

Elaboração de Schema e Stylesheet para composição de um Jornal Electrónico produzido em XML

Relatório do trabalho prático da disciplina de Processamento
Estruturado de Documentos

Mestrado em Informática – Ano lectivo 2002/2003

Autor: Alexandre Sérgio Mano – MI6076

Professor: José Carlos Ramalho

Braga, Julho de 2003

Índice

1. Introdução	1
2. Desenvolvimento do projecto	1
2.1. O schema.....	1
2.1. Tipos complexos e simples	3
2.1.1. Tentidade.....	3
2.1.2. Tlocal	4
2.1.3. Tlink.....	4
2.1.4. Timagem	4
2.1.5. Tparagrafo	5
2.2. A folha de estilo	5
2.2.1. Templates principais	6
2.2.2. Links para as secções, artigos e índice remissivo	6
2.2.3. Tratamento dos artigos.....	7
2.2.4. Tratamento dos classificados	8
2.2.5. Elementos opcionais	8
3. Conclusões	8
4. Bibliografia	9

1. Introdução

O objectivo deste trabalho é criar um Schema e uma StyleSheet capazes de representar semanticamente e tratar uma instância XML para a criação de um jornal electrónico.

Para o efeito, foram produzidos os ficheiros jornal.xsd e jornal.xslt, que contém, respectivamente, o schema e a stylesheet para o jornal electrónico. Adicionalmente, foi criada uma instância (jornal.xml) como exemplo de funcionamento do jornal e são disponibilizadas as imagens contidas na instância do jornal.

Neste relatório serão explicadas as escolhas e procedimentos adoptados na criação dos ficheiros descritos acima, em conjunto com uma visão geral do projecto desenvolvido.

2. Desenvolvimento do projecto

2.1. O schema

Para a criação do schema foram consultados alguns exemplos de jornais electrónicos (em especial o Público) e, com a experiência adquirida nas aulas, foi construído o diagrama que dá suporte ao jornal electrónico.

Foi também assumido como objectivo deste projecto, sendo ele puramente académico, que não seriam acrescentados elementos ao schema do jornal que não tivessem qualquer importância prática na criação do jornal. Assim, procurou-se atingir um schema simples, mas completo, que derivasse também num ficheiro XML facilmente criável e manipulável por uma grande gama de utilizadores.

É de referir que todos os elementos que compõem o schema estão anotados, pelo que a repetição destas anotações será evitada neste relatório, onde será exposta uma visão geral do schema e as escolhas que o guiaram.

Comecemos pelo elemento principal do jornal:

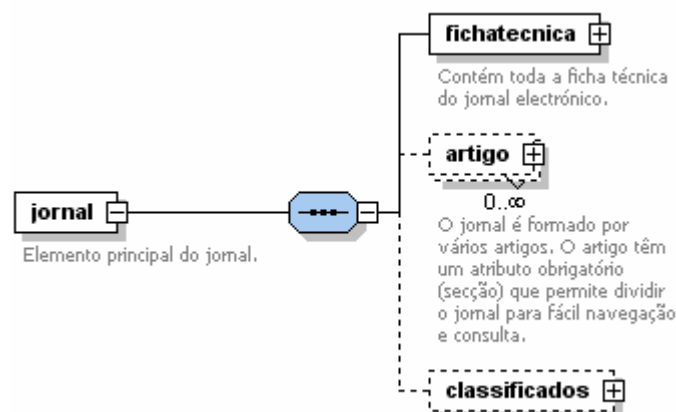


fig.1 – Representação do elemento jornal.

Portanto, o jornal é composto por uma ficha técnica, que descreve os aspectos técnicos relacionados com o jornal, um conjunto de artigos e uma secção de classificados.

