Automatizovaný návrh pravidel pro integraci dat a sémantický web*

Zdeňka Linková, Martin Řimnáč

Ústav informatiky AV ČR, Pod Vodárenskou věží 2, 182 07 Praha 8
{linkova,rimnacm}@cs.cas.cz

Abstrakt Článek se zabývá přístupem, jak se pokusit zautomatizovat mnohdy netriviální úlohu nalezení pravidel pro integraci dat. Předkládaný přístup automaticky generuje kandidáty pravidel včetně jejich ohodnocení pomocí nepřímé míry definující jejich prioritu. Priorita může následně být použita buďto návrhářem (člověkem) jako pomocný prvek pro přípravu návrhu, nebo při automatickém návrhu integračního procesu zahrnující pravidla s maximální prioritou. Studie v příspěvku se detailně věnuje dvěma základním typům pravidel, ekvivalenci a hierarchii, přičemž ohodnocení kandidátů je založeno na (strukturální) analýze aktivních domén atributů. V neposlední řadě příspěvek ukazuje možnost decentralizovaného přístupu k integraci dat, jenž je inspirován webovými technologiemi.

1 Motivace

Způsoby zpracování dat za sebou mají více než čtyřicetiletý vývoj a adekvátní výzkum. S jejich rostoucím množstvím se objevují stále nové problémy, které je potřeba efektivně řešit. Jedním z nich je i vyhledávání relevantních dat, které se dnes neomezuje jen na jeden konkrétní zdroj, ale bere v potaz hned několik různých datových zdrojů. V mnoha případech je vhodné, či dokonce nutné, tyto různé zdroje integrovat, tedy na jejich data vytvořit jeden souvislý pohled.

Mezi klasické přístupy řešení integrace dat patří používání (virtuálních nebo materializovaných) pohledů. Při nematerializovaném přístupu je klíčové stanovení integračních pravidel, tzv. *mapování*, které vyjadřuje vztahy mezi daty fyzicky uloženými v původních zdrojích a mezi poskytovaným integrovaným pohledem [1].

Nalézt takové mapování je však mnohdy netriviální úloha, která bývá ve většině přístupů řešena manuálně [2]. Takové řešení dnes nelze označit za efektivní.

* 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), projektem 1M0554 Ministersva školství, mládeže a tělovýchovy ČR "Pokročilé sanační technologie a procesy" a záměrem AV0Z10300504 "Computer Science for the Information Society: Models, Algorithms, Applications".

Z tohoto důvodu jsou hledány přístupy [3, 4], jak úlohu hledání mapování co nejvíce zautomatizovat. Výsledkem těchto přístupů je návrh kandidátů mapovacích pravidel; tedy zautomatizování hledání mapování není úplné, o konečném výběru opět rozhoduje návrhář (člověk).

Přístup k hledání kandidátů prezentovaný v tomto příspěvku využívá nepřímých fuzzy měr pro ohodnocení kvality navrhovaných kandidátů mapování. Tato ohodnocení mohou být využita k setřídění kandidátů tak, aby návrhář mohl pouze (postupně) označit pravidla odpovídající jeho interpretaci schémat integrovaných zdrojů. Pakliže nebudeme uvažovat možnost takového finálního zásahu, může být toto ohodnocení chápáno jako odhad důvěry (podpora) pro použití daného automaticky navrženého pravidla pro integraci a tento odhad následně použít pro vyjádření relevance příslušné části výsledku.

Při integraci dat však může docházet k dalším komplikacím. Ačkoli jsou původní zdroje (lokálně) konzistentní, může výsledek integrace nějakou nekonzistenci obsahovat. Jelikož nelze předpokládat možnost ovlivnění obsahu zdroje, se kterým integrační systém pracuje, je nutné tuto situaci řešit jiným způsobem v rámci integrace. I v tomto případě lze využít nepřímých měr, například ve smyslu penalizace zdroje nebo k oslabení integračního pravidla, které má nekonzistenci za následek. Konkrétní řešení pak závisí na vyhodnocení dané situace, neboť příčin nekonzistence může být obecně více.

Článek nejprve v sekci 2 definuje formalismus pro ukládání dat, jenž je inspirován myšlenkami sémantického webu, a uvádí příklad dotazování. Sekce 3 uvádí různé přístupy k integaci dat prostřednictvím zavedeného formalismu a navrhuje vedle klasického (databázového) centralizovaného řešení variatu dencetralizovanou, lépe odpovídající webovému prostředí. Dále jsou v sekci 4 obecně jmenovány přístupy používané pro automatický návrh integračních pravidel a detailně, včetně příkladu popisujícího konkrétní aplikaci - zjišťování ekvivalence a hierarchie mezi atributy na základě analýzy jejich aktivních domén.

