

XML (II)

Linguagens de Anotação de Documentos

Esquemas XML

Esquemas XML

- O XML é uma “meta-linguagem” que permite estruturar informação arbitrária
- Desde as regras de *forma* sejam seguidas, qualquer tipo de elementos é aceite, o que permite escrever dados rapidamente e de maneira flexível
- No entanto, por vezes é necessário acordar a estrutura dos dados, para que todas as partes interpretem a informação de modo consistente
- Para tal é necessário formalizar o “esquema”, ou o tipo, dos ficheiros XML considerados *válidos*

Esquemas XML

- O “esquema” essencialmente define quais são os elementos e atributos válidos, e como é que podem ser organizados
 - Que tipo de elementos pode ser definido
 - Que atributos cada elemento pode/tem que ter
 - Qual o conteúdo de cada elemento (texto, outros elementos, vazio)

Exemplo

```
<grupos ano="16/17">
  <grupo id="1">
    <aluno>
      <nome>A</nome>
      <numero>1</numero>
    </aluno>
    <aluno>
      <nome>B</nome>
      <numero>2</numero>
      <nota>2</nota>
    </aluno>
    <tema>Tema 1</tema>
  </grupo>
  . . .
</grupos>
```

Exemplo

- Cada `grupos` tem um `ano` atribuído e pode conter vários `grupos`
- Cada `grupo` contém vários `alunos`, e pode conter um `tema`
- Cada `aluno` contém um `nome` e um `numero`, e opcionalmente uma `nota`

Linguagens de Esquemas

- Duas tecnologias para definir os tipos dos ficheiros XML
- *Document type definition* (DTD): mais simples mas mais limitada
- *XML Schema Definition* (XSD): mais avançada mas mais complexa

Declaração do Tipo

- O DTD é definido através da declaração do tipo do documento
 - Dentro do próprio ficheiro XML

```
<!DOCTYPE raiz [ declarações ]>
```

- Num ficheiro anexo

```
<!DOCTYPE raiz SYSTEM "ficheiro">
```


Declaração de Elementos

- Três objetos podem ser declarados num DTD:
 - Elementos válidos, `!ELEMENT`
 - Atributos válidos, `!ATTLIST`
 - Entidades (abreviaturas), `!ENTITY`

Declaração de Elementos

- Elementos válidos e o seu conteúdo

`<!ELEMENT elemento conteúdo>`

- O conteúdo dos elementos pode ser:
 - Texto livre, `#PCDATA`
 - Vazio, `EMPTY`
 - Qualquer conteúdo, `ANY`
 - Elemento filho, `(elem)`
 - Sequência de elementos filhos, `(elem1, elem2, ...)`

Declaração de Elementos

- Multiplicidades:
 - Exatamente elemento: *elem*
 - Uma sequência de elemento: *elem**
 - Uma sequência com pelo menos um elemento: *elem+*
 - Um elemento opcional: *elem?*
- Opções:
 - Alternativa entre elementos, *elem1 | elem2*
 - Combinações, e.g., *(elem1 | elem2)**

Declaração de Elementos

```
<!ELEMENT lad (grupos+)>  
<!ELEMENT grupos (grupo+)>  
<!ELEMENT grupo (aluno+,tema?)>  
<!ELEMENT aluno (nome,numero,nota?)>  
<!ELEMENT nome (#PCDATA)>  
<!ELEMENT numero (#PCDATA)>  
<!ELEMENT nota (#PCDATA)>  
<!ELEMENT tema (#PCDATA)>
```

Declaração de Atributos

- Atributos válidos de elementos e o seu conteúdo

`<!ATTLIST elemento atributo tipo valor>`

- O valor representa a escolha predefinida do atributo:
 - Simplesmente um valor predefinido, *valor*
 - Sem valor predefinido mas definição obrigatória, `#REQUIRED`
 - Sem valor predefinido mas definição opcional, `#IMPLIED`
 - Com valor predefinido fixo `#IMPLIED valor`

DTDs

- Tipos de atributos:
 - Texto livre, `CDATA`
 - Escolha entre elementos predefinidos, `(valor1 | valor2 | ...)`
 - Identificador único, `ID`
 - Apontador para identificador único, `IDREF`
 - ...

