

## Relacionando documentos XML a um esquema de banco de dados relacional

José Roberto M. Garcia<sup>\*,1</sup>, Dr. Carlos Ho Shih Ning<sup>\*\*,1</sup>

(1) Sistemas de Informação

Laboratório Associado de Computação e Matemática Aplicada  
Instituto Nacional de Pesquisas Espaciais (INPE)

(\*) Mestrado, e-mail: [garcia@cptec.inpe.br](mailto:garcia@cptec.inpe.br), (\*\*) Orientador

### Resumo

Neste artigo descrevemos um processo de conversão entre estruturas de documentos XML e um esquema de banco de dados relacional. Apresentamos as ferramentas que nos permitem realizar consulta e inclusão de documentos XML a partir de um banco de dados bem como o mapeamento entre as colunas das tabelas do banco de dados e os elementos do documento XML. Por fim, destacamos as vantagens e desvantagens desta abordagem.

Palavras-Chave: *xml, middleware, banco de dados*

### Introdução

A quantidade de produtos que fazem uso da tecnologia XML associada ao banco de dados cresce rapidamente a cada dia. O propósito deste trabalho é o de apresentar uma implementação que tipicamente se encaixa na associação destas tecnologias.

A web é um imenso repositório de dados e de informações onde os dados são produzidos, armazenados, transferidos e apresentados por meio de navegadores. Apesar deste aparente caos, onde todos acessam informações disponibilizados a todos, há uma certa ordem. Isto é, o tráfego de informação é limitado a um tipo de protocolo (o http) e a uma porta permanentemente aberta (a porta 80). Qualquer outro tipo de comunicação como telnet ou ftp depende de um acordo pré-estabelecido ou da exigência de autenticação. Ainda assim, pode ser que, dependendo da aplicação, o protocolo empregado pode não ser de conhecimento de todas as partes envolvidas.

Os documentos em formato HTML gozam de facilidades amplas. No entanto, esse formato atrela o conteúdo a uma forma de apresentação destinada ao consumo humano. O XML é um retorno às origens, onde a informação deve ser propriamente armazenada. O XML é basicamente uma linguagem de marcação similar ao HTML onde a organização da sua estrutura é estabelecida conforme o tipo de aplicação a que se destina [3]. A apresentação desse conteúdo deve ser feita quando sabemos a quem se destina seja ele um usuário ou um programa.

A utilização de banco de dados se faz cada vez mais presente. O tipo mais comum de banco de dados é o relacional. Neste modelo os dados são dispostos tabularmente e fracionados por meio da normalização. A primeira vista tal estrutura não é compatível para armazenar dados organizados em documentos hierárquicos com os do XML. Exploramos um mapeamento que permite compatibilizar as duas organizações, uma que é voltada ao documento ("*document centric*") e a outra que é voltada ao dado ("*data centric*").

### Visão global do cenário

A figura 1 mostra como o cenário é visto de forma geral. O funcionamento da aplicação ocorre da seguinte maneira: a ferramenta middleware é iniciada (1) e faz a conexão com o banco de dados (2) através de uma ponte JDBC:ODBC. Uma vez conectado, a aplicação modela as informações do arquivo XML (5) como uma árvore de objetos, através de uma classe Java específica (3). O arquivo de mapeamento (4) é, então, lido e as associações coluna\_da\_tabela x atributos\_do\_objeto concretizadas para a execução do comando de inserção ou consulta no banco.

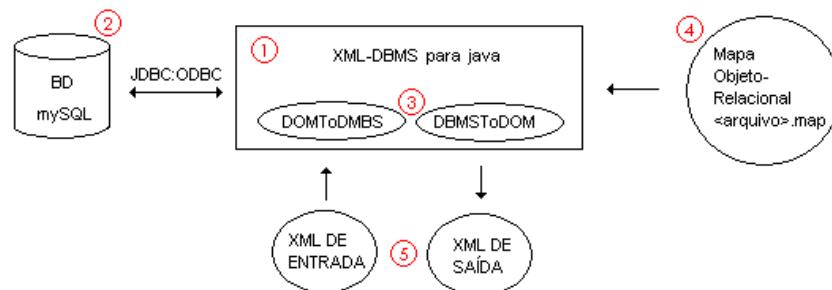


Figura 1

## Ferramentas

A seguir são detalhadas todas as ferramentas utilizadas para a realização deste trabalho. Por ser um trabalho elaborado com o intuito primeiro de adquirir conhecimento tecnológico optamos por utilizar apenas ferramentas da categoria freeware.

XML-DBMS for Java [1], versão 1.01: são classes Java que atuam como middleware para transferência de dados entre documentos XML e a base de dados relacional. Elas visualizam os documentos XML como uma árvore de objetos e mapeiam estes objetos no banco de dados, relacionando atributos e colunas. Este produto foi a base de nosso trabalho e tem Ronald Bourret como seu autor principal. É um produto categorizado como “model-driven” pois o documento XML é modelado a partir de um modelo pré-definido.

Java Development Kit 1.2.1: é o interpretador Java necessário para rodar a aplicação que realiza a conexão com o banco de dados e dispara as rotinas de recuperação e inserção no banco. Seu uso será mais detalhado na seção que trata sobre como os testes foram realizados.

MySQL versão 3.23.39-nt: que é o banco de dados relacional escolhido. É um banco de dados para windows, de simples instalação e uso, mas atende plenamente os objetivos do projeto. Há um MER - Modelo de Entidade e Relacionamento desenvolvido para o trabalho que será detalhado mais adiante.

myODBC versão 2.5: driver para windows para conexão com bancos de dados mySQL. Deve ser criado um DSN (Data Source Name) que acesse o banco criado mySQL criado. Este nome também deve constar nas classes Java utilizadas de modo que referenciem a conexão correta.

