Uma Estratégia baseada em Técnicas de KDD para apoiar o Projeto Físico em SGBDs XML Nativos.

Conference Paper · January 2007

Source: DBLP

3 authors:

Miriam Oliveira
Universidade Federal do Estado do Rio de Ja…
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Ronaldo R. Goldschmidt
Federal Rural University of Rio de Janeiro
26 PUBLICATIONS **15** CITATIONS
SEE PROFILE

M. C. Cavalcanti
Instituto Militar de Engenharia (IME)
52 PUBLICATIONS **187** CITATIONS
SEE PROFILE

Uma Estratégia baseada em Técnicas de KDD para apoiar o Projeto Físico em SGBD's XML Nativos

Miriam Oliveira dos Santos[1], Ronaldo Goldschmidt[2], Maria Cláudia Cavalcanti[1]

[1]Instituto Militar de Engenharia (IME)
Pça Gen Tiburcio, 80 – Praia Vermelha – Urca
CEP 22.290 – 270 – Rio de Janeiro – RJ – Brasil

[2]Instituto Superior de Tecnologia do Rio de Janeiro (ISTCC-RJ FAETEC)
Rua Clarimundo de Melo, 847 – Quintino Bocaiuva
CEP 21.311 – 280 – Rio de Janeiro – RJ – Brasil

{mosantos,rribeiro,yoko}@ime.eb.br

Abstract. *Technological advances have contributed for an expressive increase of Web information, in terms of volume and diversity. Much of this information is organized as XML documents and come from many different sources. Content management for heterogeneous XML documents does not provide efficient mechanisms for guiding the storage of these documents, in such a way that facilitates their retrieval. Therefore, this paper presents a strategy based on Knowledge Discovery and Data Mining to guide the storage of heterogeneous XML documents. A case study and a performance comparative analysis illustrate the potential of the proposed strategy.*

Keywords: semi-structured data, database physical design, knowledge discovery, data mining

Resumo. *Os avanços tecnológicos têm contribuído para um expressivo aumento do volume e da diversidade de informações que circulam atualmente pela Web. Muitas dessas informações encontram-se organizadas em documentos XML e provêm de várias fontes. O gerenciamento de conteúdo envolvendo documentos XML heterogêneos carece de mecanismos que orientem o processo de armazenamento desses documentos, de forma a facilitar sua posterior recuperação. Assim sendo, este artigo apresenta uma estratégia baseada em princípios de Descoberta de Conhecimento e Mineração de Dados para orientar o processo de armazenamento de documentos XML heterogêneos. Um estudo de caso e uma análise comparativa de desempenho ilustram o potencial da estratégia proposta.*

Palavras-chave: dados semi-estruturados, projeto físico de banco de dados, descoberta de conhecimento, mineração de dados

1. Introdução

Os avanços tecnológicos ocorridos sobretudo na última década têm contribuído para um expressivo aumento do volume e da diversidade de informações que circulam atualmente pela web. São inúmeras e heterogêneas as fontes de dados, assim como os dispositivos de saída a que tais informações se destinam. Diante desse cenário, mecanismos de gerenciamento de conteúdo [Pereira e Bax 2002] que apóiem a captação, organização e distribuição de informação corporativa vêm assumindo papel de

extrema importância nas organizações modernas. Os Sistemas de Gerenciamento Eletrônico de Documentos (GED's) e os Sistemas Gerenciadores de Bancos de Dados (SGBD's) são exemplos de recursos voltados ao gerenciamento de conteúdo. Os GED's têm sido utilizados no armazenamento, localização e recuperação de informações não estruturadas, e atendem a categorias específicas de documentos. Os SGBD's tradicionais, por outro lado, têm sido aplicados na gestão de informações estruturadas e requerem portanto, a definição de estruturas de dados prévias. No entanto, há uma demanda crescente no sentido de que os GED's e SGBD's se voltem para o gerenciamento de documentos semi-estruturados e heterogêneos, que vêm sendo representados através da linguagem XML.

O interesse em monitorar diferentes fontes de informações está presente em várias áreas, entre as quais podem ser citadas: prospecção tecnológica, governo, comércio eletrônico, aplicações de marketing, integração de redes de notícias, serviços de alfândega. O governo, por exemplo, através da Receita Federal, tem interesse em confrontar as informações declaradas no imposto de renda com outros documentos, em busca de dados conflitantes. Para isto, a Receita Federal deve ter acesso a diferentes informações dos contribuintes disponibilizadas em diferentes fontes, formatos e em diferentes contextos. Muitas destas informações podem ser encontradas em formato XML, que devido à sua flexibilidade quanto à forma de representação, tem sido amplamente adotado em diversas áreas.

O gerenciamento de conteúdo envolvendo documentos XML heterogêneos carece de mecanismos que orientem o processo de armazenamento desses documentos, de forma a facilitar sua posterior recuperação. Várias das abordagens encontradas na literatura [Abiteboul et al. 2005], [Flesca et al. 2005], [Yang et al. 2005], [Nierman e Jagadish 2002] e [Lee et al. 2001] propõem soluções voltadas para acervos com baixo grau de heterogeneidade, onde a intenção é extrair um esquema único. Já [Li et al. 2004], embora facilite a formulação de consultas sobre documentos heterogêneos, não provê solução para o armazenamento destes documentos. Assim sendo, este artigo tem como objetivo propor uma estratégia baseada em princípios de Descoberta de Conhecimento e Mineração de Dados para orientar o processo de armazenamento de documentos XML heterogêneos, separando-os em grupos mais homogêneos.

O presente trabalho está organizado da seguinte forma. Na Seção 2, descrevemos brevemente algumas das abordagens atuais para o armazenamento e classificação de documentos XML, discutindo alguns trabalhos relacionados. Na Seção seguinte, apresentamos uma proposta de estratégia para apoiar o projeto físico em SGBD's XML Nativos, a partir da aplicação de técnicas de KDD. A Seção 4 apresenta o protótipo implementado. As Seções 5 e 6 apresentam um estudo de caso e uma discussão sobre os resultados obtidos. Por fim, a Seção 7 conclui o artigo.