Surge logo aqui uma primeira escolha bastante importante que foi necessário assumir: os artigos não estão organizados em secções, como num jornal tradicional. A secção de um artigo é um atributo obrigatório deste elemento. Esta escolha foi assumida tendo em conta diversos factores, como:

- num jornal electrónico, os artigos (ou as ligações para eles) não aparecem normalmente organizados em secções, mas por ordem de publicação;
- uma folha de estilo pode tratar artigos localizados em diversas instâncias XML e integrá-los no jornal sem qualquer dificuldade;
- os artigos podem ser inseridos na instância XML por qualquer ordem, e não é necessário verificar se a secção correspondente já existe. Isto facilita a escrita da instância e evita erros.

No entanto, no ficheiro html de output, os artigos aparecem divididos em secções. Seria, no entanto, extremamente simples, e sem modificar o schema, mostrá-los ordenados por data, autor ou qualquer outro elemento, se tal fosse necessário.

Vejamos agora a ficha técnica do jornal.

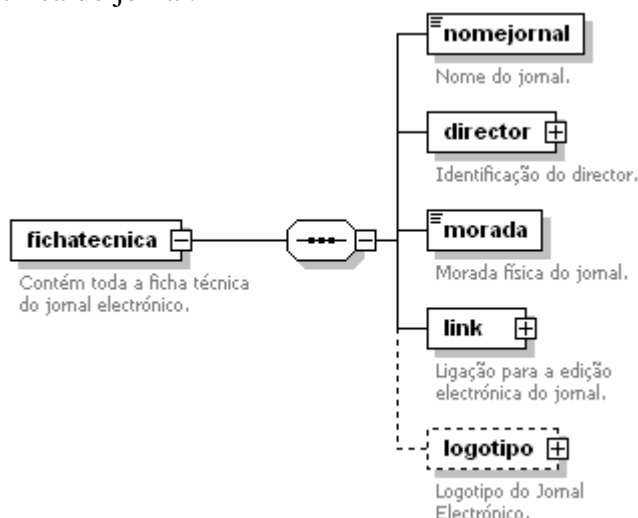


fig.2 – Ficha técnica do jornal.

Também os classificados:

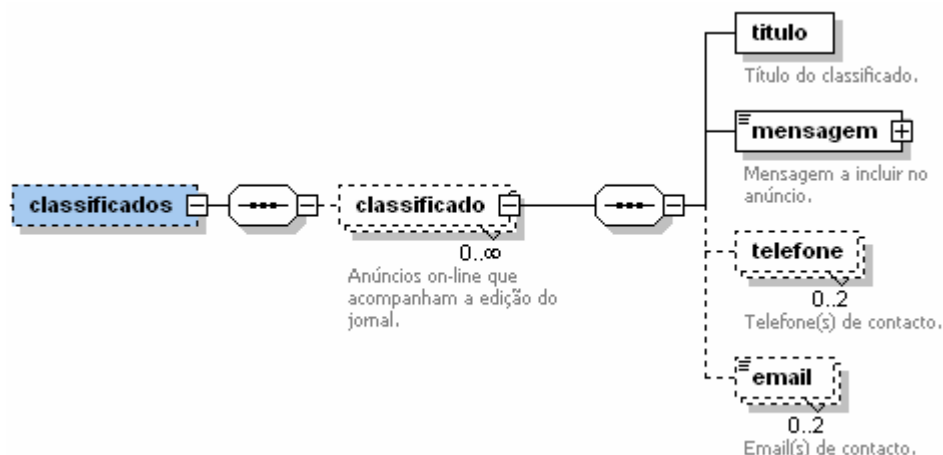


fig.3 – Representação dos classificados.

