Resolução de Conflitos em Documentos XML

Frantchesco Cecchin[1]
Orientadora: Carmem Satie Hara

PPGInf - Programa de Pós-Graduação em Informática
Departamento de Informática – Universidade Federal do Paraná
Caixa Postal 19.081 – 81.531-990 – Curitiba – PR – Brasil

{frantchesco, carmem}@inf.ufpr.br

Nível: Mestrado
Ingresso no Programa: março/2008
Previsão de Conclusão: março/2010
Etapas Concluídas: defesa de proposta

Resumo. *Melhorar a qualidade dos dados provenientes de diferentes fontes é uma tarefa complexa. Os dados importados destas fontes podem estar "sujos" e estruturados de formas distintas. Além disso, dados que se referem ao mesmo objeto do mundo real podem ser representados múltiplas vezes, causando duplicidade e inconsistência. Quando pretende-se manter essas informações de forma integrada, um aspecto importante para garantir a qualidade dos dados é a utilização de um processo eficiente de limpeza dos dados. Este trabalho propõe um conjunto de métodos e estratégias para a resolução dos conflitos identificados entre instâncias durante o processo de integração. Além disso, é proposta a utilização de um registro contendo o histórico de resoluções com a finalidade de resolver unicamente um mesmo conflito que envolve a mesma entidade do repositório. O objetivo desses métodos é aumentar o grau de qualidade dos dados no repositório integrado, bem como minimizar o trabalho manual de resolução de inconsistências.*

Palavras-chave: *resolução de conflitos, integração de dados, fusão de dados, limpeza de dados, qualidade dos dados, repositório de dados XML*

1. Introdução

Empresas de todos os tamanhos e de diversos segmentos implementam e utilizam os benefícios proporcionados pela manutenção de um repositório de dados. Está cada vez mais claro que o armazenamento de dados de forma integrada oferece uma excelente proposta para transformar a imensa quantidade de dados existentes nestas organizações em informação útil e confiável para responder suas questões e auxiliá-los nos processos de tomada de decisão. Um repositório de dados oferece uma base única para as técnicas de análise de dados existentes atualmente, tais como mineração de dados e análise multidimensional, além das tradicionais consultas e geração de relatório.

Prover o acesso a múltiplas fontes de dados através de um acesso integrado é um desafio que tem despertado o interesse de pesquisadores na área de sistemas de informação. Nesse contexto, dois problemas são fundamentais. Primeiro, como determinar se as fontes de dados contém informações semanticamente relacionadas, isto é, que dizem respeito ao mesmo conceito no mundo real. Segundo, como tratar a heterogeneidade semântica a fim de dar suporte ao processo de integração e permitir interfaces de consultas uniformes.

Sendo a Internet um enorme celeiro de informações e com o XML tornando-se o formato preferido para transmissão desses dados na grande rede, muitos trabalhos buscam melhorar a qualidade dos dados no formato XML. Além disso, é natural executar o processo de limpeza dos dados no próprio formato XML, uma vez que sua estrutura hierárquica representa naturalmente o relacionamento entre os dados. Ou seja, se os dados são mapeados para o modelo relacional, faz-se necessário a execução de junções para que seja possível obter a mesma visão de relacionamento proporcionada pela estrutura em forma de árvore XML. O objetivo deste trabalho é definir um conjunto de métodos e estratégias que permitam resolver conflitos de dados no formato XML que tenham sido identificados durante o processo de integração, para que o repositório mantenha uma visão única e consistente dos dados. Um segundo objetivo é manter um histórico de resoluções com a finalidade de resolver uma 'unica vez um conflito reincidente sobre uma mesma entidade do repositório.

O restante deste trabalho está dividido da seguinte forma: na Seção 2 são apresentados alguns trabalhos relacionados. Em seguida, na Seção 3, é apresentado o modelo da proposta e seus desafios. Por fim, a Seção 4 apresenta as considerações finais.

2. Trabalhos Relacionados

A qualidade dos dados armazenados é um assunto que preocupa os pesquisadores há alguns anos. Muitos trabalhos são encontrados na literatura tratando do estágio de limpeza dos dados. Rahm e Do [8] descrevem de forma abrangente os problemas e algumas soluções existentes nesta área de banco de dados. Potter's Wheel, um sistema interativo de limpeza de dados que proporciona resolução de conflitos é mostrado em [10]. No trabalho de Bleiholder e Naumann [2] é realizado um levantamento sobre os grandes desafios da integração de dados e resolução de conflitos. Estes trabalhos estão focados na limpeza e integração de dados heterogêneos relacionais.