2. Técnicas e Abordagens Relacionadas

As duas principais abordagens para o armazenamento de documentos XML envolvem SGBD's relacionais e SGBD's XML Nativos. Em geral, nos SGBD's relacionais o armazenamento consiste na fragmentação do documento em diversas tabelas através de técnicas de mapeamento. No mapeamento direto, tabelas específicas são criadas a partir dos elementos (*tags*) e atributos do documento XML [Vieira 2002]. Já no mapeamento indireto existem estruturas genéricas que armazenam o conteúdo dos elementos e

atributos presentes em qualquer documento XML [Florescu e Kossmann 1999], [Tian et al. 2002] e [Vieira 2002]. Uma das desvantagens do mapeamento direto é a necessidade de se criar uma estrutura específica, enquanto que no mapeamento indireto, mesmo que não seja preciso a criação de estruturas específicas, ainda há a necessidade de um conhecimento prévio do esquema genérico adotado (não necessariamente padrão).

Já nos SGBD's XML Nativos [Schöning 2001],[Jagadish et al. 2002] e [Brian 2006], os documentos são armazenados sem que haja necessidade de se realizar mapeamentos. Os SGBD's XML Nativos, próprios para o armazenamento desta categoria de documentos, oferecem maior flexibilidade em relação aos SGBD's relacionais tradicionais, porque não há obrigatoriedade da definição de um esquema. Os documentos são armazenados dentro de coleções que são repositórios de documentos XML. Não há restrição para os "tipos" de documentos armazenados em uma mesma coleção, que pode conter documentos com diferentes estruturas.

Embora estas alternativas ofereçam soluções para o armazenamento de documentos XML, elas não consideram o tratamento de um acervo de grandes proporções de documentos XML heterogêneos. Neste sentido, procura-se por soluções de armazenamento que sejam mais eficientes, como por exemplo realizar um tratamento prévio deste acervo, classificando e/ou agrupando seus documentos.

KDD (*Knowledge Discovery in Databases*) é definido por [Fayad et al. 1996] como um processo de várias etapas, não trivial, interativo e iterativo, para identificação de padrões compreensíveis, válidos, novos e potencialmente úteis a partir de grandes conjuntos de dados. A descoberta de conhecimento em bases de dados é composta por três etapas operacionais: pré-processamento, processamento ou mineração de dados e pós-processamento. A etapa de pré-processamento é responsável pela captação, organização e tratamento dos dados. A etapa de processamento ou mineração de dados é responsável pela busca efetiva de conhecimento através da aplicação de algoritmos de mineração. A etapa pós-processamento é responsável pela avaliação e pela aplicação do conhecimento obtido [Goldschmidt e Passos 2005].

Diversas abordagens baseadas em KDD têm sido propostas para tratamento de conjuntos de documentos XML heterogêneos. Em [Abiteboul et al. 2005], [Flesca et al. 2005], [Yang et al. 2005], [Nierman e Jagadish 2002] e [Lee et al. 2001] é proposta a criação de um esquema único a partir da percepção da similaridade entre os documentos. Em alguns destes trabalhos são descritas técnicas para preparar os documentos XML para mineração de dados através da equiparação terminológica dos termos presentes nos documentos. Entretanto, a idéia de derivar um esquema único a partir de um acervo heterogêneo de documentos pode se tornar uma tarefa difícil, se não inviável, dependendo do grau de heterogeneidade do mesmo. Em [Li et al. 2004] é proposta uma estratégia para se trabalhar com coleções heterogêneas, baseada na formulação de consultas a diferentes esquemas que mantenham conteúdos similares. Diferente da nossa estratégia neste caso, os documentos não são agrupados por similaridade, permanecendo nas unidades de armazenamento originais, e portanto, sem os benefícios do armazenamento em separado e da indexação, que agilizariam as consultas.

Neste trabalho, propomos uma estratégia que envolve o uso de tarefas de KDD, como clusterização, sumarização e associação, no sentido de minimizar a heterogeneidade dos documentos, através da separação do acervo em grupos mais

homogêneos. Em [Lian et al. 2004], a clusterização dos documentos também é utilizada, porém com objetivo diferente. Neste caso, a clusterização é realizada através da análise da hierarquia dos termos para minimizar o problema de fragmentação em bancos de dados relacionais.

3. Estratégia Proposta

Como já citado anteriormente, este trabalho tem como objetivo encontrar mecanismos que auxiliem o armazenamento e recuperação mais eficiente de documentos XML heterogêneos em SGBD's XML Nativos, propondo uma estratégia para armazenamento e indexação destes documentos. A estratégia proposta se utiliza das informações obtidas como resultado do processo de KDD sobre a estrutura dos documentos XML e sobre os recursos existentes nos SGBD's XML Nativos de modo a guiar o usuário no desenvolvimento do projeto físico.

Conforme exposto na seção 2, o processo de KDD apresenta três etapas. Inspirada nesta organização, a estratégia proposta também se divide nestas três etapas conforme ilustra a Figura. 1. Estas etapas envolvem desde a captação e tratamento de documentos XML até o armazenamento dos mesmos.