2 Model úložiště dat

Data jsou poskytována zdroji $z \in \mathscr{Z}$. Předpokládejme, že známe schéma (nebo jiný ekvivalentní popis) každého zdroje $S_z = (\mathscr{A}_z, \mathscr{F}_z)$ pokrývající alespoň seznam atributů $\mathscr{A}$ a funkčních závislostí $\mathscr{F}_z \subseteq \mathscr{A}_z \times \mathscr{A}_z$ mezi (jednoduchými) atributy. Předpokládejme, že data jsou obecně [5, 6] reprezentována pomocí elementů $e \in \mathscr{E}$ - přípustných dvojic atribut - hodnota $\mathscr{E}_z \subseteq \mathscr{A}_z \times \mathscr{D}_z$, kde obor hodnot zdroje $\mathscr{D}_z$ představuje všechny hodnoty atributů pokryté zdrojem, t.j. $\mathscr{D}_z = \bigcup_{\forall A \in \mathscr{A}_z} \mathscr{D}^z_\alpha(A)$. Symbol $\mathscr{D}^z_\alpha(A) \subseteq \mathscr{D}(A)$ představuje aktivní doménu atributu A, jenž zahrnuje pouze ty hodnoty z domény atributu $\mathscr{D}(A)$ dané schématem S_z, které jsou pokryty zdrojem z.

Za těchto předpokladů je možné reprezentovat data zdroje jako instance funkčních závislostí $f \in \mathscr{F}_z$ pomocí implikací $e_i \rightarrow e_j$ mezi elementy. Pokud existuje index elementů $\mathscr{I}_\mathscr{E} : \mathscr{E} \rightarrow \mathbb{N}$, můžeme tyto implikace pro každý zdroj

$z_l \in \mathscr{Z}$ vyjádřit pomocí čtvercové binární matice úložiště Φ_l definované jako

$$\Phi_l = [\phi^l_{ij}];\ \phi^l_{ij} = \begin{cases} 1 \text{ pokud zdroj } z_l \text{ pokrývá implikaci } e_i \to e_j \\ 0 \text{ jinak} \end{cases} \tag{1}$$

Analogicky lze vyjádřit matici aktivních domén atributů Δ_l zdroje z_l jako

$$\Delta_l = [\delta^l_{ij}];\ \delta^l_{ij} = \begin{cases} 1 \text{ pokud } \exists v : e_i = (A_j, v) \in \mathscr{E}_l \\ 0 \text{ jinak} \end{cases} \tag{2}$$

Poznamenejme, že každý nenulový prvek matice úložiště $\phi^l_{ij} > 0$ je instancí nějaké funkční závislosti $f = (A_{i'} \to A_{j'}) \in \mathscr{F}_l$. Díky tomuto faktu je možné definovat matici funkčních závislostí Ω_l jako[1]

$$\Omega_l = [\omega^l_{ij}] = (\Delta_l \Phi_l \Delta_l^T) \gtrdot 0 \tag{3}$$

Požadujme, aby matice úložiště Φ_l byla konzistentní, tj. pokrývala pouze instance funkčních závislostí, tedy[2] $\omega^l_{ij} = 1$

$$\Phi_l = \Phi'_l \odot (\Delta_l^T \Omega_l \Delta_l) \tag{4}$$

Vztahy (3) a (4) představují axiomy pro konzistetní úložiště dat relační povahy.

2.1 Způsoby dotazování

Na data uložená pomocí binární matice Φ_l se lze dotazovat dvěma způsoby [5]:

- Generalizace - odpovídá na dotaz, které elementy jsou implikovány z elementů aktivovaným vektorem dotazu $\boldsymbol{x}_k$:

$$\boldsymbol{x}^G_{k+1} = \Phi_l \boldsymbol{x}_k \tag{5}$$

- Specializace - odpovídá restrikčnímu dotazu - vrací elementy, ze kterých je možné elementy aktivované vektorem $\boldsymbol{x}_k$ odvodit.

$$\boldsymbol{x}^S_{k+1} = \Phi_l^T \boldsymbol{x}_k \tag{6}$$

V dalším textu předpokládejme, že matice úložiště Φ_l je konzistentní, představuje monotónní odvozovací proces (tj. 1 na diagonále) a její prvky splňují podmínku transitivity. Za těchto podmínek je výsledek dosažitelný v nejvýše $n = |\mathscr{A}_l|$ krocích. Z tohoto důvodu je nutné aplikovat generalizační, resp. specializační operátor tolikrát, dokud dochází ve vektoru $\boldsymbol{x}_k$ k aktivaci nových elementů (tj. do okamžiku, kdy $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$.)

Maticový zápis s binární maticí úložiště [5] je možné přepsat do obecnější formy umožňující vážit implikace mezi elementy [7]. Tento přístup umožňuje použití hodnot z celého intervalu $\phi_{ij} \in < 0, 1 >$. Odvozovací mechanismus, se stejným chováním v krajních mezích, pak bude definován jako zobecnění:

$$x_{k+1}(i) = \sum_{\forall j} \phi_{ij} x_k(j) \ \rightsquigarrow\ x_{k+1}(i) = \max_{\forall j} \{\phi_{ij} x_k(j)\} \tag{7}$$

[1] Operátor $\gtrdot$ je představuje porovnání každého prvku matice se skalárem.
[2] Operátor $\odot$ je představuje násobení matic prvek po prvku.