E por fim, os artigos:

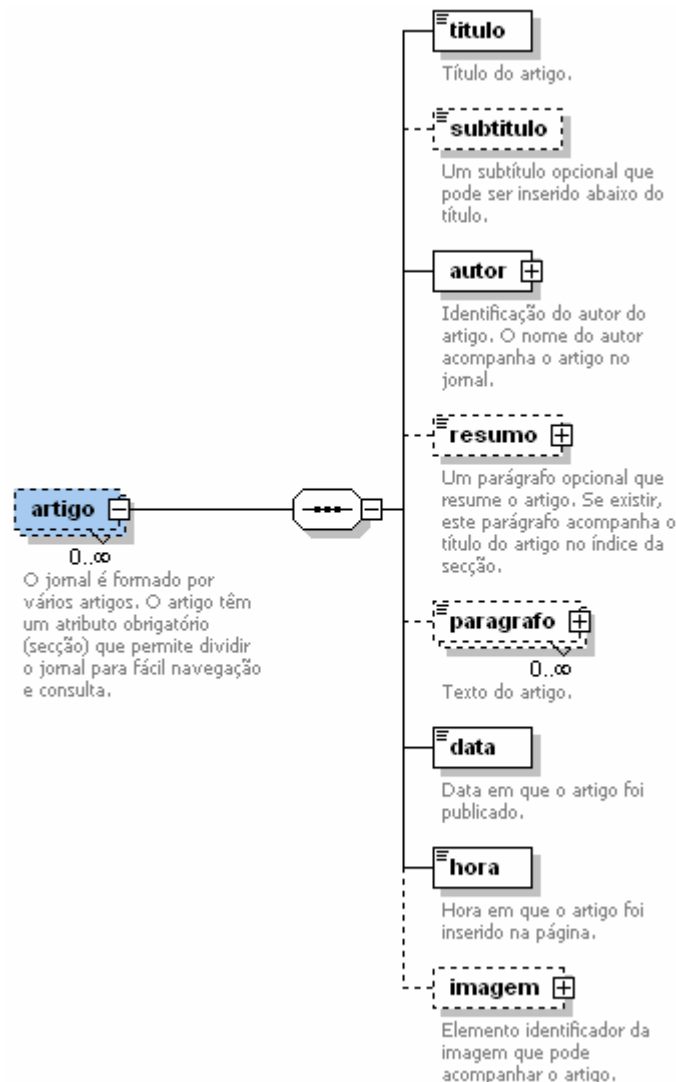


fig. 4 – Representação dos artigos no jornal.

Como a definição dos elementos dentro dos elementos ficha técnica, artigo e classificados é feita à custa de tipos complexos, eles serão apresentados mais abaixo.

2.1. Tipos complexos e simples

2.1.1. Tentidade

Este tipo complexo permite definir todas as entidades do jornal, desde o director até aquelas que aparecem nos artigos. As entidades dos artigos são depois organizadas num índice remissivo no final do jornal.

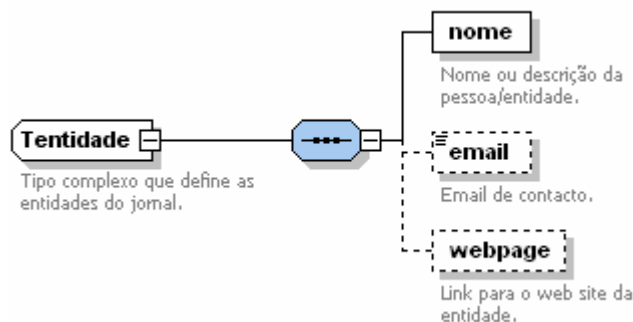


fig. 5 – Caracterização do tipo Tentidade.

2.1.2. Tlocal

Este é um tipo simples que permite identificar os locais que aparecem nos artigos e que o redactor do jornal pretende ver mencionados no índice remissivo.

2.1.3. Tlink

O tipo complexo Tlink permite gerir de maneira uniforme os links que aparecem no jornal.

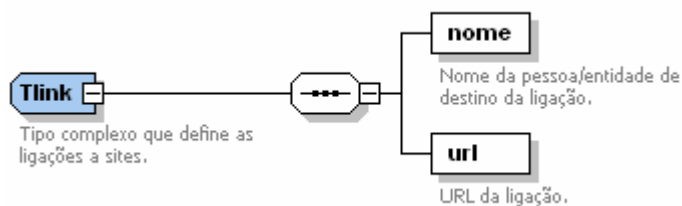


fig. 6 – Caracterização do tipo Tlink.