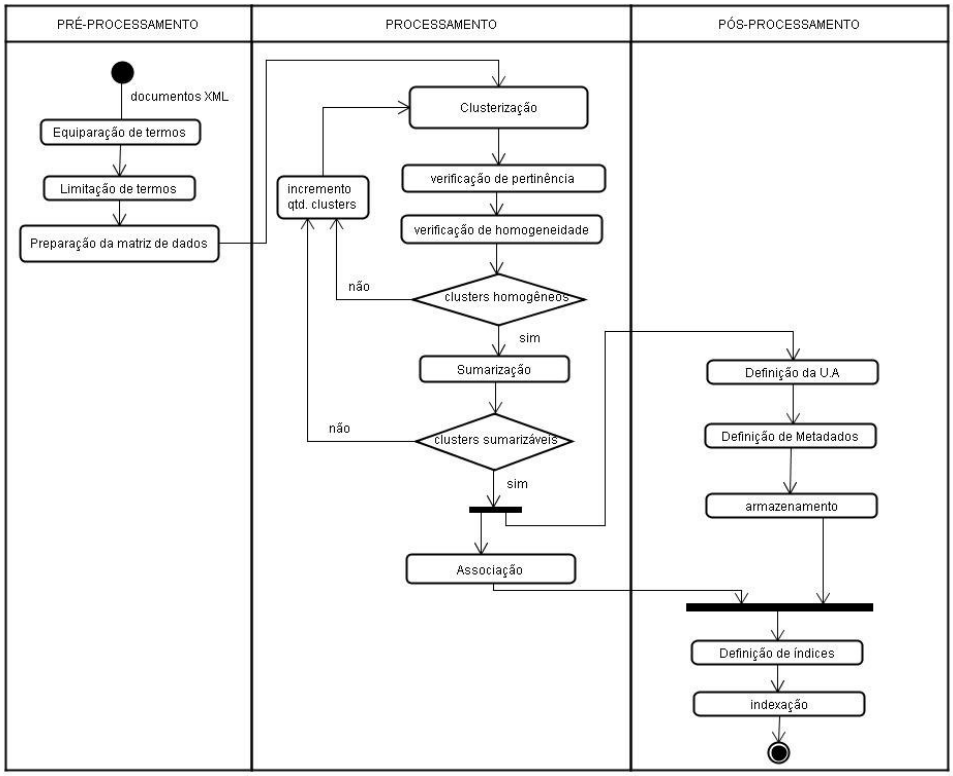

Figura 1. Etapas da Estratégia Proposta

Apesar dos documentos XML poderem estar associados a estruturas que garantam uma certa uniformidade, vale ressaltar que nossa estratégia parte do pressuposto que os documentos provêm de origens distintas, o que nos leva a ter documentos com estruturas heterogêneas.

Para alcançar o objetivo geral proposto neste trabalho, existem dois objetivos específicos que esperamos atingir por meio destas etapas que são: o armazenamento em separado de documentos similares e a criação de índices adequados para agilizar o

futuro acesso a tais documentos. Nas subseções seguintes apresentaremos o detalhamento das etapas que compõem a estratégia proposta.

3.1 Pré-processamento

Na etapa de Pré-processamento foram identificadas três sub-etapas que visam extrair informações sobre os documentos para a etapa central, denominada Processamento. A etapa de Pré-processamento inicia-se a partir da análise da estrutura dos documentos XML. A estrutura destes documentos é composta por etiquetas, denominadas *tags*. Cada etiqueta pode ser vista como um termo que indica ou descreve a natureza da informação por ela "encapsulada". Um dos objetivos deste trabalho é verificar a similaridade entre os documentos que na estratégia proposta, baseia-se na utilização destes termos pelos documentos. Por isso, deste ponto em diante, a palavra termo será utilizada no lugar da palavra "etiqueta", isto é, os termos tratados pelas sub-etapas que compõem a etapa de pré-processamento são provenientes das "etiquetas" dos documentos XML.

Na sub-etapa de Equiparação de termos, inicialmente é preciso identificar todos os termos distintos presentes em todos os documentos. A seguir, no sentido de minimizar a não uniformidade no conjunto de termos utilizados, substitui-se os termos equivalentes por um termo representante, reduzindo a quantidade de termos a tratar. Desta forma, evitamos diferenciar documentos similares só por que utilizam termos diferentes, ignorando que os mesmos têm igual significado. Isto pode ser resolvido usando-se um vocabulário controlado de termos denominado tesauro [UNESCO 1973]. Os algoritmos de *matching* constituem uma alternativa para a integração de termos de diferentes bases de dados de forma automatizada [Shvaiko e Euzenat 2005] .

A sub-etapa de Limitação de termos, consiste em restringir a quantidade de termos a serem analisados. Em KDD a limitação de termos é realizada pela função de Seleção de dados, através da redução de dados vertical, que tem como objetivo eliminar atributos pouco relevantes [Goldschmidt e Passos 2005]. A limitação de termos também pode ser realizada pelas técnicas conhecidas como *startword* e *stopword* [Kelledy e Smeaton 1997] e [Brasil 2004]. Enquanto a técnica de *startword* indica que termos devem ser considerados, a técnica *stopword* indica que termos devem ser ignorados. De forma a facilitar a escolha dos termos para execução de uma destas técnicas, calcula-se a quantidade de vezes que os termos ocorrem nos documentos através de uma atividade denominada ranqueamento de termos.

A sub-etapa de Preparação da matriz de dados consiste na criação de uma matriz de dados apropriada para "clusterização", seguida da normalização dos dados que a compõem. A matriz é composta por valores numéricos associados ao cruzamento de cada termo referenciado por cada documento. Estes valores resultam da análise da presença dos termos na estrutura dos documentos podendo representar a sua ocorrência e/ou frequência. A matriz de ocorrência de termos é composta por 0´s e 1´s, para os casos de ausência ou presença de um termo em um documento. Já a matriz de freqüência de termos informa o número de vezes em que um termo aparece em um documento. Há ainda uma terceira abordagem que pode ser considerada na análise da estrutura do documento XML. Nesta abordagem, leva-se em consideração a hierarquia do termo no documento e pode-se calcular a sua freqüência ou ocorrência. Esta abordagem permite uma maior precisão se tivermos como objetivo identificar documentos estruturalmente e hierarquicamente similares, pois os documentos, além de serem compostos pelas mesmas *tags*, têm que ter estas *tags* aninhadas em uma mesma

estrutura hierárquica para serem considerados similares. Neste caso, a matriz é construída para cada par de termos (pai-filho). No entanto, no caso de documentos heterogêneos e consequentemente com estruturas diferentes, a aplicação direta de uma matriz hierárquica dificultaria identificar documentos similares.

Os valores constantes na matriz que será submetida à sub-etapa de Clusterização deverão ser normalizados. Existem alguns métodos de normalização disponíveis, como documentado em [Goldschmidt e Passos 2005]. Neste trabalho adotamos o método de *normalização pelo valor máximo dos elementos*. Este método consiste em dividir cada valor do termo que esteja sendo normalizado pelo maior valor dentre os valores de tal termo. Ao final da normalização utilizada, todos os valores estarão entre 0 e 1.

3.2 Processamento

Na etapa de Processamento, a idéia é encontrar grupos de documentos XML similares para facilitar sua distribuição por diferentes unidades de armazenamento e ainda identificar possíveis candidatos à indexação.

A sub-etapa de Clusterização é voltada para identificar grupos de documentos similares. A matriz construída na etapa anterior é submetida ao algoritmo de clusterização. A abordagem adotada para clusterização de documentos heterogêneos propõe o uso do algoritmo *fuzzy k-means* [Goldschmidt e Passos 2005]. O algoritmo *fuzzy k-means* é um algoritmo de clusterização que permite que um objeto pertença a mais de um grupo com diferentes graus de pertinência [Rosa 2001]. Utilizando-se desta abordagem, é possível identificar documentos que compartilhem características comuns com documentos de mais de um grupo. Cada grupo identificado sugere uma unidade de armazenamento contendo documentos similares.

