

Geração de Regras para Manutenção Incremental de Visões XML Materializadas de Banco de Dados Objeto Relacional

Aluna: Valdiana da Silva Araujo
email: valdiana@lia.ufc.br

Orientadora: Vânia Maria Ponte Vidal
email: vvidal@lia.ufc.br

Nível: Mestrado
Programa: Mestrado em Ciências da Computação
Universidade Federal do Ceará
Departamento de Computação

Ano de Ingresso: 2003
Previsão de Conclusão: Dezembro/2004
Etapa concluída: defesa da proposta

Resumo

XML vem rapidamente emergindo como o formato universal para publicação e troca de dados na Web. Diante deste cenário, fontes de dados estão exportando seus dados através de visões XML. Estas visões podem ser materializadas para atingir um melhor desempenho no acesso aos dados. A principal dificuldade desta abordagem é manter a consistência dos dados da visão materializada com alterações realizadas na base de dados.

Neste trabalho, tratamos o problema de manutenção incremental de visões XML materializadas definidas sobre dados relacionais ou objeto-relacionais. No enfoque proposto, a visão XML é mantida através de regras ECA (Evento-Condição-Ação) definidas no Banco de dados. Essas regras são responsáveis por modificar corretamente o conteúdo das visões XML, de forma a refletir atualizações na base de dados.

O objetivo desse trabalho é o desenvolvimento de um ambiente para suportar a geração das regras para manutenção incremental de visões XML. As regras são definidas baseadas nas assertivas de correspondência da visão, que especificam formalmente os relacionamentos entre o esquema da visão e o esquema do banco de dados. Usando as assertivas de correspondência da visão, é possível: (i) identificar que atualizações nas tabelas do banco são relevantes para a visão; i.é. podem deixar a visão inconsistente com a banco, e (ii) dada uma operação relevante, calcular que mudanças são necessárias para propagar a atualização para a visão. O uso das assertivas de correspondência nos permite garantir que as regras geradas pelo nosso algoritmo mantêm corretamente a visão.

Palavras-chave: Visões XML Materializadas, Manutenção Incremental, Banco de Dados objeto-relacional.

1. Introdução

Nos últimos anos a Web tem se tornado o maior ambiente capaz de prover acesso a fontes de dados heterogêneas. Neste contexto XML vem emergindo como um padrão para representação e trocas de dados na Web. Isto cria a necessidade de se publicar dados, os quais estão armazenados em bases convencionais, no formato XML.

Uma forma geral de publicar esses dados é feita através do uso de visões XML, as quais podem ser materializadas ou virtuais. Visões materializadas apresentam um melhor desempenho em consultas, mas necessitam ser atualizadas para refletir atualizações realizadas no banco de dados, de forma a manter a consistência dos dados.

Basicamente, existem duas estratégias para manter visões materializadas: re-materialização e manutenção incremental. Na estratégia de re-materialização, todos os dados da visão são re-computados em tempos pré-estabelecidos. Por outro lado, na estratégia incremental, um mecanismo modifica, periodicamente, parte dos dados da visão para refletir as atualizações ocorridas no banco de dados.

No nosso enfoque, para manter a visão materializada, é necessário gerar regras que serão disparadas em reação às atualizações ocorridas no banco de dados que sejam relevantes para visão. Neste trabalho, propomos um enfoque para geração semi-automática de regras para manutenção incremental de visões XML sobre banco de dados objeto relacional. O trabalho se restringe a visões preservadoras de objetos, que são visões que mantêm um relacionamento 1-1 entre os elementos bases da fonte de dados.

As seções seguintes estão organizadas como se segue. Na seção 2, apresentamos trabalhos relacionados. Na seção 3, descrevemos o processo proposto para a geração das regras de manutenção da visão. Na seção 4, apresentamos as conclusões.

2. Trabalhos Relacionados

O problema de manutenção de visões tem sido extensivamente estudado para bancos relacionais e objetos [1, 6, 7]. Abiteboul et al [2] propôs um algoritmo de manutenção incremental para visões materializadas de dados semi-estruturados, considerando um grafo baseado no modelo de dados OEM e a linguagem de consulta Lorel. El-Sayed et al [4], propondo uma abordagem de solução algébrica baseada na álgebra XAT XML. Não temos conhecimento de nenhum trabalho que aborde o problema da Manutenção Incremental de visões XML sobre bancos de dados objeto-relacional.