2.2 Příklad dotazování

Mějme zdroj z_1, jehož struktura i data jsou popsána následně:

$$\Phi_1 = \left[\begin{array}{ccc|cc|cc} 1&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0 \\ 0&0&1&0&0&0&0 \\ \hline 1&1&0&1&0&0&0 \\ 0&0&1&0&1&0&0 \\ \hline 0&0&0&1&0&1&0 \\ 0&0&0&0&1&0&1 \end{array}\right] \begin{array}{l} \text{město, Praha} \\ \text{město, Brno} \\ \text{město, Vídeň} \\ \hline \text{země, ČR} \\ \text{země, Rakousko} \\ \hline \text{měna, CZK} \\ \text{měna, EUR} \end{array}$$

$$\Delta_1 = \left[\begin{array}{ccc|cc|cc} 1&1&1&0&0&0&0 \\ \hline 0&0&0&1&1&0&0 \\ \hline 0&0&0&0&0&1&1 \end{array}\right] \qquad \Omega_1 = \left[\begin{array}{c|c|c} 1&0&0 \\ \hline 1&1&0 \\ \hline 0&1&1 \end{array}\right] \begin{array}{l} \text{město} \\ \text{země} \\ \text{měna} \end{array}$$

Uvažujme, že matice úložiště je navržena jako binární. Budeme-li se dotazovat na všechny dostupné informace ohledně města *Praha*, bude vektor aktivovaných entit obsahovat na první pozici 1 odpovídající $[z_1 : \text{město} = \text{Praha}\ ]$, ostatní pozice budou nulové. Vynásobení maticí úložiště aktivuje navíc pozice odpovídající entitám $[z_1 : \text{země} = \text{ČR}\ ]$ a v dalším kroku pak $[z_1 : \text{měna} = \text{CZK}\ ]$.

$$\begin{aligned} \boldsymbol{x}_0 &= [1\,0\,0\,0\,0\,0\,0] \\ \Phi_1\boldsymbol{x}_0 = \boldsymbol{x}_1 &= [1\,0\,1\,0\,0\,0\,0] \\ \Phi_1\Phi_1\boldsymbol{x}_0 = \boldsymbol{x}_2 &= [1\,0\,1\,0\,1\,0\,0] \\ \Phi_1\Phi_1\Phi_1\boldsymbol{x}_0 = \boldsymbol{x}_3 &= [1\,0\,1\,0\,1\,0\,0] \end{aligned} \tag{8}$$

Neboť nedochází k žádným dalším aktivacím, t.j. $\boldsymbol{x}_2 = \boldsymbol{x}_3$, výsledek je finální.

Poznamenejme, že výhodou navrženého formalismu je jeho přímočaré propojení na formáty dokumentů sémantického webu přes trojice (X definuje zdrojovou matici ve smyslu typ vztahu):

$$(i, \mathrm{X}, j) \in \mathscr{B} \Leftrightarrow x_{ij} = 1$$

Fragment RDF dokumentu (o elementu $[z_1 : \text{město} = \text{Praha}\ ]$) může být zapsán

```
<element rdf:about="element-město-Praha">
   <of-attribute rdf:resource="#attribute-město"/>
   <of-term rdf:resource="#term-Praha"/>
   <implies-element rdf:resource="#element-země-ČR"/>
</element>
<attribute rdf:about="attribute-město">
   <rdfs:label xml:lang="cs">Město</rdfs:label>
   <implies-attribute rdf:resource="#attribute-země"/>
</attribute>
<term rdf:about="term-Praha">
   <rdfs:label xml:lang="cs">Praha</rdfs:label>
   <rdfs:label xml:lang="en">Prague</rdfs:label>
</term>
```

3 Integrace dat

Datová integrace patří mezi dlouhodobě řešené problémy. Různé přístupy [8–10] se zabývají integrací dat na různých úrovních abstrakce. Liší se i ve formě, v jaké poskytují ucelený pohled na data uložená ve více zdrojích. S ohledem na množství dat (například webové zdroje) a jejich častou aktualizaci, je k integraci stále častěji využíván nematerializovaný přístup [11]. Ten spočívá v definici *globálního virtuálního pohledu* nad integrovanými zdroji. Protože data fyzicky zůstávají uložena ve zdrojích původních, je nutné zajistit vazbu mezi nimi a globálním pohledem. K tomu slouží *mapování - integrační pravidla*, která zachycují vazby mezi lokálními schématy zdrojů a globálním integrovaným schématem a která jsou pak při zpracování dat (například při dotazování) využita.

Klasické přístupy při nematerializované integraci se obecně rozdělují na GAV a LAV [12–14]:

- *GAV (Global As View)* přístup je založen na definici globálního virtuálního pohledu pomocí pohledů nad lokálními zdroji. Každý prvek globálního schématu je tedy charakterizován jako pohled nad lokálními schématy.
- Naopak *LAV (Local As View)* spočívá v definici lokálních schémat jako pohledů definovaných nad globálním schématem. V tomto přístupu je globální schéma voleno (relativně) nezávisle na schématech zdroje. Každý zdroj je potom charakterizován v termínech globálního schématu.

3.1 Základní integrační pravidla

K popisu mapování, ať už získaného přístupem GAV nebo LAV, je možné využít různých struktur. Může se jednat o $1-1$ *mapovací pravidla*, která vyjadřují vztah mezi dvojicí prvků mapovaných schémat, či o složitější struktury, vyjadřující komplexnější vztahy či zahrnující více prvků schémat.