Após a clusterização concluída, os documentos apresentam graus de pertinência a todos os *clusters*. Por isto, é na sub-etapa de Verificação de Pertinência que os grupos são definidos de fato. Os graus de pertinência apresentados variam, o documento pode ser mais ou menos pertinente a um *cluster* do que a outro. Neste caso, é preciso definir, previamente, um limiar mínimo de pertinência a ser considerado.

A sub-etapa Verificação de Homogeneidade analisa se os documentos que compõem cada *cluster* formam um grupo homogêneo. Este tipo de análise deve ser realizada, pois nem sempre é conhecida a quantidade de contextos distintos, de onde provêm os documentos. Por isso, não é possível em um primeiro momento determinar a quantidade de clusters adequada. A partir de uma quantidade de clusters inadequada, pode-se resultar em grupos mal formados, como por exemplo um grupo que contenha subgrupos. Assim, mesmo que o usuário tenha uma idéia de quantos clusters devem ser formados, é necessário validar se o cluster é homogêneo. Isso pode ser feito pela análise dos graus de pertinência dos documentos que compõem o cluster. Dado um cluster **C**, se o maior grau de pertinência subtraído do menor grau de pertinência dos documentos que compõem o cluster, for menor ou igual a um limiar de homogeneidade ℓ', previamente definido pelo usuário, então o cluster **C** é homogêneo, senão o cluster **C** é heterogêneo.

No caso de ter sido verificada a heterogeneidade de pelo menos um cluster, volta-se à sub-etapa de Clusterização. A Clusterização deve ser refeita, incrementando-se a quantidade de clusters. Cabe ressaltar que o processo deve ser refeito para todos os documentos e não apenas para os documentos pertencentes ao cluster em que foi

constatada a heterogeneidade. A sub-etapa de Sumarização permitirá descrever o conteúdo de cada *cluster*.

Por fim, a sub-etapa de Associação irá identificar termos freqüentes que poderão ser utilizados para escolha de candidatos a índices. Tal identificação se dá através de duas abordagens: termos independentes (1 a 1 e/ou 2 a 2), e termos hierarquicamente dependentes (pai-filho). No caso da abordagem de análise dos termos independentes é montada uma matriz no formato *basket* [Goldschmidt e Passos 2005]. No contexto deste trabalho, cada documento será identificado como uma transação e cada termo será identificado como um item. Após a matriz no formato *basket* ter sido criada são realizadas duas atividades inspiradas no funcionamento do algoritmo clássico de mineração de regras de associação chamado Apriori [Goldschmidt e Passos 2005]: A formação de conjuntos de 1 elemento, responsável por identificar termos que ocorram frequentemente e a formação de conjuntos de 2 elementos gerados pela combinação dos termos resultantes da atividade anterior (conjuntos de 1 elemento), dois a dois, que atendam a um suporte mínimo, previamente estabelecido pelo usuário. Com isso são obtidas combinações de termos que aparecem de forma conjunta e freqüente nos clusters do acervo de documentos. Tais combinações podem ser consideradas pelo usuário da aplicação como alternativas de índices a serem criados.

3.3 Pós-processamento

Na etapa Pós-processamento, o usuário é guiado na elaboração de um projeto físico de banco de dados com base nas informações geradas nas etapas anteriores somado aos recursos normalmente oferecidos por SGBD's XML Nativos. Nesta etapa são definidas quantas Unidades de Armazenamento (UA's) serão necessárias para o armazenamento dos documentos, de que tipos devem ser e os índices que devem atuar sobre as mesmas.

Na sub-etapa de Definição de Armazenamento, inicialmente define-se o tipo e a quantidade de UA's que serão necessárias para armazenar os clusters criados. A princípio, a quantidade mínima de UA's corresponde a quantidade de clusters estabelecida. A quantidade máxima de UA's depende de outros fatores que vão desde a capacidade de armazenamento de cada UA a estratégias diferenciadas que podem ser aplicadas de acordo com as características dos documentos. A análise da UA quanto à estratégia de armazenamento avalia os recursos que o SGBD XML Nativo oferece. Por exemplo, há SGBD's que orientam que documentos muito grandes sejam armazenados de forma fragmentada, enquanto documentos pequenos sejam armazenados de forma contígua, sem que haja a fragmentação.

Em seguida, define-se quais metadados serão associados aos documentos. Além da referência ao documento original (que nesta estratégia é transformado para permitir a identificação de similaridades), há outras informações que podem ser armazenadas como metadados, tais como: nome do documento que na maioria dos SGBD's XML Nativos já é utilizado, pertinência do documento ao *cluster*, nome de quem criou a unidade de armazenamento, data de criação, tamanho do documento dentre outras informações que podem ser consideradas. Uma vez identificados os metadados, os documentos são então armazenados nas unidades correspondentes.

Uma vez criadas e povoadas as UA's, a partir da sugestão de índices fornecida pela etapa de processamento e das alternativas oferecidas pelo SGBD, procede-se a definição e criação dos índices. Índices unitários (simples) e compostos (sobre dois ou

mais elementos que ocorram de forma associada); índices sobre textos para elementos com textos longos; e índices sobre elementos pouco estáveis, que podem ou não estar presentes nos documentos.

3.4 Considerações sobre a Estratégia Proposta

Os documentos submetidos a esta estratégia são documentos que normalmente não sofrem alterações (e.g. livros, artigos, revistas), nos quais uma alteração gera novas versões destes documentos, e portanto, outros documentos. Com isto, a tendência é que os documentos uma vez armazenados e identificados dentro de unidades, que mantêm documentos similares, não sofram alterações, e assim, as UA's mantenham sua homogeneidade.

Neste estudo, não foi considerada a entrada de novos documentos em UA's já povoadas. A entrada de novos documentos isoladamente ou mesmo em um número considerável, requer uma nova avaliação sobre que UA receberia o(s) novo(s) documentos. De qualquer forma, cada novo documento teria que passar pelas etapas de pré-processamento, processamento e pós-processamento, podendo haver alterações na estratégia. Quanto a isto, outras questões podem ser levantadas, já que o novo documento poderia não ser similar a nenhum dos grupos existentes e com isto surgiria a necessidade de se criar um novo grupo e portanto uma nova UA a fim de manter a homogeneidade e o conjunto de regras aplicadas a esta UA.

A criação de UA's homogêneas somada à criação de índices, já nos dá uma idéia do ganho na recuperação destes documentos. Para avaliar a estratégia proposta será mostrado um estudo de caso que apresenta uma comparação com outras abordagens.