3. Gerando Regras para Manutenção Incremental de Visões XML

Nesta Seção, considere o esquema XMLS⁺ da visão **V** mostrado na figura 3.1, que publica dados do banco de dados **Bib** cujo esquema é mostrado na figura 3.2. XMLS⁺ (XML Schema Semântico) é uma extensão de XML Schema[5] que permite representar graficamente a estrutura dos tipos XML e incorpora mais semântica aos esquemas, uma vez que permite representar algumas formas de

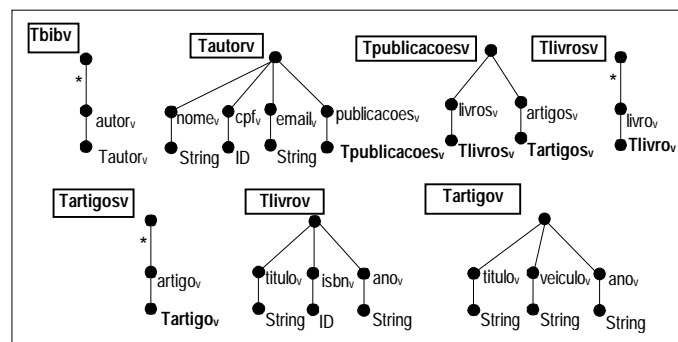


Fig. 3.1: Esquema XMLS⁺ da Visão V

relacionamentos semânticos entre os tipos XML.

O esquema XMLS⁺ de **V** contém sete árvores de tipo. A árvore do tipo **Tbib_v**, que é o tipo do elemento raiz, contém zero ou mais os elementos **autor_v** do tipo **Tautor_v**. O tipo **Tautor_v** contém os elementos **nome_v**, **email_v** e **publicacoes_v** do tipo **Tpublicacoes_v**. O tipo **Tpublicacoes_v** contém os elementos **livros_v** e **artigos_v** dos tipos **Tlivros_v** e **Tartigos_v**, respectivamente. O tipo **Tlivros_v** contém zero ou mais elementos **livro_v** do tipo **Tlivro_v** e o tipo **Tlivro_v** possui os elementos **titulo_v**, **isbn_v**, **ano_v** e **editora_v**. O tipo **ID**, especificado para o elemento **isbn_v**, permite uma identificação única para cada elemento **autor_v** do tipo **Tautor_v**.

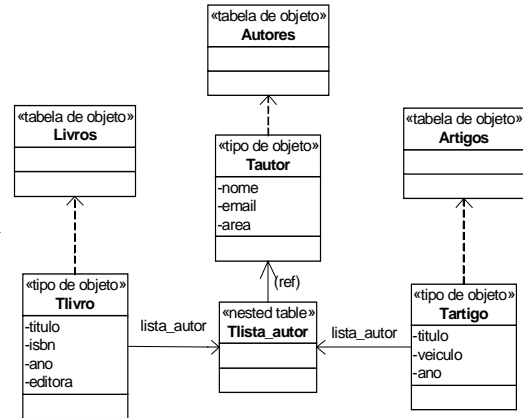


Fig. 3.2: Esquema do Banco de Dados Bib

No enfoque proposto, o processo de geração das regras consiste de 4 passos, descritos a seguir.

Passo1: Geração do esquema XMLS⁺ do Banco: Nesse passo, é realizado o mapeamento do esquema do banco de dados no esquema XMLS⁺[8] correspondente. Como veremos no passo 2, esse mapeamento é necessário para que possamos especificar de forma mais precisa as correspondências entre os elementos da visão XML e os objetos/tabelas do banco de dados.

Figura 3.3 mostra o esquema XMLS⁺ da fonte de dados **Bib**. O esquema XMLS⁺ de **Bib** contém oito árvores de tipo. A árvore do tipo **Traiz**, que é o tipo do elemento raiz, contém os elementos **livros** do tipo **Tlivros**, **artigos** do tipo **Tartigos** e **autores** do tipo **Tautores**. O tipo **Tlivros** contém zero ou mais elementos **livro** do tipo **Tlivro**. O tipo **Tlivro** contém os elementos **titulo**, **isbn**, **ano** e o elemento **lista_autor** do tipo **Tlista_autor**. O tipo **Tlista_autor** contém zero ou mais elementos **autor_ref** que é uma referência para um elemento **autor** do tipo **Tautor**.