2.1.4. Timagem

Tal como o tipo anterior, permite gerir de maneira uniforme as imagens que aparecem nos artigos. Nos artigos, o elemento imagem possui um atributo opcional (pos) que permite definir se a imagem deve aparecer à esquerda (valor 'esq') ou à direita (valor 'dir') do texto do artigo. Se o atributo não estiver definido numa imagem, ela aparece à direita do texto.

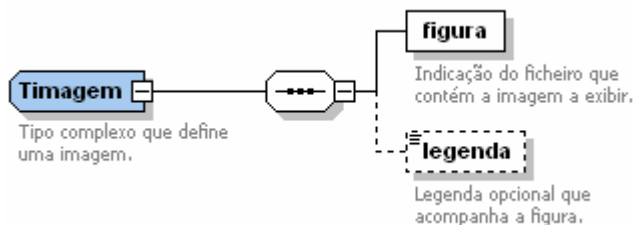


fig. 7 – Caracterização do tipo Timagem.

2.1.5. Tparagrafo

Este tipo complexo permite representar todos os possíveis conteúdos dentro de um parágrafo. Alguns conteúdos, como o negrito e itálico são triviais; os outros são definidos à custa dos elementos Tentidade, Tlink e Tlocal.

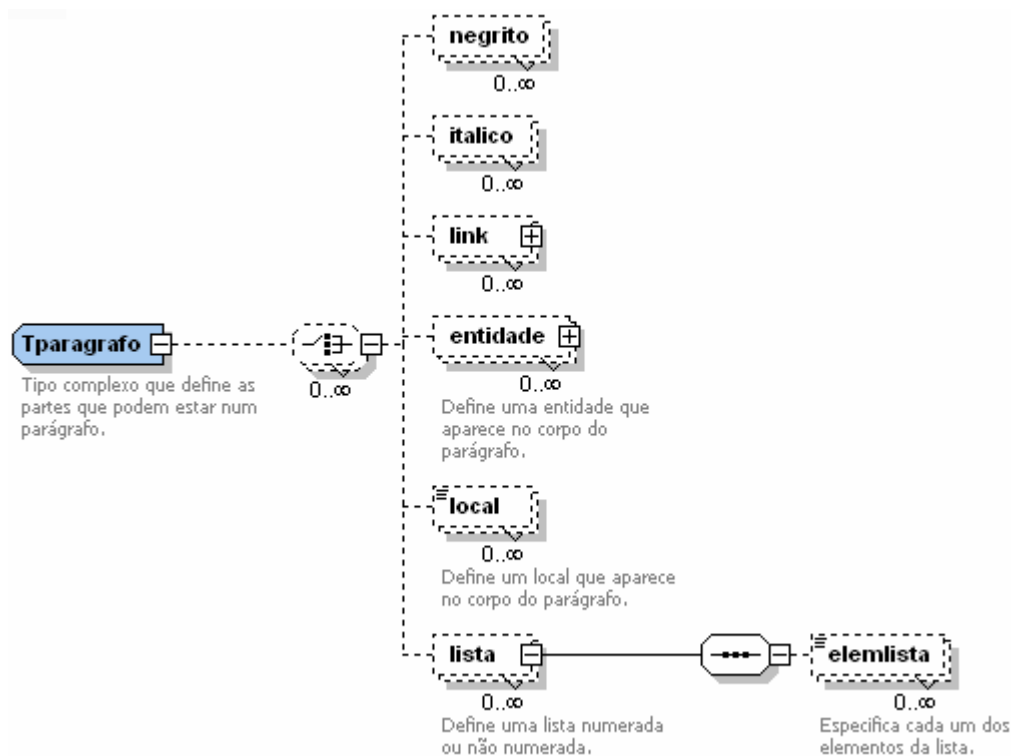


fig. 8 – Caracterização do tipo Tparagrafo.