4. Protótipo

Com o objetivo de orientar a construção do projeto físico de banco de dados em SGBD's XML Nativos, com base na estratégia proposta, foi desenvolvido um protótipo, denominado GAIDoX (Guia para Armazenamento e Indexação de Documentos XML) que cobre parcialmente as etapas de pré-processamento, processamento e pós-processamento. A Figura 2 mostra a arquitetura do protótipo.

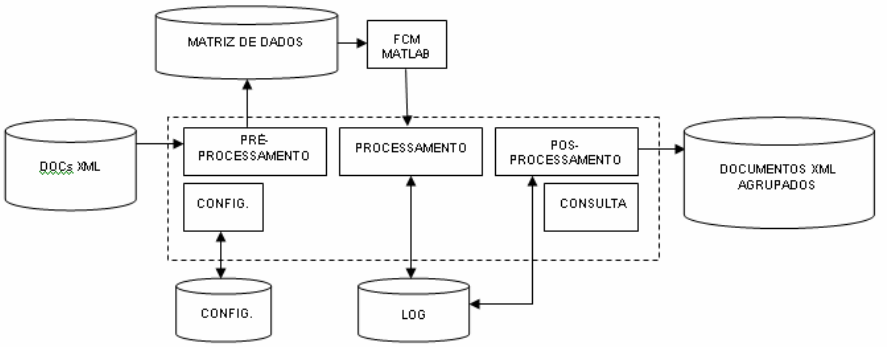

Figura 2. Arquitetura do Protótipo.

A implementação do protótipo foi feita em Java JDK versão 1.5, utilizando a versao 2.2.13 da API fornecida pelo SGBD XML Nativo Berkeley DB XML (BDBXML - http://www.oracle.com/database/berkeley-db.html). Este SGBD é utilizado para armazenar os grupos de documentos XML, após a etapa de processamento.

O módulo de *configuração*, onde são cadastradas informações básicas como a indicação de diretórios onde estão armazenados os documentos, a parametrização de valores (e.g. número de clusters, limiar de homogeneidade, etc.), e informações relacionadas ao SGBD utilizado (e.g. capacidade máxima de cada UA). Estas informações são armazenadas no BD de configuração e são consultadas pelos demais módulos.

O módulo de *pré-processamento* cobre as tarefas de identificação de termos e composição da matriz de dados. Neste módulo ainda é possível limitar a quantidades de *tags* que irão compor a matriz de dados.

No módulo de processamento, a tarefa de clusterização, que utiliza o algoritmo *Fuzzy K-Means* é realizada pelo MatLab (http://www.mathworks.com/), constituindo o módulo FCM MATLAB. Os resultados obtidos são importados para o módulo de processamento e repassados para as tarefas de verificação de pertinência e verificação de homogeneidade. Após constatada a homogeneidade dos clusters é solicitada uma descrição para os mesmos. Este módulo cobre ainda as tarefas de associação cujos resultados são utilizados para a sugestão dos índices.

No módulo de *pós-processamento*, os documentos são armazenados de acordo com os resultados da clusterização obtidos na etapa de processamento e com as estratégias de armazenamento. As estratégias oferecidas pelo BDBXML são: *whole* e *node*. A primeira força o armazenamento do documento íntegro, sem fragmentá-lo. Já a segunda, a critério do SGBD, fragmenta o documento em nós. A Figura 3 mostra o assistente de armazenamento deste módulo.

Além do armazenamento propriamente dito, o módulo de pós-processamento cobre ainda as atividades de definição das UA's, onde é indicada a quantidade de UA's necessárias de acordo com a sua capacidade e o tamanho dos clusters formados; e a atividade de definição de metadados.

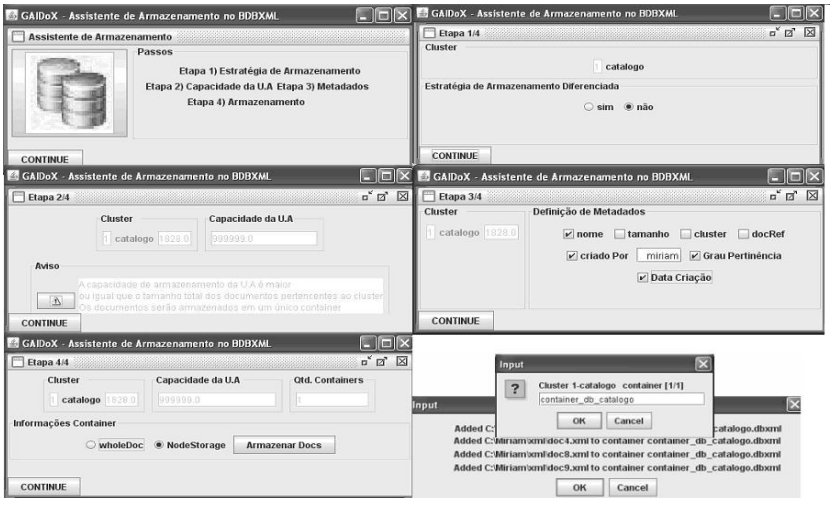

Figura 3. Assistente de Armazenamento.

Os módulos de processamento e pós-processamento mantêm um *log* para armazenar informações sobre as características do armazenamento e decisões do

usuário. Basicamente, as informações do log contêm para cada *cluster*: o tamanho ocupado, a quantidade de UA's necessárias, os *metadados* selecionados, a data de criação e informações sobre a estratégia de armazenamento.

No protótipo há ainda um módulo para realização de consultas sobre os documentos já agrupados e armazenados.

5. Estudo de Caso

Para realizar este estudo de caso, de forma a avaliar a estratégia proposta, foi preciso reunir um grande volume de documentos. Além disso, preocupou-se também com a construção de um acervo heterogêneo, com documentos de diferentes fontes, que variassem significativamente em termos estruturais, porém que guardassem alguma similaridade. Deste modo, foi possível avaliar a etapa de clusterização, permitindo antever a formação dos grupos de documentos similares, e ainda, avaliar as etapas de armazenamento/indexação através de consultas que visam recuperar documentos dos diversos grupos de documentos.

Assim, optou-se por trabalhar com três conjuntos de documentos heterogêneos, sendo que um destes conjuntos, artigos completos, é formado por documentos gerados automaticamente por um utilitário (http://www.cs.toronto.edu/tox/toxgene/). O segundo conjunto, *resumos de artigos*, contém documentos reais, provenientes da revista eletrônica *Sigmod record (http://www.sigmod.org/record/xml/).* O terceiro grupo, currículos, contém documentos criados manualmente. A Tabela 1 totaliza a quantidade de documentos do acervo utilizado neste estudo de caso, discriminando os grupos.