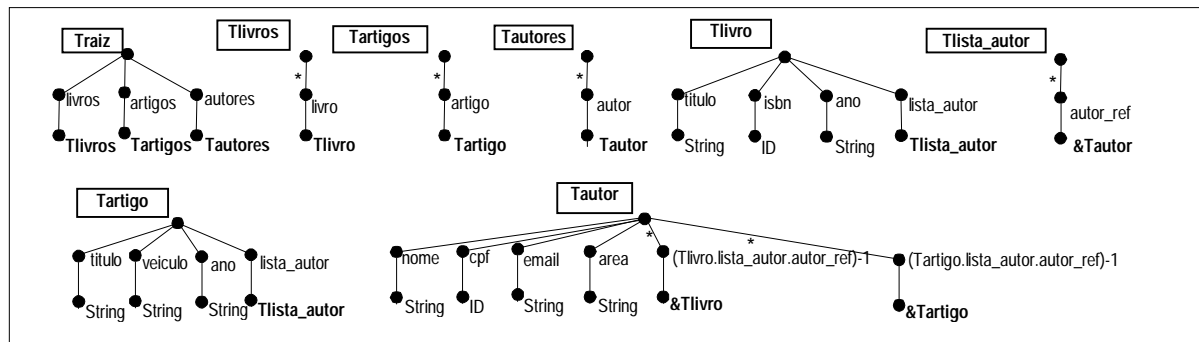


Fig. 3.3: Esquema XMLS⁺ do Banco de Dados Bib

Passo2: Geração das Assertivas de Correspondência da Visão: Neste passo é realizado o “matching” [8, 9] do esquema XMLS⁺ da visão e o esquema XMLS⁺ do banco de dados de forma a identificar as assertivas de correspondência (AC) da visão. As ACs especificam formalmente a correspondência entre o esquema XMLS⁺ da visão XML e o esquema XMLS⁺ do banco de dados. A vantagem desse formalismo é que as regras para manutenção da visão XML podem ser geradas, de forma automática, a partir das ACs da visão. As ACs são gerados baseadas no enfoque apresentado em [8].

Nós exemplificaremos o processo descrito em [8], gerando as ACs de **V**. O processo de geração das ACs consiste de dois passos:

Passo 2.1. Primeiro especificamos a assertiva de correspondência para os elementos primários de V . O conjunto de elementos autor_v em $\$/\text{autor}_v$ é definido pela assertiva de correspondência de coleção global (ACCG) [3]:

$$\Psi_1: [\$/\text{autor}_v] \equiv [\$/\text{bib/atores/autor}[\text{area} = \text{"Database"}]]$$

Ψ_1 especifica que $\$/\text{autor}_v$ e $\$/\text{bib/atores/autor}[\text{area} = \text{"Database"}]$ denotam o mesmo conjunto de objetos do mundo real.

Devemos introduzir também as assertivas de correspondência de elemento (ACE) para especificar o critério de “matching” de elementos em coleções semanticamente relacionadas. A ACE $\Psi_2: [\text{Tautor}_v, \{\text{cpf}_v\}] \equiv [\text{Tautor}, \{\text{cpf}\}]$ especifica que um autor $\$/a_v$ em $\$/\text{autor}_v$ corresponde a um autor $\$/a$ em $\$/\text{bib/atores/autor}$ sse $\$/a_v/\text{cpf}_v/\text{text}() = \$/a/\text{cpf}/\text{text}()$.

Passo 2.2: Em seguida, especificamos as ACs para os sub-elementos de autor_v . A figura 3.4 mostra as AC de Caminho (ACCs) e de Elementos obtidas da comparação dos tipos Tautor_v e Tautor . Por exemplo, Ψ_6 especifica que, dada uma instância $\$/a_v$ de Tautor_v , se existir uma instância $\$/a$ de Tautor tal que $\$/a_v \equiv \$/a$, então $\$/a_v$ contém um elemento publicacoes_v tal que $\$/a_v/\text{publicacoes}_v/\text{livros}_v/\text{livro}_v \equiv \$/a/(\text{author_ref-})^{-1}$. Como Tlivro_v é um tipo complexo, nós devemos definir a ACE Ψ_8 , que determina a correspondência entre instâncias de Tlivro_v e instâncias de Tlivro , e as ACCs Ψ_9 , Ψ_{10} , Ψ_{11} e Ψ_{12} , obtidas pela comparação entre os tipos Tlivro e Tlivro_v .

