

# Servidor de Documentos XML Usando Java

Werley Ribeiro Martins, Dilvan de Abreu Moreira

Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo (USP)

Caixa Postal 668 – 13560-970 – São Carlos – SP – Brasil

{wmartins,dilvan}@icmc.sc.usp.br

**Abstract.** *The goal of this project was to develop a server capable of doing many kinds of operations (such as adding, querying, deleting and transforming) over a repository of XML (Extensible Markup Language) and XSL (Extensible Stylesheet Language) documents. These documents are represented using Java objects conforming to the DOM (Document Object Model) interface model, a standard model used in many Internet applications. The server uses the XQL (XML Query Language) as query language. It also allows the conversion of XML documents into other languages using the XSL language.*

**Resumo.** *O objetivo desse trabalho é desenvolver um servidor capaz de realizar diversas operações (como adicionar, consultar, apagar e transformar) sobre um repositório de documentos XML (Extensible Markup Language) e XSL (Extensible Stylesheet Language). Esses documentos são representados na forma de objetos Java que obedecem ao modelo de interfaces DOM (Document Object Model), um modelo padrão usado em várias aplicações da Internet. O servidor utiliza a linguagem XQL (XML Query Language) na realização de consultas. Além disso, ele permite, através da linguagem XSL, a transformação de documentos XML para outras linguagens.*

## Resumo

Hoje em dia, a linguagem XML (*Extensible Markup Language*) vem sendo muito utilizada em diversas aplicações tanto no meio acadêmico quanto no meio comercial [Bray et al., 2000]. Porém, a disponibilidade de uma grande quantidade de dados XML apresenta vários problemas práticos que o padrão XML não discute, por exemplo, como os dados XML podem ser extraídos de grandes documentos XML.

Apesar de existir uma linguagem projetada especificamente para consultar qualquer tipo de dado XML, caso da linguagem XQL (*XML Query Language*) [Robie et al., 1998], e outra específica para realizar transformações em documentos XML, caso da linguagem XSL (*Extensible Stylesheet Language*) [Adler et al., 1997], existe dificuldade em conseguir implementações abertas de servidores de dados capazes de usar essas linguagens.

O objetivo desse trabalho é desenvolver um servidor XML capaz de realizar diversas operações, sobre um repositório de documentos XML e XSL, tais como adicionar, apagar, consultar e transformar. O servidor permite incluir novos documentos XML no repositório de duas formas: como um documento a mais no repositório ou dentro de outro documento XML já existente. No segundo caso, uma *query* definida na linguagem XQL indica em que ponto dentro do documento existente o novo documento é adicionado. Também é possível que um documento XML seja apagado por inteiro ou por partes. Para isso, a parte do documento que será apagada é declarada através de uma *query* em XQL. Ele permite ainda receber consultas nas linguagens XQL ou XSL, retornando conteúdos em várias formas, tais como:

- um documento XML do repositório;
- parte de um documento XML, tornando-se um novo documento XML;
- um novo documento em outra linguagem diferente de XML, se a consulta for feita pela linguagem de transformação XSL. Assim, a linguagem desse novo documento é dependente da transformação feita pela XSL;
- um novo documento XML gerado através da XSL.

Esses documentos são representados na forma de objetos Java que obedecem ao modelo de interfaces DOM (*Document Object Model*), um modelo padrão usado em várias aplicações da Internet [Apparao et al., 1998]. DOM é capaz de representar qualquer documento XML bem formado em forma de uma árvore de nós e tratar a informação armazenada nesses documentos como um modelo de objetos hierárquicos.

Para o armazenamento é utilizado o serviço *JavaSpaces*, que pode ser usado como um repositório onde objetos Java são armazenados de forma persistente [Freeman et al., 1999]. *JavaSpaces* armazena os documentos XML na forma de objetos DOM, que é a forma na qual são realizadas as manipulações sobre esses documentos. Isto é possível, pois a linguagem Java possui uma série de interfaces definidas de acordo com a recomendação DOM. Sendo assim, como um objeto DOM é um objeto Java, ele pode ser armazenado no serviço *JavaSpaces* e, através dele, representar todos os documentos XML que são armazenados no servidor.

Além disso, para a realização das operações, são utilizadas bibliotecas que disponibilizam os serviços necessários para atender as solicitações propostas para o desenvolvimento do servidor XML. As bibliotecas e suas respectivas versões são:

- Xerces - versão 1.2.2 - *parser* XML, escrito em Java, utilizado para converter documentos XML em objetos DOM, que a forma como esses documentos são armazenados e manipulados dentro do repositório [XERCES, 2000];
- Xalan - versão 1.2.2 - processador XSL, escrito em Java, utilizado para transformar documentos XML em documentos HTML, texto ou em outros documentos XML [XALAN, 2000];
- GMD-IPSI XQL - versão 1.0.2 - ferramenta de consulta e armazenamento, escrita em Java, que utiliza a linguagem XQL para consultar dados de documentos XML, podendo ser aplicadas em qualquer nó de uma árvore DOM [GMD-IPSI, 2000].

Para o desenvolvimento do servidor XML, foi construída uma interface Java implementada por um *proxy*, escrito em Java, através do qual é feita a conexão entre as aplicações cliente e o servidor XML. Nessa interface, estão definidos os métodos que são executados pelo servidor XML, sobre os documentos armazenados no espaço *JavaSpaces*, para a realização dos serviços solicitados pelas aplicações clientes.

## Referências Bibliográficas

- [Adler et al., 1997] Adler, S., Berglund, A., Clark, J., Cseri, I., Grosso, P., Marsh, J., Nicol, G., Paoli, J., Schach, D., Thompson, H. S. and Wilson, C. (1997) “A Proposal for XSL”, <http://www.w3.org/TR/NOTE-XSL.html>, August.
- [Apparao et al., 1998] Apparao V., Byrne, S., Champion, M., Isaacs, S., Jacobs, I., Le Hors, A., Nicol, G., Robie, J., Sutor, R., Wilson, C. and Wood, L. (1998) “Document Object Model (DOM) Level 1 Specification”, <http://www.w3.org/TR/REC-DOM-Level-1/>, October.
- [Bray et al., 2000] Bray, T., Paoli, J., Sperberg-McQueen, C. M. and Maler, E. (2000) “Extensible Markup Language (XML) 1.0 (Second Edition)”, <http://www.w3.org/TR/REC-xml>, October.
- [Freeman et al., 1999] Freeman, E., Hupfer, S and Arnold, K., *JavaSpaces Principles, Patterns, and Practice*, Addison Wesley, 1999.
- [GMD-IPSI, 2000] “GMD-IPSI XQL Engine: Version 1.0.2”. <http://xml.darmstadt.gmd.de/xql/>, November 2000.
- [Robie et al., 1998] Robie, J., Lapp, J., Schach, D., Hyman, M. and Marsh, J. (1998) “XML Query Language (XQL)”, <http://www.w3.org/TandS/QL/QL98/pp/xql.html>, September.
- [XALAN, 2000] “Xalan-Java Overview”. <http://xml.apache.org/xalan-j/overview.html>, November 2000.
- [XERCES, 2000] “Xerces Java Apache Readme”. <http://xml.apache.org/xerces-j/index.html>, November 2000.