X*Fusion*: Uma Ferramenta para Fusão e Limpeza de Dados XML

Carlo Marcello, Cristian Stroparo, Elisângela de Assis da Silva, Carmem Satie Hara

Departamento de Informática – Universidade Federal do Paraná (UFPR)
Curitiba – PR – Brasil

{cgm03, cls04, eas02, carmem}@inf.ufpr.br

Resumo. *Este artigo apresenta a ferramenta XFusion, a qual permite a fusão de documentos XML provenientes de fontes de dados heterogêneas e a limpeza das inconsistências detectadas. O XFusion utiliza um banco de dados XML nativo como repositório de dados e oferece ao usuário duas abordagens para efetuar a limpeza dos dados conflitantes através de uma interface gráfica.*

Abstract. *This paper presents XFusion, a tool that allows heterogeneous XML source documents to be combined into a single XML data repository. It also provides two approaches for solving value conflicts that may arise during the integration process. XFusion uses a native XML database as its underlying data repository, and provides the user with a graphical interface for data cleaning.*

1. Introdução

Em um repositório de dados, um dos maiores problemas enfrentados no processo de integração é a identificação de dados redundantes e correção de dados inconsistentes provenientes das diversas fontes que contribuem para o seu conteúdo. Documentos XML podem ser considerados fontes de dados potenciais, devido a sua flexibilidade estrutural e ao fato de serem amplamente utilizados para troca de informações. A ferramenta apresentada neste artigo, denominada X*Fusion*, segue a abordagem do modelo de integração de dados XML proposto em [Nascimento e Hara 2008], que possui as seguintes características: 1) identificação de entidades correspondentes entre as fontes de dados por meio de chaves para XML [Buneman et al. 2002]; 2) apresentação explícita dos conflitos encontrados entre as fontes de dados, facilitando o processo de limpeza; e 3) possibilidade de recriação da porção da árvore XML fonte que deu origem a um determinado conjunto de dados contido no repositório.

O X*Fusion* utiliza o SGBD XML nativo eXist [Meier 2002] para o armazenamento do repositório. A utilização de um SGBD permite considerar a definição de um esquema para o repositório, o que não é realizado pela implementação descrita em [Nascimento e Hara 2008]. A interação do usuário com o X*Fusion* é realizada através de uma interface gráfica, que permite a visualização do conteúdo do repositório. Além disso, são disponibilizadas operações para a resolução das inconsistências de valores das suas diversas fontes de forma manual ou automática. Desta forma, a ferramenta incorpora tanto a funcionalidade de integração como de limpeza do repositório integrado.

O restante do artigo está estruturado da seguinte forma: a seção 2 descreve a ferramenta X*Fusion*, suas funcionalidades, e um estudo experimental; os trabalhos relacionados são apresentados na seção 3 e a seção 4 conclui o artigo apresentando as considerações finais.