<p>ACCs entre Tautor_v e Tautor:</p> <p>$\Psi_3: [\text{Tautor}_v/\text{nome}_v] \equiv [\text{Tautor}/\text{nome}]$ $\Psi_4: [\text{Tautor}_v/\text{cpf}_v] \equiv [\text{Tautor}/\text{cpf}]$ $\Psi_5: [\text{Tautor}_v/\text{email}_v] \equiv [\text{Tautor}/\text{email}]$ $\Psi_6: [\text{Tautor}_v/\text{publicacoes}_v/\text{livros}_v/\text{livro}_v] \equiv [\text{Tautor}/(\text{Tlivro.lista_autor.autor_ref-})^{-1}]$ $\Psi_7: [\text{Tautor}_v/\text{publicacoes}_v/\text{artigos}_v/\text{artigo}_v] \equiv [\text{Tautor}/(\text{Tartigo.lista_autor.autor_ref-})^{-1}]$</p> <p>ACE entre Tlivro_v e Tlivro:</p> <p>$\Psi_8: [\text{Tlivro}_v, \{\text{isbn}_v\}] \equiv [\text{Tlivro}, \{\text{isbn}\}]$</p>	<p>ACCs entre Tlivro_v e Tlivro:</p> <p>$\Psi_9: [\text{Tlivro}_v/\text{titulo}_v] \equiv [\text{Tlivro}/\text{titulo}]$ $\Psi_{10}: [\text{Tlivro}_v/\text{isbn}_v] \equiv [\text{Tlivro}/\text{isbn}]$ $\Psi_{11}: [\text{Tlivro}_v/\text{ano}_v] \equiv [\text{Tlivro}/\text{ano}]$ $\Psi_{12}: [\text{Tlivro}_v/\text{editora}_v] \equiv [\text{Tlivro}/\text{editora}]$</p> <p>ACE entre Tartigo_v e Tartigo:</p> <p>$\Psi_{13}: [\text{Tartigo}_v, \{\text{titulo}_v\}] \equiv [\text{Tartigo}, \{\text{titulo}\}]$</p> <p>ACCs entre Tartigo_v e Tartigo:</p> <p>$\Psi_{14}: [\text{Tartigo}_v/\text{titulo}_v] \equiv [\text{Tartigo}/\text{titulo}]$ $\Psi_{15}: [\text{Tartigo}_v/\text{veiculo}_v] \equiv [\text{Tartigo}/\text{veiculo}]$ $\Psi_{16}: [\text{Tartigo}_v/\text{ano}_v] \equiv [\text{Tartigo}/\text{ano}]$</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Fig. 3.4 – ACCs e ACEs obtidas com a comparação entre os tipos Tautor_v e Tautor

Passo 3: Identificação das operações relevantes: Neste passo, são identificadas quais atualizações no banco de dados são relevantes para visão. Uma atualização no banco de dados é relevante para a visão quando ela pode tornar o banco de dados inconsistente com a visão. Baseado nas ACs da visão é possível identificar as operações relevantes de uma visão. Por exemplo, da AC Ψ_6 temos que a operação de atualização INSERÇÃO NA TABELA LIVROS é uma operação relevante para a visão V .

Passo 4: Geração das Regras para manutenção da visão: Para cada operação relevante, identificada no passo anterior, deve ser gerada a regra que mantém a visão. Quando uma regra é disparada por uma operação relevante, primeiro é verificado se a visão está inconsistente com o novo estado do banco. Em caso positivo, é então realizada as atualizações necessárias na visão, de forma que esta fique consistente com o novo estado do banco.

Por exemplo, a figura 3.5 mostra a regra $R\alpha$ gerada para manutenção de V com relação à inserção na tabela LIVROS.

- O comando da linha 1 obtém o conjunto de pares $\langle \$/t, \$/c \rangle$ onde $\$/t$ é o objeto alvo (tabela ou objeto pai) e $\$/c$ é o objeto inserido (na tabela ou em uma tabela aninhada de $\$/t$) em consequência da inserção de $\$/novo$ em Livros.
- O comando da linha 3 verifica para cada par de objetos identificados na linha 1 se satisfaz as condições de seleção da visão definida pela AC de coleção global da visão.