Declaração de Elementos

```
<!ATTLIST grupos ano CDATA #IMPLIED>
```

```
<!ATTLIST grupo id ID #REQUIRED>
```

Declaração de Entidades

- Abreviaturas ou atalhos para expressões

```
<!ENTITY entidade "valor">
```

- Exemplo:

- Declaração

```
<!ENTITY lad "Linguagens de Anotação de Documentos">
```

- Uso no XML

```
<tema> &lad; </tema>
```


Consultas XML

Consultas

- Para informação complexa e de grande dimensão é impossível analisar os dados manualmente
- Uma das funcionalidades mais úteis de dados estruturados é a possibilidade de fazer *consultas* (*queries*)
- Permitem seleccionar e filtrar informação do documento automaticamente
- Podem ser usadas para tarefas básicas como gerar relatórios e estatísticas
- Primeiro passo para funcionalidades mais avançadas, que permitem criar novo conhecimento a partir dos dados disponíveis

Consultas em XML

- As consultas em XML são construídas à volta da sua estrutura em árvore, com elementos aninhados
- A linguagem básica de navegação é o *XPath*, que define “caminhos” (*paths*) pela estrutura da árvore
- Está na base de outras tecnologias XML, como o XQuery para consultas avançadas, e o XSLT para a transformação de documentos

XPath

- Uma expressão XPath básica refere-se a um *caminho* para navegar por um documento XML
- Estes elementos podem ser filtrados de acordo com *predicados*
- Disponibiliza também uma biblioteca de *funções* para manipular os dados obtidos (e.g., operações numéricas)

Caminhos

- Sintaxe das expressões:
 - O operador / seleciona os filhos do elemento atual
 - O operador // seleciona os descendentes do elemento atual (filhos de filhos..)
 - O operador @ seleciona os atributos no ponto atual
 - O nome de um elemento seleciona todos os elementos desse tipo no ponto atual
 - O operador * seleciona todos os elementos no ponto atual
 - ...

Caminhos

- Operadores úteis sobre elementos
 - Todos os elementos do ponto atual, `node()`
 - Conteúdo textual no ponto atual, `text()`
 - Comentários no ponto atual, `comment()`

Caminhos

- Selecciona a raiz do documento

`/lad`

- Selecciona todos os `grupos`, de todos os `grupos`

`/lad/grupos/grupo`

- Selecciona todos os `nomes` em qualquer parte do documento

`//nome`

- Selecciona os atributos `ano` dos `grupos`

`/nome/grupos/@ano`

Predicados

- Os caminhos pode ser anotados com predicados, que filtram elementos de acordo com certos critérios
 - Sobre o conteúdo dos elementos
 - Sobre os atributos
 - Sobre a posição dos elementos
 - ...
- Definidos como *caminho*[*predicado*]

Predicados

- Selecciona o primeiro grupo de cada grupos
`/lad/grupos/grupo[1]`
- Selecciona grupos com ano definido
`/lad/grupos[@ano]`
- Selecciona alunos com nota positiva
`/lad/grupos/grupo/aluno[@nota>9]`
- Selecciona nomes de alunos de um ano especifico
`/lad/grupos[@ano="16/17"]/aluno/nome/text()`
- Selecciona nomes de alunos com nota positiva de um ano especifico
`/lad/grupos[@ano="16/17"]/aluno[@nota>9]/nome/text()`

Take-home Lesson

- Por vezes é necessário acordar nos esquemas dos documentos XML, para que todas as partes interessadas representem os dados da mesma maneira
- O processo mais simples em XML é através de DTDs, onde se definem os elementos e os atributos válidos, assim como o seu conteúdo
- Dados estruturados também permitem a utilização de linguagens de consulta para automatizar pesquisas
- Em XML expressões XPath podem ser usadas para consultas básicas que navegam a árvore XML e selecionam informação

Tutorial

https://www.w3schools.com/xml/xml_dtd_intro.asp

https://www.w3schools.com/xml/xpath_intro.asp