Tabela 1. Acervo de Documentos XML

Tipo	Origem	Qtd	Tamanho
resumos de artigos	sigmod record	911	1,96Mb
artigos completos	Xbench	405	250Mb
curriculos	CV Lattes, Web e outros	15	58,9Kb
	Total	**1331**	**252Mb**

Com relação à etapa de pré-processamento, para os grupos artigos completos e resumo de artigos, nenhum tratamento foi empregado, já que as *tags* que formam estes documentos já apresentavam homogeneidade em sua descrição. Para o grupo de currículos, para os documentos que não seguiam o formato Lattes, foi preciso uniformizar manualmente as *tags* usadas. Após computada a freqüência com que as *tags* apareciam em cada documento, montou-se uma matriz de freqüência de dados para ser submetida ao algoritmo de clusterização.

Uma característica do algoritmo de clusterização é atribuir diferentes graus de pertinência dos documentos a todos os clusters formados, estabelecendo graus de pertinência mais ou menos significativos para cada documento em cada cluster. Conhecendo previamente o acervo, optou-se por utilizar três clusters. O algoritmo de clusterização se comportou corretamente, mantendo os documentos de cada grupo, com altos graus de pertinência, em um mesmo cluster. Assim, foram gerados 3 clusters, um para cada grupo de documentos do acervo. A partir dos grupos definidos, estes documentos foram armazenados em 3 Unidades de Armazenamento (UA), segundo a estratégia proposta, e de duas outras formas (sem considerar a similaridade dos documentos):

- UAs Múltiplas Homogêneas: A estratégia proposta que realiza o armazenamento por UA somente de documentos similares. Nesta estratégia temos uma UA para cada *cluster* formado.
- UA Única Heterogênea: A estratégia que realiza o armazenamento de diferentes documentos em uma única UA.
- UAs Múltiplas Heterogêneas: A estratégia que se utiliza de múltiplas UAs para distribuir os documentos, sendo que esta distribuição é realizada de forma aleatória, sem considerar qualquer tipo de relação entre os documentos.

Para cada uma destas situações foi avaliada a utilização ou não de índices. Os resultados foram obtidos para as três estratégias de armazenamento, e são apresentados e analisados na seção seguinte. Para cada uma destas estratégias, considerou-se somente o modo de armazenamento *node*. O BDBXML não foi capaz de armazenar a quantidade de documentos do acervo preparado, no modo *whole*.

Quatro funcionalidades estão sendo avaliadas pelas consultas selecionadas, são elas: *consultas por valores exatos*, *ordenação*, verificação de *dados irregulares* e *referências e joins*. Na Tabela 2 são listadas as consultas selecionadas, direcionadas a cada grupo de documentos, e os índices empregados em cada uma delas.

Tabela 2. Consultas realizadas nos testes e os respectivos índices usados

Consultas para o grupo Artigos Completos	**Indices utilizados**
1. for $a in collection(“container”)/article[@id=”1”] return $art/prolog/title	id (node-attribute-equality-string)
2. for $prolog in collection("container ")/article/prolog where $prolog/authors/author/name="Ben Yang" return $prolog/title	Name(node-element-equality-string)
3. for $a in collection("container.dbxml")/article/prolog where $a/dateline/country="Canada" order by $a/dateline/date return {$a/title} {$a/dateline/date}	country(node-element-equality-string)
4. collection("container")/article/prolog[not(genre)]/title	Genre(node-element-presence-none)
5. collection("container.dbxml")/article/prolog/genre/title	Genre(node-element-presence-none)
6. for $a in collection("container")/article[@id="7"]/epilog/references/a_id, $b in collection("container")/article where $a = $b/@id return {$b/prolog/title}	id (node-attribute-equality-string)
Consultas para o grupo Resumos de Artigos	**Indices utilizados**
1. for $art in collection("container ")/IndexTermsPage/title[@id="00585"] return $art/title	id (node-attribute-equality-string)
2. for $art in collection("container")/IndexTermsPage where $art/authors/author="Dick Tsur" return $art/author	author(node-element-equality-string)
3. for $a in collection("container")/IndexTermsPage where $a/confyear="1993" order by $a/title return {$a/title}	confyear(node-element-equality-string)
4. collection("container")/IndexTermsPage/categoryAndSubjectDescriptors [not(categoryAndSubjectDescriptorsTuple)]	categoryAndSubjectDescriptorsTuple (node-element-presence-none)
5. collection("container")/IndexTermsPage/categoryAndSubjectDescriptors /categoryAndSubjectDescriptorsTuple/category	categoryAndSubjectDescriptorsTuple (node-element-presence-none)
6. for $a in collection("container")/IndexTermsPage/title[@id="00585"]/authors/author, $b in collection("container")/IndexTermsPage/authors/author where $a = $b return {$b/IndexTermsPage/title}	id (node-attribute-equality-string)
Consultas para o grupo Currículos	**Indices utilizados**
1. for $cur in collection("container") where contains($cur//instituicao,"IME") return $cur//dados_pessoais	instituicao(node-element-substring-string)
2. collection("container ")[//doutorado]//nome	doutorado(node-element-presence-none)
3. collection("container ")[//cidade="Rio de Janeiro"]//experiencia_profissional	cidade(node-element-equality-string)

4. for $tese in collection("container")//tese where contains ($tese/title,"XML") return $tese	title(node-element-substring-string)
5. collection("container")[//instituicao/nome="IME"]	Nome(node-element-equality-string)
6. for $cur in collection("container") where contains ($cur//dados_pessoais/nome,"ana") order by $cur/dados_pessoais/nome return $cur	Dados_pessoais.nome(edge-element-equality-string)

Os tempos das execuções das consultas foram coletados considerando-se como tempo de execução final, a média entre os tempos de execução a quente. Para computação dos tempos foram realizadas 6 (seis) execuções, sendo descartado o tempo da primeira execução. O ambiente utilizado para os testes foi o SGBD BDBXML. Os testes foram realizados utilizando-se um microcomputador com processador Pentium 4.3, memória RAM de 2 Gbytes, disco rígido de 160 Gbytes e sistema operacional Windows XP Professional.