- Os comando das linhas 4-5 identificam o elemento alvo($\$t_v$), o elemento a ser inserido($\$c_v$) e realiza a inserção. O procedimento **ConstrutorTlivro_Tlivro_v**, chamado na linha 5, cria um elemento da visão do tipo Tlivro_v a partir de objeto do banco de dados ($\$c$) do tipo Tlivro. O elemento criado é semanticamente equivalente ao objeto do banco de dados, para isto o procedimento é definido automaticamente baseado nas ACs de Tlivro_v e Tlivro. A figura 3.6 mostra o procedimento **ConstrutorTlivro_Tlivro_v**.

```

{
1. ObjetosAtualizados := {<$t,$c> | $t em $novo.autores e $c = $novo}
2. para cada tupla <$t, $c> em ObjetosAtualizados faça
3.   se $t.area="Banco de Dados" então
4.     $t_v := $V/autor[email_v=$t.email]/publicacoes_v/livros_v
5.     $c_v := ConstrutorTlivro_Tlivro_v($c)
6.     INSERE($t_v,$c_v)
}

```

Fig. 3.5: Regra $R\alpha$ gerada para manter a visão V com relação à Inserção na Tabela LIVROS.

```

ConstrutorTlivro_Tlivro_v($c)
<livro_v>
<titulo_v> $c.titulo </titulo_v>
<isbn_v> $c.isbn </isbn_v>
<ano_v> $c.ano </ano_v>
<editora_v> $c.editora <editora_v>
</livro_v>

```

Fig. 3.6: ConstrutorTlivro_Tlivro_v.

A regra $R\alpha$ é executada sempre que ocorre uma inserção na tabela livros do banco de dados. A seguir mostramos a execução da regra $R\alpha$ para uma inserção na tabela LIVROS.

Considere os estados iniciais do banco de dados Bib e da visão XML V mostrados, respectivamente, nas figuras 3.7 e 3.8.

Artigos				
	titulo	veiculo	ano	lista_autor
\$ar1	XML	VLDB	2001	ref(\$a1)

Autores			
	nome	email	area
\$a1	Alan	alan@oxford.com	Banco de Dados
\$a2	Max	max@oxford.com	Inteligência Artificial

Livros					
	titulo	isbn	ano	editora	lista_autor
\$b1	Relations	1-003-05	2001	Klower	ref(\$a1)

Fig. 3.7 Estado inicial do Banco Bib

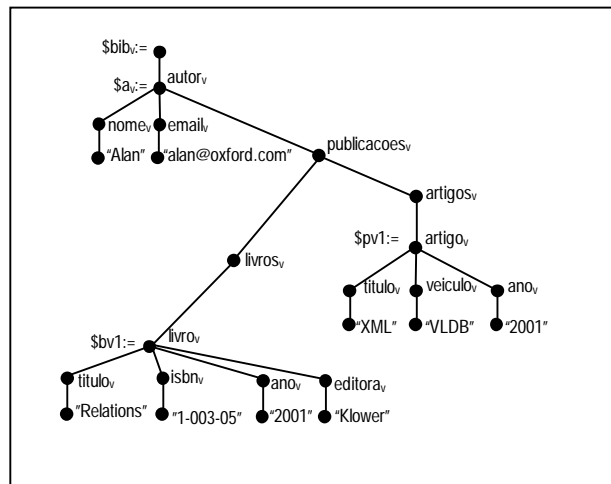


Fig. 3.8 Estado inicial da Visão V

Suponha a atualização α_1 , sendo a inserção de um objeto do tipo Tlivro na tabela Livros de **Bib**. A figura 3.9 mostra o novo estado do banco após a execução de α_1 .

Para esta atualização realizada no banco, a variável ObjetosAtualizados na regra $R\alpha$ recebe as seguintes tuplas: {<\$a₁, \$b₂>, <\$a₂, \$b₂>}. Seguindo a execução da regra, para cada tupla em ObjetosAtualizados, é verificada a condição de seleção, a qual é satisfeita apenas para a tupla <\$a₁, \$b₂> (\$a₁.area = "Banco de Dados"). Para essa tupla, temos:

- (i) $\$t_v := \$V/autor[email_v=\$t.email]/publicacoes_v/livros_v$
- (ii) $\$c_v := <livro_v><titulo_v> Data Mining </titulo_v>$
 $<isbn_v> 1-00305 </isbn_v>$
 $<ano_v> 2002 </ano_v>$
 $</livro_v><editora_v> Klower <editora_v>$

Em seguida é executado o comando de inserção do elemento $\$c_v$ na visão V como filho do elemento $\$t_v$. O novo estado da visão V é mostrado na figura 3.10