Para dados XML, os trabalhos estão voltados principalmente para problemas como casamento de esquemas [6, 4] e identificação de entidades semanticamente corresponden-

tes [11, 7]. Ainda são raros os trabalhos que tratam especificamente da resolução de conflitos em dados XML. Os trabalhos que contribuem nesse sentido são, em sua maioria, baseados na utilização de ontologias [9, 3]. No entanto, percebe-se que a manutenção de ontologias é um processo árduo e complexo. Assim, a proposta deste trabalho leva em conta o histórico de resoluções como recurso para resolver inconsistências de dados de forma eficiente. Além disso, ele analisa a aplicabilidade em XML das estratégias de resolução de conflitos existentes para dados relacionais.

3. Modelo Proposto

Este trabalho utiliza como base o modelo definido em [5] que descreve uma forma de integração de dados XML na qual as inconsistências são apresentadas explicitamente. Neste artigo, para facilitar a apresentação do modelo, suas características são exemplificadas sobre o modelo relacional. Considere a Figura 1, na qual duas fontes de dados S e T, junto com seus respectivos esquemas (`A,B,ID`) e (`B,C,ID`), são integradas usando informações do mapeamento de esquemas (neste caso identificados pelos mesmos nomes) e também da detecção de duplicidade (neste caso, assume-se que as tuplas com o mesmo `ID` referem-se a mesma entidade do mundo real). Como resultado dessa integração tem-se um novo esquema integrado (`S.A,B,ID,T.C`). Observe que existem dados conflitantes para o atributo `B` das entidades com `ID=2` e `ID=3`. Este resultado é o objeto de entrada para a aplicação do conjunto de métodos para resolução de conflitos que este trabalho propõe.

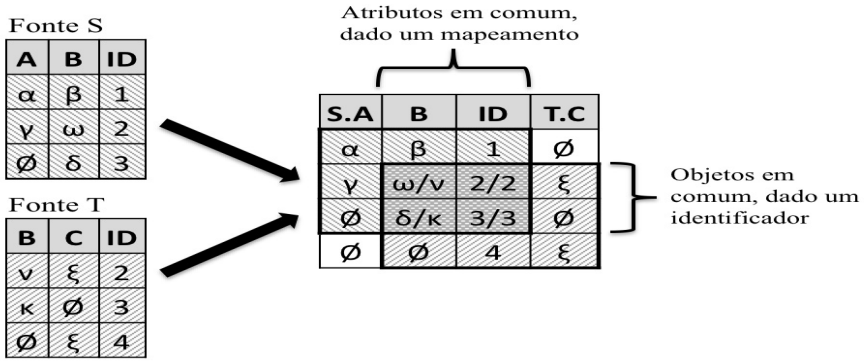

Figura 1. Integração de duas fontes de dados (S e T) com seus respectivos esquemas (`A,B,ID`) e (`B,C,ID`) em um resultado integrado (`S.A,B,ID,T.C`).

O modelo proposto por Nascimento [5] resolve uma parte importante do problema de integração de dados. Em seu modelo o autor considera questões referente a mapeamento entre os esquemas da fonte e do repositório e também trata da identificação de entidades. Assim, o processo de integração realizado em seu trabalho gera uma árvore XML combinada, incluindo informações de proveniência e explicitando os conflitos encontrados.

Como forma de garantir dados consistentes e sem redundância no repositório de dados, a proposta do presente trabalho é adicionar ao modelo um estágio de análise e limpeza de dados. A abordagem estendida está ilustrada na Figura 2. Neste modelo uma (ou mais) fonte de dados S é combinada com uma cópia do repositório R', gerada a partir do mapeamento definido entre fonte e repositório (omitido neste artigo, por limitação de espaço). Como resultado desse processo obtém-se um novo documento XML ($M_{S,R'}$) com dados combinados de ambas as fontes e conflitos explicitamente representados.

O documento $M_{S,R'}$ gerado pelo processo de combinação, passará por um estágio de limpeza de dados e resolução de conflitos. É neste estágio que este trabalho faz sua maior contribuição, pois aqui são definidas estratégias que decidem qual valor o atributo deve tomar para que o conflito seja resolvido. O diferencial que o modelo propõe é o armazenamento de um histórico (log) de resoluções de conflitos efetuadas sobre determinada entidade. Assim, quando surgirem novos conflitos envolvendo aquela entidade, o modelo terá informações adicionais que poderão ser consultadas e utilizadas na tomada de decisão. Por fim, após efetuar a resolução dos conflitos apresentados em $M_{S,R'}$, são gerados comandos de atualização que irão fazer a carga no repositório somente com dados consistentes e sem a redundância anteriormente existente. Assim, tem-se uma economia no espaço de armazenamento e maior precisão nas respostas das consultas.