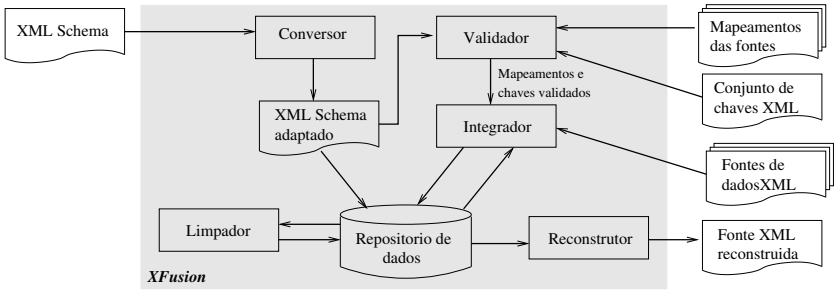


Figura 1. Arquitetura do X*Fusion*

2. A Ferramenta X*Fusion*

A ferramenta X*Fusion* possui um conjunto de módulos que compõem um sistema de integração e limpeza de documentos XML, cuja arquitetura é ilustrada na Figura 1. O sistema possui quatro entradas: 1) as fontes de dados XML; 2) os mapeamentos das fontes de dados: para cada fonte, o mapeamento determina quais dados são extraídos da fonte e quais são seus destinos dentro do repositório; 3) um XML Schema que define o esquema do repositório; 4) um conjunto de chaves para XML, que determinam de que forma os dados são integrados no repositório. O `Integrador` corresponde ao módulo que efetua a integração das fontes de dados XML ao repositório. Ele recebe as fontes de dados, seus mapeamentos, e o conjunto de chaves definido para o repositório. O XML Schema fornecido tem que ser modificado para incorporar características do modelo de integração, como anotações de proveniência e representação dos conflitos de valores (detalhados na seção 2.1). Por essa razão, existe um `Conversor`, que aplica as alterações necessárias no XML Schema original. O módulo `Limpador` permite ao usuário resolver os conflitos decorrentes do processo de integração, e o `Reconstrutor` reconstrói uma fonte de dados a partir do identificador da fonte.

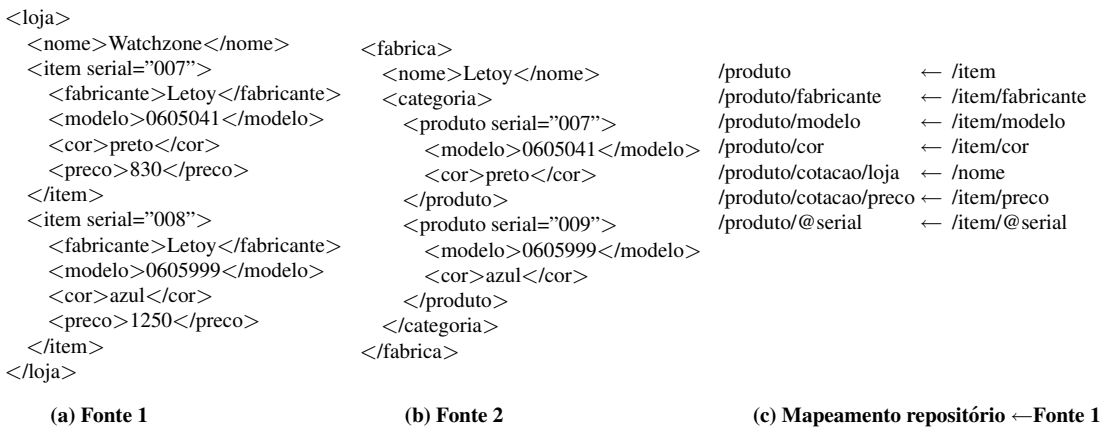
A ferramenta foi implementada utilizando os seguintes recursos: 1) linguagem de programação Java; 2) SGBD XML nativo eXist [Meier 2002]; 3) API JDOM[1]; e 4) API JAXEN[2]. O acesso ao banco de dados eXist é feito através da API XML:DB. Esta API fornece uma interface padrão para operações em bancos de dados XML, facilitando a portabilidade para outros SGBDs. Para consultas e atualizações dos dados armazenados são utilizadas as linguagens XQuery e XQuery Update. A API JDOM é usada para fazer o *parsing* e manipulação dos dados XML e a JAXEN é utilizada como uma *engine* XPath.

2.1. Funcionalidade da Ferramenta

O X*Fusion* efetua basicamente três operações: 1) mapeamento da fonte de dados para o esquema do repositório; 2) integração dos dados mapeados com o repositório, com as inconsistências representadas explicitamente; e 3) limpeza dos dados conflitantes. A operação de integração utiliza chaves para XML para determinar de que forma os dados são combinados. Cada nodo folha da árvore do repositório é anotado com a informação de proveniência, o que facilita o processo de limpeza de dados em caso de conflitos e também permite a reconstrução do documento XML fonte.

Considere as duas fontes de dados XML apresentadas na Figura 2(a) e (b). A `Fonte 1` descreve produtos vendidos por uma loja e a `Fonte 2` descreve uma empresa

[1] `http://www.jdom.org/`
[2] `http://jaxen.codehaus.org/`



(a) Fonte 1 **(b) Fonte 2** **(c) Mapeamento repositório ←Fonte 1**

Figura 2. Fontes de dados e Mapeamento

e os produtos que ela fabrica. O mapeamento consiste em definir uma correspondência entre o esquema da fonte de dados e o esquema do repositório. Através desse processo é possível convergir diversos esquemas heterogêneos provenientes das fontes de dados em um único esquema, facilitando a integração dos dados. A Figura 2(c) ilustra o mapeamento da `Fonte 1` para o repositório. Ele é composto por um conjunto de regras nas quais o lado direito representa elementos da fonte de dados que são extraídos e inseridos nos elementos do repositório descritos no lado esquerdo. A árvore mapeada é integrada ao repositório com base nas chaves para XML definidas sobre o mesmo. Por exemplo, pode-se definir uma chave que determine que um produto é identificado por seu fabricante e modelo. Logo, produtos com mesmo fabricante e modelo são dispostos em um mesmo ramo da árvore XML resultante. Se for definido que um produto deve possuir apenas uma cor, cores diferentes para um mesmo produto gerarão um conflito.