Pro potřeby příspěvku se omezme pouze na $1-1$ mapovací pravidla. Ta mohou být použita k vyjádření

- hierarchického vztahu $A_i \sqsubset A_j$ mezi atributy schémat A_i a A_j
- ekvivalence $A_i \sim A_j \Leftrightarrow A_i \sqsubset A_j \wedge A_j \sqsubset A_i$

Integrační pravidla uvedená výše mohou být pro zdroje z_k a z_l popsána pomocí matice Π_{kl} definované jako

$$\Pi_{kl} = [\pi_{ij}^{kl}] : \pi_{ij}^{kl} = \begin{cases} 1 & \text{pokud } A_i \sqsubset A_j, \forall A_i \in \mathscr{A}_l, \forall A_j \in \mathscr{A}_k \\ 0 & \text{jinak} \end{cases} \tag{9}$$

Tato pravidla se projeví na úrovni elementů, obecně

$$\Psi_{kl} = [\psi_{ij}^{kl}] : \psi_{ij}^{kl} = \begin{cases} 1 & \text{pokud } e_i \sqsubset e_j, \forall e_i \in \mathscr{E}_l, \forall e_j \in \mathscr{E}_k \\ 0 & \text{jinak} \end{cases} \tag{10}$$

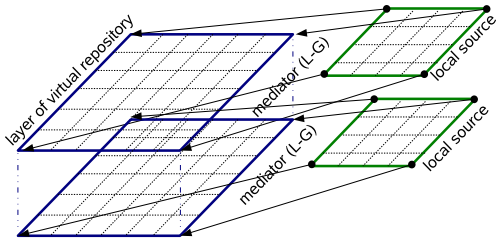
3.2 Virtuální globální úložiště

Pomocí integračních pravidel můžeme celou množinu zdrojů $\mathscr{Z}$ „složit“ v jeden virtuální zdroj. Prvním řešením je použití centralizovaného přístupu, který je znám z klasické integrace relačních dat. Toto řešení spočívá v zavedení mapování mezi elementy každého lokálního zdroje a elementy definovaném na globální úrovni (v globálním schématu). Takové mapování může být realizováno pomocí matice Γ_l definované jako:

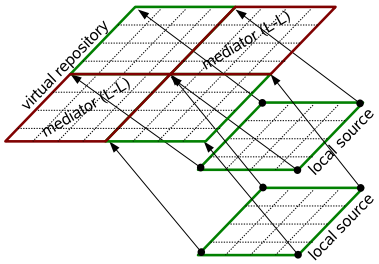
$$\Gamma_l = [\gamma_{ij}^l] : \gamma_{ij}^l = \begin{cases} 1 & \text{pokud } e_i \sim e_j, \forall e_i \in \mathscr{E}_l, \forall e_j \in \mathscr{E}_{\mathscr{Z}} \\ 0 & \text{jinak} \end{cases} \tag{11}$$

Virtuální úložiště pak bude reprezentováno pomocí součtu transformovaných matic úložiště lokálních zdrojů.

$$\Phi_{\mathscr{Z}} = \sum_{\forall z_l \in \mathscr{Z}} \Gamma_l \Phi_l \Gamma_l^T \tag{12}$$



Obrázek 1. Centralizované řešení



Obrázek 2. Decentralizované řešení

Alternativně v souladu s webovým prostředím, mapování může být zavedeno přímo mezi dvěma zdroji. Pak

$$\Psi_{kl} = \Gamma_l^T \Gamma_k \tag{13}$$

a virtuální úložiště lze složit blokově pomocí

$$\Phi = \begin{bmatrix} \Phi_1 & \Psi(\Pi_{12}) & \cdots & \Psi_{1|\mathscr{Z}|} \\ \Psi_{21} & \Phi_2 & \cdots & \Psi_{2|\mathscr{Z}|}) \\ \vdots & & \ddots & \\ \Psi_{|\mathscr{Z}|1} & \cdots & \cdots & \Phi_{|\mathscr{Z}|} \end{bmatrix} \tag{14}$$

Výhodou tohoto přístupu je možnost zvolit zdroj, jehož se primárně budeme dotazovat a tak určit preference výsledků.

4 Metody hledání kandidátů integračních pravidel

Při hledání korespondencí mezi schématy je možné využít různé úrovně informací, které jsou k dispozici. Porovnávání jednotlivých prvků může být založeno na jejich názvech (přičemž může být využito lexikálních technik, dalších informací o vztazích mezi pojmy, například synonyma apod.), na jejich datových typech, aktivních doménách, či jejich struktuře. Využitím těchto informací je možné určit, které elementy schémat spolu pravděpodobně souvisí, případně i druh vztahu. V mnohých řešených projektech je pak po této fázi nutná interakce s lidským uživatelem, který rozhodne, zda se nalezená korespondence mezi elementy skutečně vyskytuje.