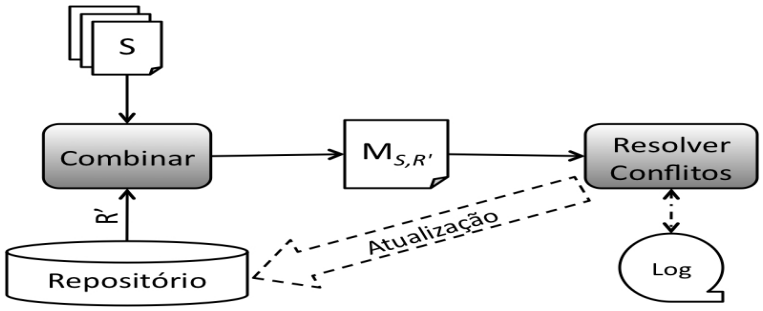

Figura 2. Modelo de integração de dados com resolução de conflitos, para atualização em repositório de dados XML.

Para proporcionar uma integração com qualidade este trabalho preocupa-se em garantir as seguintes propriedades:

I) "Fazer uma vez só". Esta propriedade garante que uma vez tomada a decisão sobre um conflito ocorrido em uma entidade, este conflito será resolvido automaticamente sempre da mesma forma. O comportamento da estratégia muda somente quando o administrador desejar.

II) Manter proveniência dos dados. A proveniência representa a origem de cada elemento do repositório integrado. Esta informação é essencial para determinar a qualidade e confiança que podem ser atribuídas aos dados.

III) Permitir a reconstrução da fonte. Com o documento reconstruído, torna-se possível efetuar comparações com uma versão atualizada da fonte e detectar quais foram as alterações ocorridas. Com isso, é possível propagar ao repositório de dados apenas as atualizações necessárias.

Estas propriedades auxiliam o sistema de integração a ter um bom desempenho na fase de resolução de conflitos e também a manter a qualidade e integridade dos dados armazenados. Mantendo um histórico de resoluções separado do repositório, pode-se solucionar de forma mais eficiente conflitos reincidentes sem sobrecarregar o repositório com informações adicionais. As anotações sobre proveniência ajudam a garantir as três propriedades, pois tem-se informações sobre a origem e confiabilidade do dado armazenado e possibilita, posteriormente, a reconstrução da fonte que originou aquele dado.

3.1. Estratégias para Resolução de Conflitos

Estratégias para resolução de conflitos são políticas aplicadas em um nível mais alto de tomada de decisão sobre o que fazer com os dados inconsistentes. Algumas dessas estratégias descrevem a decisão sobre qual valor tomar, como combinar os valores, criar um

novo valor ou até mesmo solicitar intervenção humana. Para verificar a aplicabilidade em XML das estratégias de resolução de conflitos em dados relacionais, inicialmente serão implementadas duas classes de estratégias, apresentadas em [1]:

Anular conflito: esta estratégia não resolve realmente o conflito, todavia manipula os dados inconsistentes. Ela não espera que o conflito aconteça para resolvê-lo, ou seja, não considera os valores dos dados antes de tomar a decisão. São estratégias aplicadas quando é necessário uma decisão rápida sobre o dado. Assim, caso haja um conflito em determinado dado sabe-se previamente qual valor tomar. Esta classe está subdividida em duas subclasses, uma que considera os metadados no momento da avaliação do conflito (baseada em metadado) e a outra que não faz essa consideração (baseada em instância). Um exemplo de estratégia que se encaixa na primeira subclasse é confiar na proveniência de um determinada fonte de dados, passando a assumir sempre os valores desta fonte quando houver um conflito. Para a segunda subclasse, um exemplo é considerar sempre a informação existente e deixar de lado valores nulos.

Resolver conflito: ao contrário das duas classes anteriores, esta classe de estratégia considera todos os dados e metadados antes de tentar resolver o conflito. Além disso, esta classe está subdividida em estratégias de decisão e estratégias mediadoras. A principal característica das estratégias de decisão é que elas escolhem seus valores a partir de todos os valores conflitantes e podem ou não levar em conta os metadados. As mediadoras, por outro lado, podem escolher um valor que não esteja entre os valores conflitantes, ou seja, pode criar um valor que não existia antes. Um exemplo de estratégia decisora é a escolha do valor que ocorre com mais frequência entre os valores conflitantes. Já no caso de um estratégia mediadora, poderia ser criado um valor com a média dos valores conflitantes, mantendo assim uma pequena margem de erro.