Tanto os mapeamentos quanto as chaves são fornecidos à ferramenta por meio de arquivos XML. O resultado da integração das duas fontes apresentadas na Figura 2 é dado na Figura 3. No exemplo é possível verificar dois conflitos. O primeiro ocorre no valor do elemento `cor` do produto identificado pelo modelo `0605041` (linhas 5 a 8). Cada valor conflitante é separado em um elemento `source` distinto. Um atributo especial, `prov`, armazena a informação de proveniência. Esta informação consiste em um par de valores, sendo que o primeiro corresponde à ordem de Dewey [Tatarinov et al. 2002] do elemento no documento fonte original e o segundo ao caminho do elemento no mesmo documento. No produto identificado pelo modelo `0605999` o conflito ocorre no atributo `serial` (linha 14). Elementos e atributos que não apresentaram conflitos também armazenam a proveniência. Por exemplo, na linha 17, o elemento `cor` possui dois valores de proveniência, indicando que as duas fontes de dados coincidem no valor deste elemento.

A interface gráfica implementada pela ferramenta disponibiliza a funcionalidade de integração de fontes e a visualização do repositório. Além disso, fornece funções para a resolução das inconsistências de valores utilizando duas abordagens: 1) *manual*: o usuário escolhe uma das fontes como sendo a detentora do valor correto para uma entrada inconsistente; alternativamente, pode-se definir manualmente um valor arbitrário caso nenhuma fonte contenha o valor correto; e 2) *baseada em precedência de fontes de dados*: a ferramenta permite que seja criada uma prioridade de fontes de dados de forma que na existência de conflitos dê-se precedência ao valor proveniente daquela com maior

```
<dw>
    <produto serial="007,[1.2.serial /item/@serial],[2.2.1.serial /categoria/produto/@serial]">
        <fabricante prov="[1.2.1,/item/fabricante];[2.1,/nome]">Letoy</fabricante>
        <modelo prov="[1.2.2,/item/modelo];[2.2.1.1,/categoria/produto/modelo]">0605041</modelo>
        <cor>
            <source prov="[1.2.3,/item/cor]">preto</source>
            <source prov="[2.2.1.2,/categoria/produto/cor]">branco</source>
        </cor>
        <cotacao>
            <loja prov="[1.1,/nome]">Watchzone</loja>
            <preco prov="[1.2.4,/item/preco]">830</preco>
        </cotacao>
    </produto>
    <produto serial="008,[1.3.serial /item/@serial];009,[2.2.2.serial /categoria/produto/@serial]">
        <fabricante prov="[1.3.1,/item/fabricante];[2.1,/nome]">Letoy</fabricante>
        <modelo prov="[1.3.2,/item/modelo];[2.2.2.1,/categoria/produto/modelo]">0605999</modelo>
        <cor prov="[1.3.3,/item/cor];[2.2.2.2,/categoria/produto/cor]">azul</cor>
        <cotacao>
            <loja prov="[1.1,/nome]">Watchzone</loja>
            <preco prov="[1.3.4,/item/preco]">1250</preco>
        </cotacao>
    </produto>
</dw>
```

Figura 3. Resultado da integração das fontes de dados da Figura 2

prioridade. Desta forma, os conflitos podem ser resolvidos automaticamente.