Předpokládejme, že při integraci dvou či více zdrojů jsou k dispozici OWL ontologie popisující strukturu zdroje (pomocí ontologických tříd a jejich vlastností) a data (jako instance definovaných tříd). Při hledání korespondence mezi jednotlivými třídami lze využít [3]:

- *lexikální analýzu.* Porovnávány mohou být všechny pojmy použité k popisu tříd, především pak její název, ale například i názvy jejích vlastností apod. Tyto pojmy mohou být zkoumány jak ze syntaktického hlediska (např. úplná shoda dvou znakových řetězců, jedno slovo je prefixem/sufixem druhého, slova mají stejný kořen apod.), tak z hlediska sémantického (např. slova jsou synonyma, nebo hyponymum a hypernymum).
- *porovnávání na úrovni instancí.* Při určování, zda spolu dané třídy souvisí, je možné také porovnávat jejich extenze, tedy instance (individua, členy dané třídy). A to především ve smyslu, zda je instance jedné třídy zároveň instancí druhé třídy. Je možné uvažovat situace jako například, zda je jedna množina instancí podmnožinou jiné, či zkoumat průnik obou množin.
- *strukturální analýzu.* Třídy mohou být porovnávány z hlediska jejich struktury - kolik vlastností a jaké mají porovnávané třídy definovány, jakého typu jednotlivé vlastnosti jsou, zda mají třídy společného předka v hierarchii tříd a podobně.

Po nalezení možných korespondencí a určení jejich ohodnocení je pak možné dále usuzovat o tom, které korespondence skutečně jako integrační pravidla definovat. V obvyklém případě, kdy z kandidátů vybírá sám uživatel, je možné mu tuto úlohu díky stanovenému ohodnocení usnadnit. Kandidáty lze setřídit od těch nejvíce pravděpodobných, takže se uživatel nejprve zabývá těmi nejrelevantnějšími a může kdykoliv v průběhu úlohu ukončit s úmyslem, že ostatní kandidáti nebudou vybrány, aniž by se jimi musel zabývat. Ohodnocení také může uživateli sloužit k podání informace o tom, jak moc vysoce relevantní jsou zkoumané prvky schémat viděny.

Není-li z jakéhokoliv důvodu možné, aby lidský uživatel z kandidátů sám zvolil, je nutné celý proces dokončit automaticky. Z navržených korespondencí jsou pak vybrány takové, jejichž míra ohodnocení je maximální, jednoznačně odpovídá kandidátovi a zároveň překročila zadanou prahovou hodnotu. Takové jsou pak považovány za odvozená integrační pravidla; možné korespondence s ohodnocením nižším nejsou dále uvažovány, neboť jsou nahlíženy jako neplatné.

4.1 Automatický návrh decentralizovaných pravidel

Pro potřeby tohoto článku se omezme na triviální lexikální analýzu. Ta může být založena na faktu, že elementy shodných hodnot v označíme za ekvivalentní. Tedy

$$\Psi'_{kl} = [\psi^{kl}_{ij}] : \psi^{kl}_{ij} = \begin{cases} 1 & \text{pokud } e_i = (A_{i'}, v) \in \mathscr{E}_l \wedge e_j = (A_{j'}, v) \in \mathscr{E}_k \\ 0 & \text{jinak} \end{cases} \tag{15}$$

Použití takového mapování není v praxi vhodné, avšak lze z něj vyjádřit překryv domén atributů různých zdrojů pomocí:

$$\Pi'_{kl} = [\theta^{kl}_{ij}] = \Delta^T_l \Psi'_{kl} \Delta^T_k \tag{16}$$

4.2 Integrační pravidlo ekvivalence atributů a jeho ohodnocení

Jako důsledek strukturálních vazeb je možné usuzovat, že ekvivalentní atributy budou mít podobné (aktivní) domény. Za tohoto předpokladu definujeme míru μ^{kl}_{ij} ekvivalence atributů $A_i \in \mathscr{A}_k, A_j \in \mathscr{A}_l$ pomocí

$$\hat{\Pi}_{kl} = [\mu^{kl}_{ij}];\ \mu^{kl}_{ij} = \mu^{lk}_{ji} = \frac{|\mathscr{D}^k_\alpha(A_i) \cap \mathscr{D}^l_\alpha(A_j)|}{|\mathscr{D}^k_\alpha(A_i) \cup \mathscr{D}^l_\alpha(A_j)\ |} = \frac{\theta^{kl}_{ij}}{|\mathscr{D}^k_\alpha(A_i) \cup \mathscr{D}^l_\alpha(A_j)\ |} \tag{17}$$

Tato míra μ^{kl}_{ij} představuje preferenci kandidáta integračního pravidla, na základě které lze kandidáty uspořádat a návrhář pak může postupně procházet kandidáty (od maximálního do minimálního překryvu domén) a následně rozhodnout. Pakliže zásah návrháře není (principiálně) možný, vybere se pravidlo s jednoznačně nejvyšší preferencí, tj.

$$\Pi_{kl} = [\pi^{kl}_{ij}]; \pi^{kl}_{ij} = \begin{cases} \mu^{kl}_{ij} & \text{pokud } j = \arg\max_{\forall j'} \mu^{kl}_{ij'} \\ 0 & \text{jinak} \end{cases} \tag{18}$$

Následně jsou ponechány pouze ekvivalence elementů odpovídající zvoleným Π_{kl} ekvivalencím mezi atributy:

$$\Psi_{kl} = \Psi'_{kl} \odot \Delta^T_l \Pi_{kl} \Delta_k \tag{19}$$

4.3 Příklad automatického návrhu pravidel ekvivalence

Mějme zdroj z_1 z předchozího příkladu a přidejme zdroj z_2 popsaný pomocí:

$$\Phi_2 = \left[\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{array}\right] \quad \begin{array}{l} z_2\text{:stát} = \text{ČSFR} \\ z_2\text{:stát} = \text{Rakousko} \\ z_2\text{:hlavní\_město} = \text{Praha} \\ z_2\text{:hlavní\_město} = \text{Vídeň} \end{array} \tag{20}$$

