

DTD

Carlos A. Heuser
UFRGS

03/1

Esquema de documentos XML

- Para um documento XML pode ser definido um **esquema**
 - Análogo ao esquema de um banco de dados
- Documento pode ser
 - **bem-formado** (*well-formed*): está de acordo com a sintaxe XML (seção precedente)
- Adicionalmente, documento pode ser
 - **válido** (*valid*): consistente com seu esquema

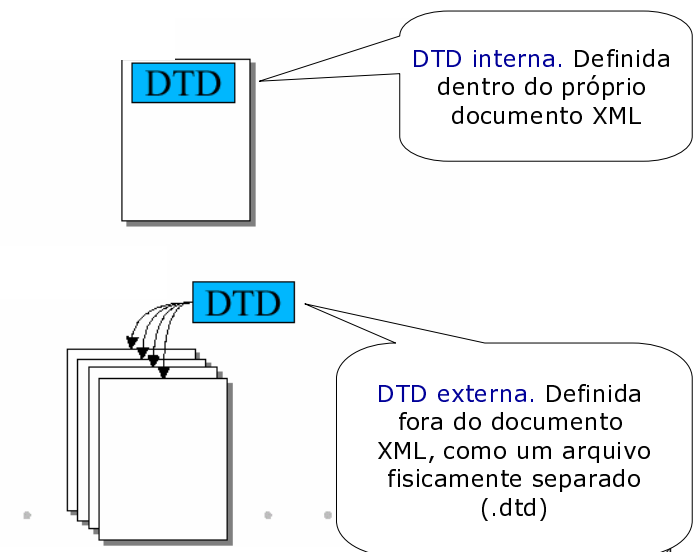
03/2

Linguagens para especificar esquemas XML

- DTD - *Document Type Definition*
 - Linguagem herdada do SGML
 - Sintaxe de: Gramática Regular
 - Poder de expressão limitado
- *XMLSchema*
 - Sintaxe: XML
 - Maior poder de expressão

03/3

DTD interna e externa



03/4

DTD interna

arquivo.xml

```
<?xml version="1.0"?>
<!DOCTYPE artigo [
  <!ELEMENT artigo (titulo, autor)>
  <!ELEMENT titulo (#PCDATA)>
  <!ELEMENT autor (nome)>
  <!ELEMENT nome (#PCDATA)>
]>
<artigo>
  <titulo> ... .. </titulo>
  <autor>
    <nome> ... .. </nome>
  </autor>
</artigo>
```

03/5

DTD externa

artigo.xml

```
<?xml version="1.0"?>
<!DOCTYPE artigo SYSTEM "artigo.dtd">
<artigo>
  <titulo> ... .. </titulo>
  <autor>
    <nome> ... .. </nome>
  </autor>
</artigo>
```

artigo.dtd

```
<!ELEMENT artigo (titulo, autor)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT autor (nome)>
<!ELEMENT nome (#PCDATA)>
```

03/6

Declaração da DTD

```
<?xml version="1.0"?>
<!DOCTYPE artigo SYSTEM "artigo.dtd">
<artigo>
  <titulo> ... .. </titulo>
  <autor>
    <nome> ... .. </nome>
  </autor>
</artigo>
```

Nome do tipo de documento
e do elemento raiz
devem ser **IGUAIS**

Declaração de elementos

- ❑ Feito com a declaração `ELEMENT`
- ❑ Exemplo de declaração do elemento denominado `capitulo`
`<!ELEMENT capitulo definição_do_conteúdo>`
- ❑ Observar que antes e depois do nome do elemento deve vir branco

03/8

Declaração de elemento texto

- Quando um elemento é composto por texto:

```
<!ELEMENT titulo (#PCDATA)>
```

- Na DTD, há somente um tipo de elemento texto (folhas na árvore DOM)
- #PCDATA é um acrônimo para *Parsable Character Data*
 - Texto que vai ser processado pelo processador XML a procura de rótulos
- Em XMLSchema é possível definir muitos outros tipos de texto

03/9

Declaração de elemento vazio

```
<!ELEMENT imagem EMPTY>
```

- Usado para definir elementos **vazios**
- Elemento vazio **não contém**:
 - nem texto,
 - nem outros elementos

03/10

Declaração de elemento vazio

- Quando usar (1):

- Presença/ausência do elemento vazio é uma informação

- Exemplo:

```
<profissoes>  
  <professor/>  
  <engenheiro/>  
</profissoes>
```

03/11

Declaração de elemento vazio

- Quando usar (2):

- Elemento contém apenas atributos

- Exemplo:

```
<essoa  
  codigo="123"  
  nome="José Antônio"  
  dataDeNascimento="21/07/1982"  
>
```

03/12

Declaração ANY

□ Exemplo:

```
<!ELEMENT titulo ANY>
```

- O elemento pode ser **qualquer** coisa **válida** em XML
- É útil quando está-se construindo a DTD e ainda não foi decidido qual será o tipo do elemento

03/13

Declaração de um elemento composto

```
<!ELEMENT capitulo (elementos_filho)>
```

- Declara um elemento XML **composto** por outros elementos conforme a declaração *elementos_filho*

□ Exemplo:

```
<!ELEMENT capitulo (titulo,secao)>
```

- Declara que o elemento `capitulo` contém os elementos `titulo` e `secao`

03/14

Declaração de elementos filhos – sequência

□ Elementos filhos em **sequência**

- Usa o conector `“, ”`

□ Exemplo

```
<!ELEMENT artigo (titulo,autor,secao)>
```

- Declara que `artigo` contém:
 - exatamente um elemento `titulo`,
 - seguido de exatamente um elemento `autor`
 - seguido de exatamente um elemento `secao`