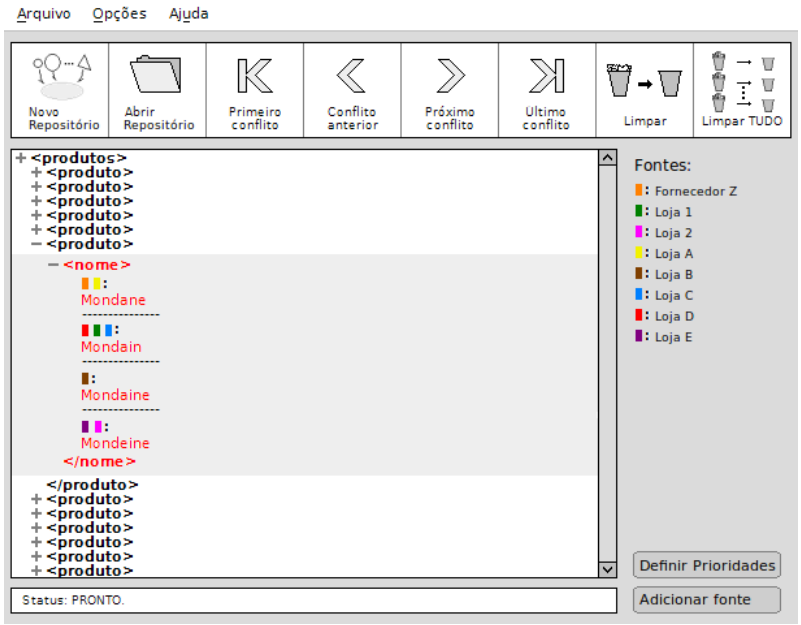


Figura 4. Interface do X*Fusion*

A interface da ferramenta é apresentada na Figura 4. A barra superior da ferramenta apresenta suas funções principais. Pode-se efetuar uma nova integração e gerar um novo repositório, visualizar seu conteúdo, bem como navegar pelos conflitos. Os dois últimos botões são para a resolução de conflito: o botão `Limpar` chama uma janela de diálogo na qual o usuário pode escolher um dentre os valores conflitantes ou criar um novo valor para o elemento; e o botão `Limpar Tudo` corresponde à correção automática baseada na precedência de fontes de dados. Na área de visualização principal, os conflitos são representados utilizando a cor vermelha para o nome do elemento e seu valor. Os valores conflitantes são precedidos pela identificação das suas fontes. Cada fonte de dados é identificada por um retângulo de uma determinada cor. A associação entre as cores e

os nomes das fontes é apresentada no painel vertical à direita da interface. Nesse painel é possível realizar duas operações: `Definir prioridades` e `Adicionar fonte`. Tanto um quanto o outro abrem uma nova janela de diálogo.

2.2. Estudo Experimental

Para demonstrar a funcionalidade da ferramenta e determinar o tempo de carga no repositório com diferentes níveis de reestruturação das fontes de dados, foram realizados dois experimentos. Os dados foram extraídos do repositório DBLP[3]. Apenas uma parte do dados foram utilizados, totalizando 739 KB (aproximadamente 20.000 linhas). Os experimentos foram realizados em um computador com processador Intel Centrino Core 2 Duo de 1.83GHz e 3GB de memória RAM. No primeiro experimento o mapeamento foi realizado preservando a estrutura da fonte de dados. No segundo, os dados foram agrupados por um elemento cujo valor era comum entre vários tipos de publicação, como `ano`. No primeiro experimento, o tempo de carga da fonte de dados no repositório foi de 668 segundos, resultando em um documento com tamanho de 1,28 MB. O tempo de carga para o segundo experimento foi de 519 segundos, e o tamanho do documento resultante foi de 1,26 MB. Devido ao agrupamento realizado no segundo experimento, e a inclusão de uma chave sobre o elemento agrupador, o espaço de busca das chaves diminuiu, resultando em um tempo de carga menor. O tamanho do documento resultante do segundo experimento também foi menor, pois o elemento agrupador, que antes se repetia no primeiro experimento, foi inserido apenas uma vez no repositório no segundo experimento. O tamanho do repositório gerado em ambos os experimentos é consistente com os resultados apresentados em [Nascimento e Hara 2008]. Os tempos elevados de carga precisam de uma melhor investigação, como alteração dos algoritmos propostos em [Nascimento e Hara 2008] e realização de testes de desempenho do SGBD eXist, como os reportados em [GEANT2-PerfSONAR 2006].