Um elemento especial definido no parágrafo é lista. A lista é formada por uma série de elementos de texto e possui um atributo (tipo) que permite definir se a lista é numerada (valor 'numero') ou não-numerada (valor 'bola').

2.2. A folha de estilo

Dada a estrutura apresentada no schema, seria possível construir várias folhas de estilo que produzissem um output html diverso. As possibilidades estudadas foram:

- Criar uma série de folhas de estilo, dividindo os artigos por secções (ou por dias), que dariam origem a uma série de páginas html. Seria então construída uma folha de estilo adicional para criar os frames e índices necessários para a navegação no jornal
- Criar uma folha de estilo que desse origem a um documento html onde só fosse visível um artigo de cada vez, recorrendo às divisões (tags DIV) no ficheiro HTML.
- Criar uma folha de estilo com todos os artigos visíveis em simultâneo, recorrendo a links para a navegação no documento.

Como se nota, o schema construído permite uma grande flexibilidade na escolha do output html. Optou-se por seguir o último modelo indicado, atendendo a que ele é o mais versátil de todos, já que permite 4 tipos diferentes de leitura do jornal:

- Leitura sequencial dos artigos;

- Leitura por secções, recorrendo aos links apropriados;
- Leitura por uma ordem aleatória dos artigos, com os links para os artigos;
- “Pesquisa” por um tema, com recurso ao índice remissivo.

2.2.1. Templates principais

Como referido acima, foi necessário criar 3 índices: um para as secções, outro para os artigos, e mais um remissivo, que recolhe todas as entidades e locais dos artigos e cria um link para estes elementos.

Assim, o template principal do documento cria a ficha técnica, percorre os artigos uma vez para construir os índices de secção, uma segunda vez para criar os links para os artigos e ainda mais uma vez para tratar os parágrafos e outros elementos dos artigos. Além disso, chama também os templates que tratam os classificados e o índice remissivo.

Nos dois primeiros índices, já que os artigos podem ser introduzidos no documento XML por qualquer ordem, é necessário ‘arrumá-los’ por secção antes de os tratar, o que foi conseguido com o elemento **xsl:sort**, como no excerto abaixo:

```
<!-- Template para o índice para as secções que aparece a seguir à ficha
técnica do jornal -->
<xsl:apply-templates select="artigo" mode="indicegeral">
  <xsl:sort select="@seccao"/>
  <xsl:sort select="data" order="descending"/>
  <xsl:sort select="hora" order="descending"/>
</xsl:apply-templates>
```

2.2.2. Links para as secções, artigos e índice remissivo

Na criação de todos os índices propriamente ditos, foi utilizado o elemento **generate-id()**. Desta maneira, não foi difícil criar as ligações propriamente ditas. O problema encontrado na criação dos links para as secções (que aparecem logo a seguir à ficha técnica) foi que, como uma secção pode ter vários artigos, e a folha de estilo retira o nome das secções do atributo correspondente nos artigos, isto significava uma série de repetições de nomes de secções. Isto foi resolvido com um elemento **xsl:if** desta forma:

```
<!-- Geração do índice para as secções -->
<xsl:template match="artigo" mode="indicegeral">
  <xsl:if test="not(@seccao=following-sibling::*/@seccao)">
    <a href="#{@seccao}" style="text-decoration: none">
      <xsl:value-of select="@seccao"/>
      <xsl:text> | </xsl:text>
    </a>
  </xsl:if>
</xsl:template>
```

O elemento **xsl:if** garante que cada secção só aparece no índice uma vez, já que só permite que a folha de estilo escreva o nome da secção se não houver mais nenhum irmão (*following-sibling*) seguinte ao artigo actual da mesma secção. O mesmo processo é utilizado para que os nomes das secções não sejam repetidos no corpo do jornal.

No caso dos artigos não foi necessário recorrer a nenhum processo especial para a criação dos links apropriados. Note-se que, para ajudar à consulta do jornal, as secções, datas e hora de publicação são apresentados, assim como o resumo do artigo, se existir.