Na základě strukturální analýzy normované podle (17) získáme

$$\hat{\Pi}_{12} = \hat{\Pi}^T_{21} = \begin{bmatrix} 0 & \frac{2}{3} \\ \frac{1}{3} & 0 \\ 0 & 0 \end{bmatrix} \tag{21}$$

Budeme-li se nyní zdroje z_1 dotazovat na všechny dostupné informace ohledně $[z_1 : \text{město} = \text{Praha }]$, postupně získáme

$$\begin{aligned} \boldsymbol{x}_0 &= [1\ 0\ 0\ 0\ 0\ 0\ 0 | 0\ 0\ 0\ 0] \\ \Phi\boldsymbol{x}_0 = \boldsymbol{x}_1 &= [1\ 0\ 0\ 1\ 0\ 0\ 0 | 1\ 0\ \tfrac{2}{3}\ 0] \\ \Phi\Phi\boldsymbol{x}_0 = \boldsymbol{x}_2 &= [1\ 0\ 0\ 1\ 0\ 1\ 0 | \tfrac{2}{3}\ 0\ \tfrac{2}{3}\ 0] \\ \Phi\Phi\Phi\boldsymbol{x}_0 = \boldsymbol{x}_2 &= [1\ 0\ 0\ 1\ 0\ 1\ 0 | \tfrac{2}{3}\ 0\ \tfrac{2}{3}\ 0] \end{aligned} \tag{22}$$

Tento výsledek lze interpretovat jako

z_1:město, z_2:hlavní_město	$\frac{2}{3}$	Praha	$1 + \frac{2}{3} = 1.66$
z_1:stát, z_2:země	$\frac{1}{3}$	ČR	1
	$\frac{1}{3}$	ČSFR	$\frac{2}{3}$
z_1:měna	1	CZK	1

(23)

Je patrné, že zdroje se neshodnou na hodnotě atributu $z_1 : \text{stát} \sim z_1 : \text{země}$. Tato nekonzistence je ve výsledku zobrazena včetně preferencí hodnot a ohodnocení jistoty pravidla a finální interpretace je ponechána na koncovém uživateli. Poznamenejme, že nekonzistence lze rovněž vážit; platí, že nekonzistence u integračního pravidla s dobrou podporou a malý rozdílem aktivací nekonzistentních elementů je pro výslednou interpretaci výsledku více „nebezpečná“. Na příkladu je dobře patrná výhoda možnosti zvolit zdroj a fakt, že váhy integračních pravidel oslabují aktivaci elementů z jiných zdrojů.

4.4 Integrační pravidlo hierarchie atributů a jeho ohodnocení

Nepřímou míru pro hierarchii atributů A_i ze zdroje z_k a A_j ze zdroje z_l můžeme získat analogicky, avšak je potřeba rozhodnout o tom, která z následujících možností nastala:

1. ekvivalence $A_i \sim A_j = A_i \sqsubset A_j \wedge A_j \sqsubset A_i$
2. nadřazenost $A_i \sqsubset A_j$
3. nadřazenost $A_j \sqsubset A_i$
4. žádný vztah

Jako v případě ekvivalence vyjedeme z předpokladu, že atributy jsou si tak nadřazené, jak

$$\nu_{ij}^{kl} = \frac{|\mathscr{D}_\alpha^k(A_i) \cap \mathscr{D}_\alpha^l(A_j)|}{|\mathscr{D}_\alpha^k(A_i)|} = \frac{\theta_{ij}^{kl}}{|\mathscr{D}_\alpha^k(A_i)|} \tag{24}$$

Aby ohodnocení přiřazení vztahu do kategorie bylo porovnatelné, zavedeme

$$\sigma_{ij}^{\sim} = \nu_{ij}^{kl} \cdot \nu_{ji}^{lk} \tag{25}$$

$$\sigma_{ij}^{\sqsubset} = \nu_{ij}^{kl} \cdot (1 - \nu_{ji}^{lk}) \tag{26}$$

Výběr pravidel podle ohodnocení provedeme analogicky podle (18), avšak díky faktu, že hierarchie je (na rozdíl od ekvivalence) nesymetrická, je nutné zajistit maximum jak v řádku, tak ve sloupci. Navíc je nutné zajistit, aby nedošlo

k situaci, kdy domény atributů se překrývají a navíc platí $\pi_{ki} \geq \pi_{kj}$ a zároveň $\pi_{jk} \geq \pi_{ik}$. Jinými slovy data vedou na návrh pravidel $A_j \sqsubset A_k$ a $A_k \sqsubset A_i$. V tomto případě, pakliže bychom zvolili tuto konfiguraci pravidel, je principiálně možné odvodit element (A_j, v) na základě aktivace elementu (A_i, v). Taková aktivace však nemusí odpovídat instanci funkční závislosti (která navíc nemusí vůbec existovat). Takováto integrační pravidla, vedoucí na instance funkčních závislostí, nemohou být použita. Poznamenejme, že situace je zapříčiněna

$$\frac{||\mathscr{D}_\alpha(A_i)||}{||\mathscr{D}_\alpha(A_j)||} \geq \frac{||\mathscr{D}_\alpha(A_i) \cap \mathscr{D}_\alpha(A_k)||}{||\mathscr{D}_\alpha(A_j) \cap \mathscr{D}_\alpha(A_k)||} \geq 1 \tag{27}$$

