{13}$$

Matice závislostí prvního řádu umožňuje dekompozici níže uvedených modelů závislostí.

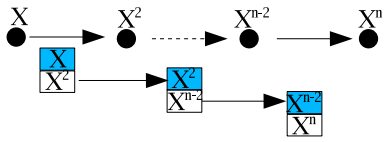
5.3.1 Model hierarchické závislosti: Tato závislost říká, že primární klíč je cizím klíčem předchozí relace, přičemž relaci tvoří dvojice (primární, cizí) klíč. Pro model formálně platí:

$$\forall i = 2..N : A_i \to A_{i-1}\,. \tag{14}$$

Pak díky transitivitě (7) platí, že

$$\sum_k m_{ik} > \sum_k m_{jk} \Leftrightarrow i < j\,. \tag{15}$$

Pokud vstupní data (s libovolným uspořádáním atributů) lze popsat pomocí modelu (14), pak tento model lze jednoznačně ze vstupních dat rekonstruovat na základě uspořádání atributů podle kritéria (15).



Obrázek 2: Příklad hierarchické závislosti

5.3.2 Model hierarchické závislosti se závislými atributy: Tento model vychází z předchozího modelu (14), avšak každý primární klíč je vzájemně závislý s jiným jedním atributem, který není závislý na žádném ze svých následníků. Model formálně popíšeme:

$$\forall i = 1..N/3 \quad \forall k > 3i-2 : A_{3i} \to A_{3i-1} \leftrightarrow A_{3i-2} \wedge A_k \nrightarrow A_{3i-1}\,. \tag{16}$$

Opět na základě transitivity (7) dokážeme, že

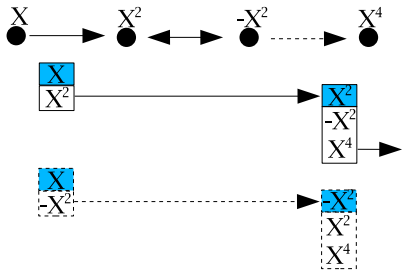
$$\sum_k m_{ik} > \sum_k m_{jk} \Rightarrow i < j\,. \tag{17}$$

Díky faktu, že

$$\forall s = 1..N/3 : \sum_k m_{ik} = \sum_k m_{jk} \Leftrightarrow i = 3s - 1 \wedge j = 3s - 2 \tag{18}$$

platí implikace (17) pouze jedním směrem.

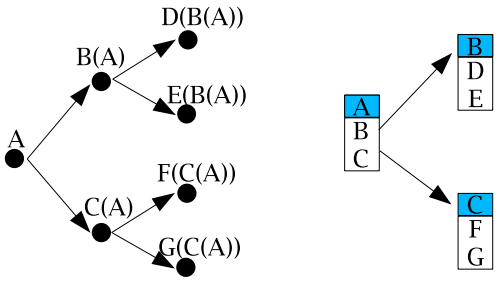
Určení primárního klíče není jednoznačné, protože k cizích klíčů nadřazené subrelace je navzájem závislých. Všechny cizí klíče nadřazené subrelace jsou rovnocennými kandidáty na primární klíč subrelace. Rekonstruovaný model je tedy jednoznačný až na určení primárního klíče podřízené subrelace.



Obrázek 3: Příklad hierarchické závislosti se závislými atributy

5.4. Modely s vícearitními závislostmi

Na základě předchozího odstavce se můžeme domnívat, že použitím podobných kritérií bude možné postupně rozšiřovat množinu modelů závislostí. Uvažme, že libovolná z relací má navzájem nezávislé cizí klíče. Pak sice selhává postup z minulého odstavce, kritérium však můžeme rozšířit o matice závislostí vyšších řádů (řád odpovídá aritě závislosti) a toto kritérium bude využívat vlastnosti hierarchických závislostí podle (8).



Obrázek 4: Příklad vícearitní závislosti

Selhání modelování se v tomto případě projevuje porušením podmínky

$$A_i \rightarrow A_j \Leftrightarrow i > j\,. \tag{19}$$

Tato podmínka platila ve všech předchozích modelech, avšak neplatí v případě, že relace obsahuje více nežli jeden cizí klíč.

5.5. Studie proveditelnosti

Na základě intuitivní myšlenky autora byla provedena studie proveditelnosti. Pro úplnost byla nastíněna i zcela původní myšlenka (čistě grafový přístup, odvození na základě počtu prvků domén jednotlivých atributů, odvozený z (1)), která však byla slepá, avšak poznatky z tohoto řešení lze parciálně využít pro snížení výpočetní složitosti a paměťových nároků na uložení "surových" dat. Studie proveditelnosti poukázala na směr dalšího řešení problematiky.

Metodice můžeme vytýkat kombinatorickou explozi při testování vícearitních funkčních závislostí, avšak v tomto kontextu lze argumentovat podstatnou redukcí prohledávaného stavového prostoru. Navíc při použití fuzzy–závislostí je možné vícearitní funkční závislosti pouze odhadovat a testovat teprve v případě použití takové závislosti ve výsledném databázovém schématu.