No caso do índice remissivo, a única dificuldade consistia em encontrar o título do artigo, dada uma entidade ou local. A solução foi recorrer ao eixo de navegação *ancestor* e localizar o título do artigo.

```
<xsl:for-each select="//paragrafo/entidade">
  <xsl:sort select="nome"/>
  <a href="#{nome}" style="text-decoration: none"><xsl:value-of
select="nome"/></a><br/>
  
  <xsl:value-of select="ancestor::artigo/titulo"/><br/>
</xsl:for-each>
```

2.2.3. Tratamento dos artigos

No tratamento dos artigos também foram levadas em conta uma série de possibilidades de apresentação. Como os parágrafos dos artigos da instância em XML são pequenos em tamanho, foi criada uma tabela com uma linha e duas colunas. A imagem pode ser colocada deste modo à direita ou à esquerda da imagem, conforme o valor do atributo correspondente (pos) no elemento imagem do parágrafo. Se este atributo não estiver presente, a imagem fica à direita do artigo. Se o artigo não tem uma imagem associada, a tabela não é criada.

```
<!-- A imagem pode ser colocada à direita do texto... -->
<xsl:if test="imagem/@pos='dir' or not(imagem/@pos)">
  <xsl:if test="imagem">
    <table>
      <tbody>
        <tr>
          <td>
            <font face="Garamond" size="3">
              <xsl:apply-templates select="paragrafo"/>
            </font>
          </td>
          <td align="center">
            
            <br/>
            <font face="Arial" size="1">
              <xsl:value-of select="imagem/legenda"/>
            </font>
          </td>
        </tr>
      </tbody>
    </table>
  </xsl:if>
</xsl:if>
```

Todos os outros elementos do artigo (autor, data, parágrafo, etc.) não necessitam nenhum tratamento especial.

2.2.4. Tratamento dos classificados

Os classificados são colocados numa tabela, onde constam o título, a mensagem e os telefones e e-mails de contacto. Como podem existir dois telefone e dois e-mails, a sua apresentação é tratada numa template à parte. Note-se que, se não existir um telefone ou e-mail de contacto, a entrada na tabela é um hífen.

```
<td align="center">
  <xsl:if test="not(telefone)">
    <xsl:text> - </xsl:text>
  </xsl:if>
  <xsl:apply-templates select="telefone"/>
</td>
<td align="center">
  <xsl:if test="not(email)">
    <xsl:text> - </xsl:text>
  </xsl:if>
  <xsl:apply-templates select="email"/>
</td>
```

2.2.5. Elementos opcionais

Na definição do schema, há uma abundância de elementos opcionais. Assim, a folha de estilo recorre profusamente ao elemento **xsl:if**, para testar a ocorrência dos elementos. Um exemplo pode ser visto na apresentação do sub-título ou o resumo dos artigos. Um artigo pode ter um sub-título, um resumo, ambos ou nenhum.

```
<xsl:if test="subtitulo">
  <i><b><xsl:value-of select="subtitulo"/></b></i>
  <p/>
</xsl:if>

<xsl:if test="resumo">
  <b><xsl:value-of select="resumo"/></b>
  <p/>
</xsl:if>
```

3. Conclusões

O objectivo de criar um jornal electrónico foi conseguido, e em especial, o objectivo principal, que era compreender a utilização do XML em contexto de trabalho foi amplamente alcançado.

Este jornal electrónico deve ser encarado como um ‘work-in-progress’. Obviamente poderiam ser introduzidos uma série de melhoramentos que aproximassem este projecto do mundo real. Possivelmente, isto será um objectivo para os próximos projectos desenvolvidos em XML.

4. Bibliografia

Ramalho, José C., Henriques, Pedro, XML & XSL – da Teoria à Prática, FCA – Editora de Informática, 2002.

Apontamentos das aulas de Processamento Estruturado de Documentos, Universidade do Minho, 2003.