Um Modelo Para Conversão de Documentos de Texto Baseados em XML

Caetano Sauer, Nicolas B. Maillard, Philippe O. A. Navaux
Instituto de Informática - Universidade Federal do Rio Grande do Sul
Caixa Postal 15064, CEP 91501-970. Porto Alegre, Brasil.
{csauer,nicolas,navaux}@inf.ufrgs.br

Resumo

XML-based file formats are becoming very popular nowadays, as they have plenty of advantages over pure binary data. Most of this advantages appears because they follow an extensible, well-defined and human readable markup language grammar. This paper will briefly analyse the pros and cons of using XML as a format for word processing documents. Also, it proposes a model for converting any structured text document to a XML-based format, with focus on the Microsoft Office OpenXML standard. The model is based on the interaction between two application modules: a parser for the original format and a library that generates XML code through the definition of generic components of a text document.

1. Formatos de arquivo baseados em XML

Linguagens de *markup* têm como característica o fato de combinar um determinado texto com informações adicionais sobre ele. Tais informações podem, por exemplo, carregar uma estrutura de apresentação, ou ainda associar semântica a um determinado dado. A XML (Extensible Markup Language) é uma linguagem de *markup* de propósitos gerais, cujo objetivo é servir de base para a definição de outras linguagens específicas, onde a informação carregada pelo texto ganha contexto e significado. A classificação de extensível surge justamente disto, usuários podem definir suas próprias *tags* (palavras-chave que carregam meta-dados) e, a partir daí construir outras linguagens como, por exemplo, XHTML, RSS, SVG, MathML, entre outras. A definição formal de tais linguagens, juntamente com a associação de semântica, pode levar também à criação de novos tipos de arquivo, com aplicações diversas. Com isto, o arquivo formatado sobre tal linguagem herda todas as características de um arquivo XML, sendo as mais notáveis a legibilidade pelo ser humano e a portabilidade.

Além de ser legível tanto pela máquina quanto pelo ser humano e de ser totalmente portável, servindo como método para compartilhar dados entre diferentes sistemas, a linguagem descrita sobre XML tem outras vantagens. Dentre elas, temos como mais notáveis o suporte a Unicode, permitindo que documentos sejam escritos em qualquer língua, a estrutura hierárquica, adequada para representar grande parte dos tipos de arquivos e estruturas de dados, e a representação através de arquivos de texto, muito menos restritos que formatos binários proprietários.

Porém, existem também algumas desvantagens, e quase todas elas surgem devido à grande redundância da sintaxe e à repetitividade de informação que surge com o uso de *tags*. Tais características implicam em arquivos grandes e, consequentemente, em custos mais altos de processamento, transmissão e armazenamento. Além disso, o modelo hierárquico é limitado se comparado a um modelo relacional ou baseado em grafos.

Entretanto, por serem arquivos de texto, documentos XML têm o seu tamanho consideravelmente reduzido quando processados por um algoritmo de compressão, e o modelo hierárquico, apesar de limitado, é suficiente para representar grande parte dos dados manipulados no mundo real.

1.1. Documentos de texto e o OpenXML

Partindo da idéia de associar informações a um determinado texto, a idéia de representar textos formatados em XML surge naturalmente. Basta que os meta-dados carreguem a formatação de cada bloco de texto. Tal idéia é utilizada na HTML, linguagem que, assim como a XML, é derivada da SGML (Standard Generalized Markup Language). Em HTML, boa parte das *tags* de *markup* definem a formatação associada ao conteúdo do documento.

Entretanto, uma linguagem mais poderosa faz-se necessária quando os documentos são gerados por processadores de texto, que possuem, além de uma formatação mais completa, estruturas auxiliares como folhas de estilo, macros e tabelas de fontes. É o caso do Office OpenXML, pro-

posto pela Microsoft para a sua suíte de produtividade Office 2007. Na norma OpenXML, descrita formal e detalhadamente em [1], a linguagem WordprocessingML é utilizada para a especificação de documentos de texto. Abaixo, um exemplo de um trecho de texto codificado em WordprocessingML.

```
<w:p>
   <w:r>
      <w:rPr>
         <w:b />
      </w:rPr>
      <w:t>
         Hello World!
      </w:t>
   </w:r>
   <w:r>
      <w:t>
         This is a text document.
      </w:t>
   </w:r>
</w:p>
```

Figura 1. Exemplo de parágrafo contendo duas linhas, sendo a primeira formatada em negrito.

2. Gerando documentos OpenXML

Para gerar documentos OpenXML, mais especificamente documentos de texto em WordprocessingML, o modelo proposto neste artigo prevê o uso de uma biblioteca que permite definir uma estrutura de árvore, representando a hierarquia de *tags* de um arquivo XML. Cada documento possui um único nodo raiz, e cada nodo deve possuir um nome e opcionalmente ter associado a ele atributos com seus respectivos valores e um texto a ser posicionado dentro das *tags*.
Utilizando esta biblioteca, constrói-se um aplicativo contendo uma série de classes para abstrair as estruturas de um documento de texto, tais como: parágrafo, seção, cabeçalho, nota de rodapé, etc. Cada objeto construído a partir de uma dessas estruturas deve possuir métodos para definir suas respectivas propriedades. Um objeto parágrafo deve, por exemplo, carregar a informação de espaçamento das linhas. Além disso, os objetos devem ter associados a eles os respectivos nodos da hierarquia de *tags*.
Por fim, a biblioteca deve ser capaz de organizar os arquivos em um pacote ZIP, que forma o arquivo do documento de texto conforme descrito em [1]. Existem bibliotecas específicas para a geração de documentos OpenXML, algumas delas ([2] e [3]) foram utilizadas como referência para este modelo.

3. Parsing de outros formatos e conversão

O modelo especificado na seção anterior define uma biblioteca para geração de documentos OpenXML. Essa biblioteca, se utilizada em conjunto com um parser, forma um conversor para OpenXML cujos arquivos de entrada possuem o formato reconhecido pelo parser. Este último deve ser capaz de identificar as estruturas sobre as quais pode-se construir objetos da biblioteca, ou seja, deve reconhecer parágrafos, citações, imagens, cabeçalhos, entre outros. A partir daí, o parser deve criar a estrutura em árvore do documento, gerando os arquivos XML e finalmente o pacote correspondente ao documento OpenXML.
A norma RTF, descrita em [4], possui uma associação direta com o modelo descrito acima, pois a estruturação de um documento Rich Text Format é feita a partir de um modelo hierárquico. O projeto hospedado em [5] visa a construção de um conversor utilizando um parser RTF e uma implementação em C++ do modelo de conversão.

4. Conclusões

Este artigo fez uma breve análise sobre o uso da XML como linguagem base para a definição de tipos de arquivo, com ênfase em documentos de texto. Foram apresentados o formato Office OpenXML e um modelo de conversão para este formato a partir de outros tipos de texto formatado.

Referências

[1] ECMA International. *Office OpenXML File Formats*, December 2006.

[2] C. Julien. Openxml4j - office open xml api for java. `http://www.openxml4j.org/`.

[3] Microsoft Corporation. *Microsoft MSDN Library*, January 2007. `http://msdn2.microsoft.com/en-us/library/default.aspx`.

[4] Microsoft Corporation. *Word 2007: Rich Text Format (RTF) Specification, version 1.9*, 2007.

[5] C. Sauer. openxmlconv - conversor de textos fotmatados para openxml. `http://www.codeplex.com/openxmlconv`.