XERCES: Java parser (analisador sintático) que verifica se o documento XML está em conformidade com suas regras de criação e que contém suporte a especificação API DOM nível 1 for Java da W3C [2]. Este produto é um conjunto de classes Java que atuam analisando o arquivo XML e sendo intermediárias entre os documentos XML de input e output, e a aplicação Java. A API DOM (Document Object Model) citada serve para representar as tags XML em formato de uma árvore de objetos.

## Esquema de Banco de Dados

O esquema de banco de dados selecionado para demonstrar nosso trabalho (Figura 2) engloba relacionamentos N:N, como é o caso de PUBLICACAO x AUTORES e PALAVRAS\_CHAVE, e 1:N como é o caso de PUBLICACAO x RELATORIO.

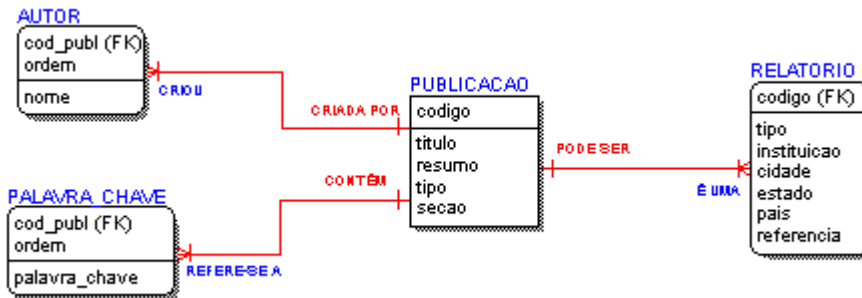


Figura 2

## Documentos XML

O documento XML mostrado a seguir pode ser utilizado para entrada de dados no banco ou como sendo a saída de uma consulta.

```
<?xml version="1.0" encoding="UTF-8"?>
<publicacoes>
  <publicação id="RT-2" tipo="RT">
    <titulo> Relacionando documentos XML... </titulo>
    <resumo> Este artigo descreve uma forma ...</resumo>
    <relatorio id="RT-2">
      <secao>Secao xxxxxxxx</secao>
    </relatorio>
  </publicação>
  <relatório id="RT-2" tipo="RT">
    <instituicao> LAC / INPE </instituicao>
    <cidade> SJC / SP <cidade>
    <estado> SP </estado>
    <pais> Brasil </pais>
    <referencia> RW45-XSRPP </referencia>
  </relatório>
  <autor id="RT-2">
    <nome> Jose Roberto M. Garcia </nome>
  </autor>
  <autor id="RT-2">
    <nome> Carlos Ho Shih Ning </nome>
  </autor>
  <palavra_chave id="RT-2">
    <palavra>xml</palavra>
  </palavra_chave>
  <palavra_chave id="RT-2">
    <palavra>middleware</palavra>
  </palavra_chave>
</publicações>
```

## Mapeamento

Este tipo de mapeamento é realizado através de um documento texto que possui o relacionamento entre os elementos do documento XML, transformados em objetos e seus atributos através da API DOM, e as tabelas e suas colunas do banco de dados. Este mapeamento se baseia no relacionamento entre uma estrutura XML, um objeto e uma tabela, por isso é denominado mapeamento objeto-relacional. A figura 3 mostra estas relações.

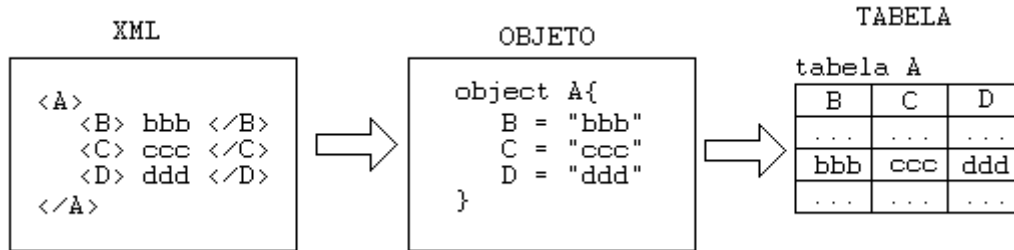


Figura 3

Sendo assim os elementos pais do documento XML correspondem a tabelas no banco de dados. Os filhos correspondem às colunas das tabelas. Um elemento complexo, ou seja, que possua uma outra estrutura de dados como conteúdo, é mapeado como outra tabela, com uma chave estrangeira para ela.

## Considerações Finais

Optamos pelo uso de um banco de dados relacional, pois este modelo é o mais comum e está bastante maduro. Além disso, não podemos desprezar os dados existentes ou que continuam a serem produzidos. Apesar do cunho acadêmico, pretendemos colocar o sistema em operação.

Este trabalho ainda se encontra em desenvolvimento e foram produzidos resultados parciais. As seguintes tarefas estão por realizar:

- Validação dos arquivos XML contra uma DTD – Document Type Definition
- Novas alternativas de buscas levando-se em conta outros campos como autor, título, palavras-chave, período de publicação, palavras no corpo do documento, por categoria (tese, artigo, etc).
- Códigos das publicações está sendo gerado manualmente, quando deveria ser automático.
- Alteração ou correção das informações armazenadas.

## Referências

- [1] Bourret, R. Middleware for Transferring Data between XML Documents and Relational Databases, <http://www.rpbouret.com/xmldbms/index.html>.
- [2] Chang, B.; Scardina, M.; Karun, K.; Kiritzov, S.; Macky, I.; Novoselsky, A.; Ramakrishnan, N. *Oracle XML – O manual oficial*, Campus, Rio de Janeiro, 2001.
- [3] Furgeri, S. *Ensino Didático da Linguagem XML*, Érica, São Paulo, 2001.