4.5 Příklad ohodnocení pravidel

Ukažme si nyní ohodnocení pravidel na příkladu. Použijme stejné zdroje z_1 a z_2 jako v předchozím příkladu. Strukturální analýzou získáme

$$\Sigma_{12}^{\sqsubset} = [\sigma_{ij}^{12}] = \begin{bmatrix} 0 & \frac{1}{2} \\ \frac{2}{2} & 0 \\ 0 & 0 \end{bmatrix} \qquad \Sigma_{21}^{\sqsubset} = [\sigma_{ji}^{21}] = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{3} & 0 & 0 \end{bmatrix} \tag{28}$$

Podle přepočtu získáme ohodnocení pro ekvivalenci a hierarchii:

$$\Sigma_{12}^{\sqsubset} = \begin{bmatrix} 0 & \frac{1}{2} \cdot \frac{1}{2} \\ \frac{2}{2} \cdot \frac{2}{3} & 0 \\ 0 & 0 \end{bmatrix} \tag{29}$$

$$\Sigma_{21}^{\sqsubset} = \begin{bmatrix} 0 & \frac{1}{2} \cdot \frac{1}{2} & 0 \\ \frac{1}{3} \cdot \frac{1}{3} & 0 & 0 \end{bmatrix} \tag{30}$$

$$\Sigma_{21}^{\sim} = \begin{bmatrix} 0 & \frac{1}{2} \cdot \frac{1}{2} & 0 \\ \frac{1}{3} \cdot \frac{2}{3} & 0 & 0 \end{bmatrix} \tag{31}$$

Nyní seřadíme pravidla podle relevance a (při přednosti ekvivalence) získáváme:

$$\begin{array}{l|l|l} z_2 : \text{hlavní\_město} \sqsubset z_1 : \text{město} & \frac{2}{3} & \oplus \\ z_2 : \text{hlavní\_město} \sim z_1 : \text{město} & \frac{2}{9} & \ominus \\ z_2 : \text{stát} \sim z_1 : \text{země} & \frac{1}{4} & \oplus \\ z_2 : \text{stát} \sqsubset z_1 : \text{země} & \frac{1}{4} & \ominus \\ z_1 : \text{země} \sqsubset z_2 : \text{stát} & \frac{1}{4} & \ominus \\ z_1 : \text{město} \sqsubset z_2 : \text{hlavní\_město} & \frac{1}{9} & \ominus \end{array} \tag{32}$$

Na základě tohoto rozboru stanovíme

$$\Pi_{12} = \begin{bmatrix} 0 & \frac{1}{2} \\ \frac{2}{3} & 0 \\ 0 & 0 \end{bmatrix} \qquad \Pi_{21} = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{33}$$

Budeme-li se nyní dotazovat zdroje z_2 na informace o Praze, dostaneme

$$\begin{aligned} \boldsymbol{x}_0 &= [0\ 0\ 0\ 0\ 0\ 0\ 0 | 0\ 0\ 1\ 0] \\ \Phi \cdot \boldsymbol{x}_0 = \boldsymbol{x}_1 &= [\tfrac{2}{3}\ 0\ 0\ 0\ 0\ 0\ 0 | 1\ 0\ 1\ 0] \\ \Phi \cdot \Phi \cdot \boldsymbol{x}_0 = \boldsymbol{x}_2 &= [\tfrac{2}{3}\ 0\ 0\ \tfrac{2}{3}\ 0\ 0\ 0 | 1\ 0\ 1\ 0] \\ \Phi \cdot \Phi \cdot \Phi \cdot \boldsymbol{x}_0 = \boldsymbol{x}_2 &= [\tfrac{2}{3}\ 0\ 0\ \tfrac{2}{3}\ 0\ \tfrac{2}{3}\ 0 | 1\ 0\ 1\ 0] \\ \Phi \cdot \Phi \cdot \Phi \cdot \Phi \cdot \boldsymbol{x}_0 = \boldsymbol{x}_2 &= [\tfrac{2}{3}\ 0\ 0\ \tfrac{2}{3}\ 0\ \tfrac{2}{3}\ 0 | 1\ 0\ 1\ 0] \end{aligned} \tag{34}$$

což můžeme interpretovat jako

z_1:město $\sqsupset$ z_2:hlavní_město		Praha	$1 + 0.66 = 1.66$
z_1:stát, z_2:země	$\frac{1}{4}$	ČSFR	1
	$\frac{1}{4}$	ČR	0.66
z_1:měna	1	CZK	0.66

(35)

Povšimněte si, že kdybychom se dotazovali na data zdroje z_1, díky orientaci hierarchického pravidla z_1 : město $\sqsubset$ z_2 : hlavní_město by nedošlo k aktivaci obou elementů odpovídající Praze; jinými slovy by výsledek pokrýval pouze data ze zdroje z_1.