6. Resultados e Discussão

Foram avaliados os tempos de execução das consultas com e sem índice em cada uma das 3 estratégias de armazenamento propostas no estudo de caso: homogêneo múltiplo, heterogêneo único e heterogêneo múltiplo. Algumas consultas foram reescritas, de modo a garantir a utilização correta dos índices, e com isto a melhora do desempenho. Os resultados das execuções sem índice mostraram que a estratégia proposta por este trabalho apresenta os menores tempos de processamento (Santos, 2007). As subseções seguintes discutem os resultados obtidos a partir das execuções com a utilização de índices, apresentando gráficos comparativos entre a estratégia proposta (homogêneo múltiplo) e as demais estratégias (heterogêneo mútiplo e único).

6.1 Resultados das Consultas sobre o Grupo Artigos Completos (Xbench)

Pode-se perceber através da análise dos gráficos da Figura 4, que a estratégia proposta apresenta os menores tempos de processamento na maior parte das consultas. Os ganhos de desempenho são significativamente maiores para as consultas C2, C3. Os índices usados nestas consultas são baseados em igualdade de valores sobre elementos (*tags*), diferentemente dos índices usados nas demais consultas que são índices para dados irregulares (presença) e para atributos. Em relação a estratégia Heterogêneo Múltiplo, exceto nas consultas C1 e C6, os ganhos são sempre significativamente maiores. A consulta C6, que envolve o uso de índices para realizar junção, apresentou, para a estratégia homogêneo múltiplo, piores resultados em relação a estratégia heterogêneo múltiplo.

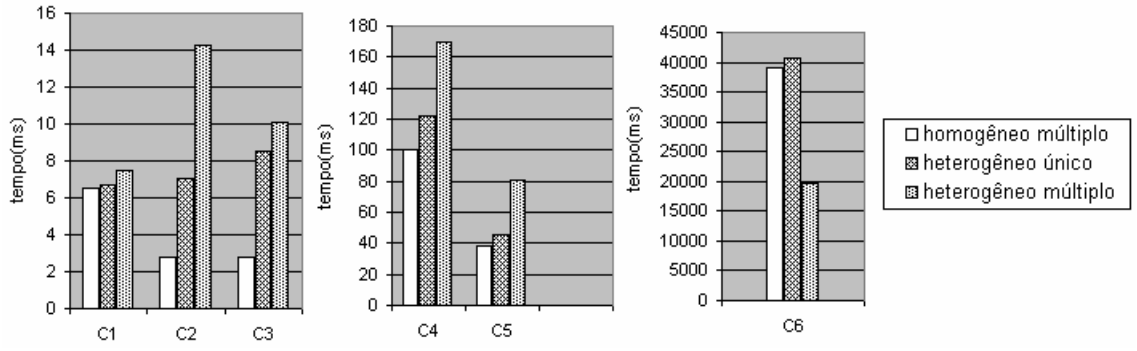

Figura 4. Tempos (em ms) por consultas c/ índices – Grupo Artigos Completos

6.2 Resultados das Consultas sobre o Grupo Resumos de Artigos (Sigmod)

Pode-se perceber através da análise dos gráficos da Figura 5, que a estratégia proposta apresenta os menores tempos de processamento em todas as consultas, exceto na consulta C5 que perde para a estratégia heterogêneo único e não apresenta grandes diferenças em relação à estratégia heterogêneo múltiplo. O índice usado nesta consulta oferece suporte a dados irregulares (presença).

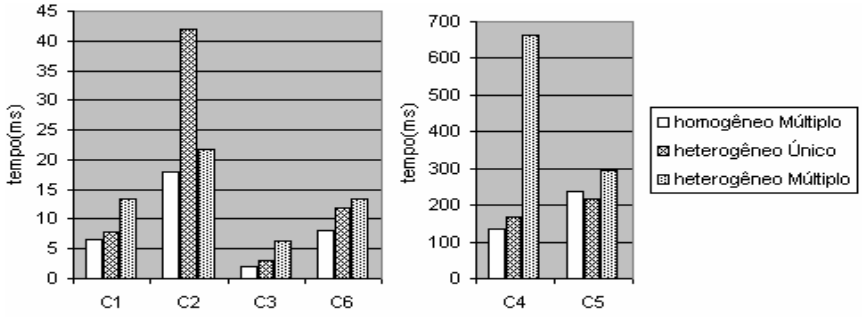

Figura 5. Tempos (em ms) por consulta c/ índice - Grupo Resumos de Artigos

6.3 Resultados das Consultas sobre o Grupo de Curriculos

A Figura 6 não apresenta os tempos para a estratégia heterogêneo múltiplo, pois ocorreram diversos erros no processamento destas consultas sobre este grupo de documentos. Todas as consultas aplicadas a este grupo utilizam-se do recurso de barras duplas, que indica que a *tag* subsequente pode aparecer como um descendente da *tag* precedente. Este recurso foi utilizado, já que há grandes diferenças estruturais neste grupo de documentos.

A consulta C2 apresentou o maior ganho na estratégia proposta, homogêneo múltiplo, em relação às demais estratégias. Esta consulta envolve o uso de índices para dados irregulares (presença). Já a consulta C3 foi a que apresentou tempos de execução mais próximos entre as estratégias homogêneo múltiplo e heterogêneo único. O índice utilizado nesta consulta é baseado em igualdade de valores sobre elementos.

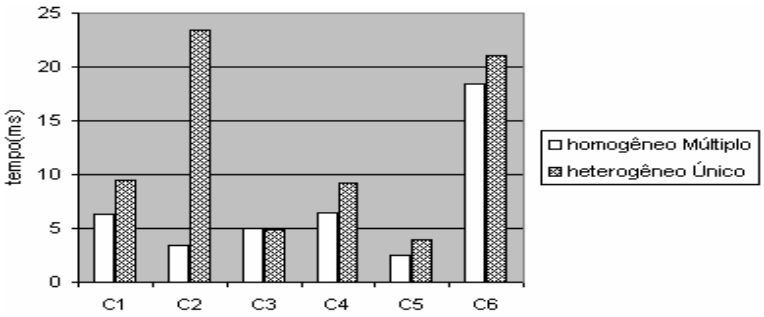

Figura 6. Tempos (em ms) por consulta c/ índice – Grupo de Currículos

6.4 Considerações sobre os Resultados

Durante a realização dos testes, foram encontrados alguns problemas no uso do SGBD BDBXML, o que nos leva a considerar refazer os testes com outros SGBD's XML Nativos O principal problema diz respeito ao uso de índices, pois o comportamento do otimizador de consultas do BDBXML não apresentou um comportamento muito regular. Por este motivo, algumas consultas tiveram que ser reescritas para que o

otimizador reconhecesse os índices. Em outras consultas, percebeu-se que os índices foram utilizados de forma diferenciada pelo otimizador de consulta, ora beneficiando, ora prejudicando o desempenho das consultas, o que tornou a comparação, nestes casos, inconclusiva, levando ao descarte destas consultas.