3. Trabalhos Relacionados

O processo de limpeza de dados, também conhecido como *data cleaning*, é essencial no processo de integração de dados, pois as diferentes fontes podem possuir formato e conteúdo distintos. Em [Rahm e Do 2000] diversas abordagens para efetuar a limpeza de dados, tanto para esquemas como para instâncias são apresentadas. De acordo com a classificação apresentada em [Bleiholder e Naumann 2008], o modelo de integração proposto em [Nascimento e Hara 2008] e implementado pelo X*Fusion* é o *Pass It On*. Isto significa que ele entrega os dados a um usuário ou aplicação com os valores conflitantes pendentes para limpeza. A resolução de conflitos de dados de forma manual tem sido objeto de intensa pesquisa nos últimos anos. Neste caso, o repositório integrado é chamado de *curated database* [Buneman et al. 2008]. Para resolução de inconsistências, os sistemas Fusionplex [Anokhin e Motro 2006] e HumMer [Bleiholder e Naumann 2006] utilizam diversos parâmetros além da proveniência para a resolução de conflitos de forma automática. Porém, ambos são definidos sobre o modelo relacional. Tendo em vista que o formato XML é atualmente um padrão para troca de informações, a adoção do mesmo para a visualização e resolução de conflitos é mais natural. A ferramenta descrita em [Tomazela et al. 2008] é voltada para o domínio de dados acadêmicos e permite reconciliar documentos XML. Porém, ao contrário do X*Fusion*, que permite a resolução de

[3]http://dblp.uni-trier.de

conflitos entre diversos documentos, em [Tomazela et al. 2008] isto é feito sempre entre pares de documentos. Além disso, ela não possui uma operação de resolução de conflito baseada em prioridade de fontes.

4. Conclusão

A ferramenta X*Fusion* apresentada neste artigo possui funcionalidades tanto para a integração de fontes XML quanto para a resolução de conflitos. O modelo de integração é baseado no trabalho descrito em [Nascimento e Hara 2008], mas o estende com as seguintes funcionalidades: validação de chaves e mapeamentos entre o esquema da fonte e do repositório; modificação do esquema do repositório para incorporar dados de proveniência e conflitos de dados; e utilização do SGBD XML nativo eXist para o armazenamento do repositório. Esta última característica permite a adoção das linguagens XQuery e XQuery Update para a manipulação dos dados armazenados. Trabalhos futuros incluem a investigação para a melhoria do desempenho do sistema, além da adição de novas estratégias de resolução de conflitos e uma máquina de aprendizagem para semi-automatizar o processo de limpeza dos dados.

Referências

Anokhin, P. e Motro, A. (2006). Fusionplex: Resolution of data inconsistencies in the integration of heterogeneous information sources. *Information Fusion Archive*, 7(2):176–196.

Bleiholder, J. e Naumann, F. (2006). Conflict handling strategies in an integrated information system. In *Workshop on Information Integration on the Web (IIWeb)*.

Bleiholder, J. e Naumann, F. (2008). Data fusion. *ACM Computing Surveys*, 41.

Buneman, P., Cheney, J., Tan, W., e Vansummeren, S. (2008). Curated databases. In *Proceedings of PODS' 2008*.

Buneman, P., Davidson, S., Fan, W., Hara, C., e Tan, W. (2002). Keys for XML. *Computer Networks*, 39(5):473–487.

GEANT2-PerfSONAR (2006). eXist DB XML performance tests. `https://wiki.man.poznan.pl/perfsonar-mdm/index.php/EXist_DB_XML_performance_tests`.

Meier, W. (2002). eXist: An open source native XML database. In *Web, Web-Services, and Database Systems: NODe 2002 Web and Database-Related Workshops*.

Nascimento, A. M. e Hara, C. (2008). A model for XML instance level integration. In *Anais do XXIII Simpósio Brasileiro de Banco de Dados (SBBD'2008)*.

Rahm, E. e Do, H. H. (2000). Data cleaning: Problems and current approaches. *IEEE Bulletin of the Technical Committee on Data Engineering*, 23(4).

Tatarinov, I., Viglas, S. D., Beyer, K., Shanmugasundaram, J., Shekita, E., e Zhang, C. (2002). Storing and querying ordered XML using a relational database system. In *Proceedings of SIGMOD'2002*, pages 204–215, Madison, Wisconsing, USA.

Tomazela, B., Ciferri, C. D. A., e Traina, C. (2008). Reconciliando dados de cunho acadêmico. In *Anais do XXIII Simpósio Brasileiro de Banco de Dados (SBBD'2008)*, pages 283–297.