V případě, že by zdroj z_2 pokrýval element [stát : ČR] namísto [stát : ČSFR], došlo by k aplikaci odpovídajícího pravidla ekvivalence a díky funkční závislosti i k aktivaci elementu $[z_2$: hlavní_město, Praha]. Jednoduše je možné ukázat, že oba atributy z_1 : město a z_2 : hlavní_město budou ve výsledku vystupovat odděleně (v interpretaci, že každému městu ve státě přísluší právě jedno město):

z_1:město		Praha	1
z_1:stát, z_1:země	1	ČR	$1 + 1 = 2$
z_1:měna	1	CZK	1
z_2:hlavní_město	1	Praha	1

(36)

5 Závěr

Příspěvek je orientován na problematikou automatického návrhu integračních pravidel a ukazuje použití metod na jednoduchém příkladě. Známé metody poskytnou na základě různých mechanismů seznam možných kandidátů na integrační pravidla. Vzhledem k tomu, že integračních pravidel bývá reálně mnoho, množina všech možných kandidátů bude o to více početná. Příspěvek proto navrhuje vážit pravidla pomocí nepřímých měr vycházející ze (strukturální) analýzy dat jednotlivých lokálních zdrojů a tyto váhy následně použít pro vyjádření priority pravidel; návrhář tím získá seřazený seznam kandidátů, z nichž podle své interpretace lokálních schémat postupně vybere ty, které považuje za platné.

V případě, že není možné počítat se zásahem návrháře do výběru pravidel, mohou tyto váhy sloužit jako nejlepší možný odhad podpory pro existenci takového pravidla. Na základě tohoto odhadu jsou pravidla s maximální váhou označena jako platná. Velmi dobrých výsledků, jak ukazují příklady, je dosaženo za podmínky, kdy (globální) domény atributů jsou navzájem disjunktní; v ostatních případech metoda vede na váhy ze středu intervalu a jsou možná víceznačná rozhodnutí.

V neposlední řadě se příspěvek zabývá alternativou ke (klasickému) centralizovaném přístupu k integraci dat. Díky orientaci tématu na webové technologie, příspěvek zavádí možnost decentralizovaného přístupu k integraci, kdy zdroj vedle svých vlastních dat může poskytovat i odkazy na data jiných zdrojů. Váhy integračních pravidel pak mohou sloužit i jako ochrana dat zdroje před zavlečenými chybami (nekonzistencemi) způsobených zahrnutím dat ostatních zdrojů

do výsledku. Tento přístup umožní na základě dotazu na jeden zdroj získat kompletní informaci pokrývající všechny, odkazy navzájem propojené, zdroje, přičemž dotazovaný zdroj garantuje správnost vráceného výsledku.

Reference

1. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. **3730** (2005) 146–171
2. Mitra, P., Wiederhold, G., Jannink, J.: Semi-automatic integration of knowledge sources. In: Proc. of the 2nd Int. Conf. On Information FUSION'99. (1999)
3. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB Journal: Very Large Data Bases **10**(4) (2001) 334–350
4. Yi, S., Huang, B., Chan, W.T.: Xml application schema matching using similarity measure and relaxation labeling. Inf. Sci. **169**(1-2) (2005) 27–46
5. Řimnáč, M.: Data structure estimation for rdf oriented repository building. In: Proceedings of the CISIS 2007, Los Alamitos, CA, USA, IEEE Computer Society (2007) 147–154
6. Bednárek, D., Obdržálek, D., Yaghob, J., Zavoral, F.: Access rights definition and management in an information system based on a datapile structure. ITAT 2004, Workshop on Information Technologies, Application and Theory (2004)
7. Řimnáč, M., Špánek, R., Linková, Z.: Semantic web: Vision of distributed and trusted data environment? In: Proceedings of WWM 2007, 1st International Web X.0 and Web Mining Workshop, IEEE (2007) 627–634
8. Do, H.H., Rahm, E.: Matching large schemas: Approaches and evaluation. Inf. Syst. **32**(6) (2007) 857–885
9. Nottelmann, H., Straccia, U.: Information retrieval and machine learning for probabilistic schema matching. Information Processing and Management **43** (2006) 552–576
10. Xu, L., Embley, D.W.: A composite approach to automating direct and indirect schema mappings. Inf. Syst. **31**(8) (2006) 697–732
11. Ullman, J.D.: Information integration using logical views. Theoretical Computer Science **239**(2) (2000) 189–210
12. Lenzerini, M.: Data integration: a theoretical perspective. In: PODS '02: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of Database Systems, New York, NY, USA, ACM Press (2002) 233–246
13. Calí, A., Calvanese, D., Giacomo, G.D., Lenzerini, M.: On the expressive power of data integration systems. In: ER '02: Proceedings of the 21st International Conference on Conceptual Modeling, London, UK, Springer-Verlag (2002) 338–350
14. Linková, Z.: Mapování schémat v prostředí sémantického webu. Doktorandské dny na KM FJFI 07 (2007) 117–126

Annotation:

On Semiautomatic Design of Data Integration Rules and Semantic Web

Methods aimed at automatizing the task of finding rules for data integration are presented. The proposed methods generate candidates of integration rules together with a (cosine) measure expressing uncertainty of the rule. This measure can be used for sorting the candidates for a (human) designer or for considering priority of the automatic rule choice. Methods observing attribute equivalence and attribute hierarchy are discussed and their result is shown on the example.