Apesar de terem sido avaliadas poucas consultas, pode-se perceber que o fato de ter UA's homogêneas, mantendo documentos com características comuns, proporciona algumas vantagens sobre as outras estratégias. Uma delas é saber para onde as consultas serão direcionadas, o que não acontece na forma de armazenamento heterogêneo múltiplo, onde para cada pesquisa, várias UA's precisam ser consultadas. Isto não acontece na forma de armazenamento heterogêneo único, que mantém todos os documentos em uma única UA, mas por outro lado, não se pode aplicar estratégias de armazenamento diferenciadas. A estratégia de armazenamento proposta procura possibilitar um emprego adequado dos recursos de armazenamento e acesso ao tipo de documento sendo armazenado.

Nos três conjuntos de documentos apresentados, foi possível perceber claramente os ganhos, em especial para o grupo de *currículos*. Embora este grupo seja pequeno em termos de número de documentos e espaço exigido para armazenamento, estamos preparando uma base maior para confirmar os resultados obtidos.

Identificamos ainda, outros testes importantes que poderiam ser realizados para avaliar melhor a estratégia proposta, como por exemplo, para avaliar acesso concorrente, ou para avaliar consultas com resultados possíveis em todos os grupos.

7. Conclusão

Como contribuição deste trabalho, destaca-se a especificação de uma estratégia, composta por etapas bem definidas, para auxiliar o armazenamento e recuperação mais eficientes de documentos XML heterogêneos em SGBD's XML Nativos. Além disso, há também o protótipo implementado para auxiliar o usuário a desenvolver um projeto físico de banco de dados sobre SGBD's XML Nativos. Apesar de ter sido avaliado apenas o SGBD BDBXML, vale ressaltar que a estratégia proposta é aplicável a qualquer SGBD XML Nativo, assim como poderiam ter sido aplicados outros algoritmos de mineração correspondentes às tarefas de associação e clusterização utilizados. Para tornar o protótipo independente do SGBD a ser utilizado, precisaríamos realizar pequenas adaptações principalmente na fase de pós-processamento onde o SGBD é acionado em atividades que envolvem a sub-etapa de armazenamento.

Como trabalho futuro podemos citar a geração de esquema único, sobre cada UA gerada pela estratégia proposta. Um outro trabalho interessante seria no sentido de determinar automaticamente a quantidade inicial de clusters. Outra avaliações da estratégia proposta também poderiam ser realizadas, como aplicá-la sobre outros SGBD's XML Nativos, e ou sobre acervos desconhecidos. Por fim, pretende-se também proceder a adequação da estratégia para tratar a inclusão de novos documentos, considerando os grupos já formados.

Referências

Abiteboul, S.; Manolescu, I.; Nguyen, B. e Preda, N. (2005) "A Test Plataform for the INEX heterogeneous track", Springer Berlin, Advances in XML Information Retrieval, v. 3493, p.358-371.

Brasil, Christiane Regina. (2004) "Ferramenta Inteligente de Apoio a Pesquisa: Mineração de Artigos Científicos na WEB", USP, Dissertação.

Brian, Danny. (2006) "The Definitive Guide to Berkeley DB XML", APRESS.

Fayad, U.M.; Piatetsky-Shapiro, G. e Smyth, P. (1996) "From Data Mining to Knowledge Discovery: An Overview. Knowledge Discovery and Data Mining", Menlo Park: AAAI Press.

Flesca, S.; Manco G., Masciari, E.; Pontieri, L. e Pugliese, A. (2005) "Fast Detection of XML Structural Similarity", IEEE.

Goldschmidt, R. e Passos, E. (2005) "Data Mining Um Guia Prático", Campus.

Jagadish,H.V.; Al-Khalifa, S.; Chapman, A. et al. (2002) "TIMBER: A native XML database", Springer Berlin, VLDB Journal, V.11, p.274-291.

Kelledy, F.; Smeaton, A. F. (1997) "Automatic Phrase Recognition and Extraction from Text", Proceedings of the 19th Annual BSC-IRSG Colloquium on IR Research.

Lee, J.; Lee, K. e Kim, W. (2001) "Preparations for Semantics-Based XML Mining", IEEE.

Li, Y.; Yu, C. e Jagadish, H.V. (2004) "Schema-Free Xquery", VLDB.

Lian, Wang; Cheung David; Mamoulis, Nikos e Yiu, Siu-Ming. (2004) "An efficient and scalable algorithm for Clustering XML Documents by Structure", IEEE.

Nierman, A. e Jagadish, H. V. (2002) "Evaluating Structural Similarity in XML Documents", Proc. Int. Workshop on the Web and Databases (WebDB), Madison, WI.

Pereira, J. C. L. e Bax, M. P. (2002) "Introdução a Gestão de Conteúdos", Workshop Brasileiro de Inteligência Competitiva e Gestão do Conhecimento.

Rosa, J.M.C; Tanscheit, R.; Vellasco, M.; Zanini, A; Klein, C.H; Bloch, K.V; Nogueira, A.R; Salis, L.H e Souza e Silva, N.A. (2001) "Aplicação Fuzzy Clustering a Banco de Dados de Amostra Domiciliar da População da Ilha do Governador", SBAI.

Santos, M. O. (2007) "Armazenamento e Recuperação de Documentos XML Heterogêneos: Aplicando Técnicas de KDD para apoiar o projeto físico em SGBD's XML Nativos", IME, Dissertação.

Schöning, H. (2001) "Tamino-A DBMS designed for XML", Int. Conf. on Data Engineering.

Shvaiko, P.;Euzenat, J. (2005) "A Survey of Schema-Based Matching Approaches", Journal on Data Semantics IV, volume 4, p.146-171.

UNESCO. (1973) "Guidelines for the establishment and development of monolingual thesauri". Paris, 37p.

Yang, J.; Cheung, W. e Chen, X. (2005) "Integrating Element and Term Semantics for Similarity-Based XML Document Clustering", Int. Conf. on Web Intelligence.