Z uvedených dílčích výsledků můžeme usuzovat, že řešení této úlohy pomocí nastíněné metodiky má smysl. Při následné rešerži literatury se jeví perspektivní použití některých myšlenek jiných algoritmů, např. [10]. Zajímavým v tomto kontextu může být i nasazení genetických algoritmů na matici závislostí (13) při vhodně definovaném kritériu tak, aby nebylo nutné procházet NP–úplné testy na normální formu subrelace [5].

Během studie byly některé části implementovány na databázovém serveru PostGres, což přináší vedle ověření teoretických odvození i možnost testování na reálných datech. Tyto výsledky je možné zpřístupnit v rámci osobních stránek autora [13].

6. Budoucí práce

Během studie proveditelnosti bylo rovněž i experimentováno přímo s fuzzy závislostmi atributů, tedy bez provedení defuzzyfikace (10). Ty lépe vystihují závislosti mezi vstupními daty, která mohou být zatížena chybami nebo mohou být víceznačná.

Fuzzy míru příslušnosti můžeme formálně zavést jako procentuální vyjádření počtu záznamů (podle (9)), které splňují testovanou funkční závislost, tedy

$$\mu_{ij} = \frac{1-c}{M}, \text{ kde } M \text{ je počet všech záznamů, z nichž } 1-c \text{ splňuje } A_i \rightarrow A_j \,. \tag{20}$$

Matici závislostí pak modifikuje

$$\widetilde{M}^1 = \{\mu_{ij} - \mu_{ji}\} \quad \forall i,j = 1..N \,. \tag{21}$$

Tato modifikace umožňuje pracovat po celou dobu dekompozice z fuzzy–závislostmi. To vede k přesnějšímu popisu ze vstupních dat extrahované sémantiky, obzvláště vzhledem k intenzivnímu, daty orientovanému, přístupu ke generovanému schématu.

7. Závěr

Doktorand si klade za cíl provést detailní rozbor podobných metod a některou z metod fuzzyfikovat, případně použít metodiku novou (vyplývající ze studie proveditelnosti). Zajímavá bude konfrontace výsledků těchto metod právě s ohledem na extrahovanou sémantiku dat.

Tato metoda by měla korespondovat s rámcem sémantického webu. Předpokládá se, že bude implementován celý nástroj na získávání informací z webových stránek, příp. z jiných, veřejně přístupných, internetových zdrojů. Teoretické aspekty práce pak budou součástí disertační práce doktoranda.

Míra rozpracovanosti tématu odpovídá době necelých 3 měsíců, po kterou se autor danou problematikou detailně zabývá. Autor se snaží zohledňovat především praktickou část problematiky, o čemž svědčí i částečná implementace nástroje na základě dílčích teoretických výsledků.

References

[1] M. Řimnáč, "Mapa webové sítě", *Diplomová práce* Katedra řídicí techniky, FEL, ČVUT. 2004.

[2] J.A. Hoffer, D.G. Severance, "The Use of Cluster Analysis in Physical Data Base Design", in *First International Conference on Very Large Data Bases*, pp. 69–86, 1975.

[3] B.N. Shamkant, R. Minyoung, "Vertical Partitioning for Database Design – A Graphical Algorithm", in *SigMod*, pp. 440–450, 1989.

[4] G. Ausiello, A. D'Atri, M. Moscarini, "Chordality Properties on Graphs and Minimal Conceptual Connections in Semantic Data Models", in *Symposium on Principles of Database Systems*, pp. 164–170. 1985.

[5] G. Grahme, K. Räihä, "Database Decomposition into Fourth Normal Form", in *Conference on Very Large Databases*, pp. 186–196, 1983.

[6] M.A. Melkanov, C. Zaniolo, "Decomposition of Relations and Synthesis of Enitity–Relationship Diagram", in *Entity-Relationship Approach Conceptual Modelling*, pp. 277–294, 1979.

[7] J.A. Hoffer, D.G. Severance, "A Sophisticate's Introduction to Database Normalisation Theory", in *Fourth International Conference on Very Large Data Bases*, pp. 113–124, 1978.

[8] J. Biskup, U. Dayal, P.A. Bernstein, "Synthesizing Independend Database Schemas", in *SigMod*, pp. 143–150, 1979.

[9] P.A. Bernstein, J.R. Swenson, D.C. Tsichristzis, "A Unified Approach to Functional Dependecies and Relations", in *SigMod*, pp. 237–245, 1975.

[10] D. Maier, D.S. Warren, "Specifying Connections for Universal Relation Scheme Database", in *SigMod*, pp. 1–7, 1979.

[11] D. Rosaci, G. Terracina, D. Ursino, "A Framework for Abstracting Data Sources Having Heterogennous Representation Formats", in *Data & Knowledge Engineering*, vol. 48, pp. 1–38. 2004.

[12] J. Ullman, "Principle of Database Systems", *Computer Sience Press*. 1980.

[13] M Řimnáč, "Osobní stránka", *http://www.cs.cas.cz/∼rimnacm/* . [online].