3.2. Estado Atual e Desafios

O trabalho está dividido em três fases. A primeira está concluída com a definição da proposta e o embasamento teórico sobre sistemas de integração de dados, processo de limpeza de dados e, especificamente, resolução de dados conflitantes. Na segunda fase, a ser iniciada, serão definidos os algoritmos e estratégias para resolução de conflitos entre dados XML. Por fim, na terceira e última etapa, ocorre a validação do modelo através de experimentos de integração com fontes de dados heterogêneas, possivelmente, provenientes da DBLP[1] (*Digital Bibliography Library Project*).

Os grandes desafios que surgem da definição dessa proposta são a formalização de operações e estratégias que irão compor o conjunto de métodos desse trabalho e a aplicação dos mesmos no estágio de limpeza de dados em sistemas de integração. Além disso, uma preocupação que demandará uma análise detalhada é a definição do registro de histórico. É necessário identificar quais são as informações mínimas que este registro precisa armazenar para garantir as propriedades mencionadas anteriormente.

4. Considerações Finais

Este trabalho apresentou uma proposta de extensão para o modelo proposto em [5]. A abordagem desta proposta sugere a adição de um estágio para resolução de conflitos, com o objetivo de melhorar a qualidade dos dados armazenados. Utilizando estratégias para

[1] http://www.informatik.uni-trier.de/ ley/db/

tomada de decisão sobre dados conflitantes, sugere-se a utilização de um histórico de resoluções capaz de armazenar decisões tomadas anteriormente sobre um determinado conflito. Com isso, espera-se resolver uma única vez um mesmo conflito sobre a mesma entidade. Com esta abordagem é esperado que o repositório integrado mantenha-se consistente e permita responder as consultas feitas de forma completa e precisa. Além disso, com o auxílio do histórico é esperado um bom desempenho dos algoritmos de resolução de conflito. Todos esses resultados serão validados com a aplicação de casos de testes durante a fase de experimentos.

Referências

[1] BLEIHOLDER, J., AND NAUMANN, F. Conflict handling strategies in an integrated information system. In *Proceedings of the International Workshop on Information Integration on the Web (IIWeb)* (2006).

[2] BLEIHOLDER, J., AND NAUMANN, F. Data fusion. *ACM Computing Survey 41*, 1 (2008), 1–41.

[3] DERONG, S., GE, Y., NAN, Y., AND TIEZHENG, N. Heterogeneity resolution based on ontology in web services composition. In *CEC-EAST'04: Proceedings of the E-Commerce Technology for Dynamic E-Business, IEEE International Conference* (2004), pp. 274–277.

[4] DO, H. H., AND RAHM, E. Matching large schemas: Approaches and evaluation. *Information Systems 32*, 6 (2007), 857–885.

[5] DO NASCIMENTO, A. M., AND HARA, C. S. A model for XML instance level integration. In *SBBD'08: Proceedings of the 23rd Brazilian symposium on Databases* (2008), pp. 46–60.

[6] HERNÁNDEZ, M. A., PAPOTTI, P., AND TAN, W.-C. Data exchange with data-metadata translations. *VLDB'08: Proceedings of the 34th International Conference on Very Large Data Bases 1*, 1 (2008), 260–273.

[7] LEITÃO, L., CALADO, P., AND WEIS, M. Structure-based inference of xml similarity for fuzzy duplicate detection. In *CIKM'07: Proceedings of the 16th ACM conference on Conference on information and knowledge management* (2007), pp. 293–302.

[8] RAHM, E., AND DO, H. H. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull. 23*, 4 (2000), 3–13.

[9] RAM, S., AND PARK, J. Semantic conflict resolution ontology (scrol): An ontology for detecting and resolving data and schema-level semantic conflicts. *IEEE Transactions on Knowledge and Data Engineering 16*, 2 (2004), 189–202.

[10] RAMAN, V., AND HELLERSTEIN, J. M. Potter's wheel: An interactive data cleaning system. In *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases* (2001), pp. 381–390.

[11] WEIS, M., AND NAUMANN, F. Detecting duplicate objects in xml documents. In *IQIS'04: Proceedings of the 2004 international workshop on Information quality in information systems* (2004), pp. 10–19.