Artigos				
	titulo	veiculo	ano	lista_autor
\$a1	XML	VLDB	2001	ref(\$a1)

Autores			
	nome	email	area
\$a1	Alan	alan@oxford.com	Banco de Dados
\$a2	Max	max@oxford.com	Inteligência Artificial

Livros					
	titulo	isbn	ano	editora	lista_autor
\$b1	Relations	1-003-05	2001	Klower	ref(\$a1)
					ref(\$a1)
\$b2	Data Mining	1-003-06	2002	Klower	ref(\$a2)

Fig. 3.9 Estado do banco Bib após a atualização α_1

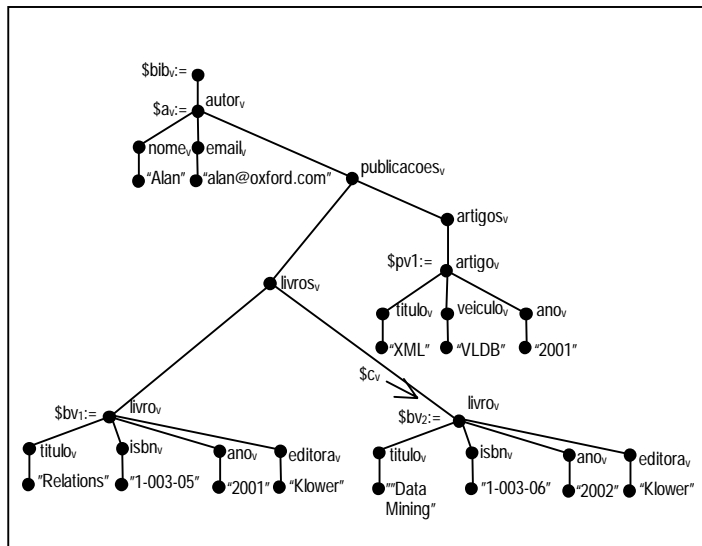


Fig. 3.10: Estado da Visão V após a execução da regra $R\alpha$

4. Conclusões

O objetivo deste trabalho é desenvolver um ambiente para geração de regras para manutenção incremental de visões XML de dados armazenados em Banco de dados Objeto-Relacional. O ambiente deve suportar todos os passos do processo de geração das regras. Atualmente estamos trabalhando no desenvolvimento dos algoritmos para identificação das operações relevantes e geração das regras.

Referências

- [1] Ali, M.A., Fernandes, A.A., Paton, N.W.: **Incremental Maintenance for Materialized OQL Views**. In Proc. DOLAP (2000) 41–48
- [2] Abiteboul, S., McHugh, J., Rys, M., Vassalos, V., Wiener, J.L.: **Incremental Maintenance for Materialized Views over Semistructured Data**. In Proceedings of the International Conference on Very Large Databases. New York City (1998) 38-49
- [3] Casanova, M. A., Vidal, V. M. P: **Efficient Maintenance of XML Views Using the View's Correspondence Assertions**. A ser publicado no ECWeb 2003
- [4] EL-Sayed, M., Wang, L., Ding, L., Rudensteiner, E.: **An algebraic approach for Incremental Maintenance of Materialized Xquery Views**. In Proceedings of Fourth International Workshop on Web Information and Data Management. McLean, USA (2002)
- [5] Fallside D.: **W3C Recommendation - XML Schema Part 0: Primer**. <http://www.w3.org/TR/xmlschema-0/>, Maio 2001.
- [6] Gupta, A., Mumick, I.S., Subrahmanian, V.S.: **Maintaining Views Incrementally**. In SIGMOD (1993) 157–166
- [7] Kuno, H.A., and Rudensteiner, E.A.: **Incremental Maintenance of Materialized Object-Oriented Views in MultiView: Strategies and Performance Evaluation**. IEEE Transaction on Data and Knowledge Engineering, 10(5):768–792 (1998)
- [8]. Vidal, V.M.P., Vilas Boas, R.: **A Top-Down Approach for XML Schema Matching**. In Proceedings of the 17th Brazilian Symposium on Databases. Gramado, Brazil (2002).
- [9] Vilas Boas, R.: **XMLS+Matcher: Um Método para Matching de XML Schemas Semânticos**. Dissertação de Mestrado, Universidade Federal do Ceará, Fortaleza, Ceará, Brasil, 2002.