03/15

Declaração de elementos filhos – escolha

□ Conector de **escolha**

- Semântica de um “ou exclusivo”

□ Usa o símbolo `“|”`

Ex.:

```
<!ELEMENT endereco (particular|comercial)>
```

- Significa que um elemento `endereco` contém um elemento `particular` ou um elemento `comercial`, mas não ambos

03/16

Controle de cardinalidade

□ Cardinalidade (0,1)

- Elemento opcional sem repetição: "?"

```
<!ELEMENT autor (prenome,nomemeio?,sobrenome)>
```

□ Cardinalidade (1,n)

- Elemento requerido e com repetição: "+"

```
<!ELEMENT nome (palavra+)>
```

□ Cardinalidade (0,n)

- Elemento opcional e com repetição: "*"

```
<!ELEMENT autor (nome,telefone*)>
```

03/17

Combinando operadores

□ Os operadores podem ser combinados

□ Exemplo:

```
<!ELEMENT endereco  
  (empresa,  
   (ruaNumero | (rua,numero)),  
   (telPrivado | telComercial | telCelular)*)>
```

□ Cuidado, padrão não estabelece regras de prioridade de operadores.

□ O que significa?

```
<!ELEMENT endereco (ruaNumero | rua,numero)>
```

03/18

Conteúdo misto

□ Elemento de conteúdo misto contém tanto outros elementos, quanto texto.

□ Exemplo:

```
<nota>  
  <de>Carlos</de>  
  <para>Eliane</para>  
  Nao esquecer <ingles>birthday</ingles>  
  amanha  
</nota>
```

03/19

Especificação de conteúdo misto

□ Toda definição de elemento de conteúdo misto deve:

- Começar por #PCDATA
- Ser um grupo deve ser de **escolha**, e
- Ser **opcional** e com **repetição**.

□ Exemplo:

```
<!ELEMENT nota (#PCDATA | de | para | ingles)*>
```

03/20

Declaração de atributos

- Os atributos de um elemento são especificados separadamente em uma ou mais declarações ATTLIST

- Sintaxe:

```
<!ATTLIST element-name
    attribute-name attribute-type default-value>
```

- Exemplo

```
<!ATTLIST nota data CDATA #REQUIRED
    status ( old | new ) #REQUIRED>
```

Para:

```
<nota data="12/05/1982" status="new"/>
```

03/21

Declaração de atributos

- Atributo **obrigatório** e sem valor *default*: palavra reservada **#REQUIRED**

```
<!ATTLIST autor codigo CDATA #REQUIRED>
```

- Atributo **opcional** palavra reservada **#IMPLIED**

```
<!ATTLIST autor email CDATA #IMPLIED>
```

- Atributo pode estar ausente. Neste caso, *default* é assumido

```
<!ATTLIST autor sexo ( F | M ) "F" >
```

```
<!ATTLIST autor sexo CDATA "F" >
```

- Atributo obrigatório e com valor constante: palavra reservada **#FIXED**

```
<!ATTLIST artigo versao CDATA #FIXED "1.0">
```

03/22

Múltiplas declarações

- Um elemento pode ter mais de uma declaração de lista de atributos.

- Um atributo pode ser declarado múltiplas vezes.

- Exemplo:

```
<!ATTLIST livro id ID #REQUIRED
    tipo (romance| policial) #REQUIRED>
<!ATTLIST livro tipo (pocket| normal) #REQUIRED
    autor CDATA #IMPLIED>
```

- As listas de atributos são combinadas.

- No caso de haver duas declarações para o mesmo atributo, a **primeira declaração** é a que vale!

- Estilo padrão de XML:

- regras locais vão no início dos arquivos e sobre-escrevem as regras que estão no arquivo

03/23

Tipos de atributos vistos até aqui

- CDATA – texto

- Exemplo:

```
<ATTLIST documento versao CDATA #REQUIRED>
```

- Uso:

```
<documento versao="10.10.2001">...</documento>
```

- (val1 | val2 | ...) – enumeração de valores

- Os valores da lista tem que ter a sintaxe de um nome de elemento (não podem conter brancos)

- Exemplo:

```
<!ATTLIST nota status ( old | new ) #REQUIRED>
```

03/24

Tipo de atributo - NMTOKEN

- ❑ Valor do atributo é um string com a mesma sintaxe de um nome de elemento XML.
- ❑ Definição:
`<ATTLIST documento digito_verificacao NMTOKEN #IMPLIED>`
- ❑ Uso:
`<documento digito_verificacao="art1"/>`
- ❑ NMTOKENS – Valor do atributo é um string com uma lista de nomes de elemento separados por brancos.
- ❑ Definição:
`<ATTLIST documento paginas NMTOKENS #IMPLIED>`
- ❑ Uso:
`<documento paginas="elem10 elem45 el77 r102"/>`

03/25

Tipos de atributo – ID e IDREF

- **ID** – um identificador → **deve ter valor único no documento**
Definição:
`<ATTLIST capitulo Nr ID #REQUIRED>`
Uso:
`<capitulo Nr="A1">...</capitulo>`
Os valores para o tipo ID seguem as mesmas regras de construção de nomes de elementos
- **IDREF** – referência a um ID
Definição:
`<ATTLIST referencia cap IDREF #REQUIRED>`
Uso:
O capítulo `<referencia cap="A1"/>` mostra as características de ...
- **IDREFS** – referência a vários IDs
Definição:
`<ATTLIST referencia cap IDREFS #REQUIRED>`
Uso:
Os capítulos `<referencia cap="A2 A3"/>` apresentam ...

03/26

ID e IDREF - exemplo

- Na DTD:
`<!ELEMENT capitulo(...)>`
`<!ATTLIST capitulo nr ID #REQUIRED>`
`<!ELEMENT referencia EMPTY>`
`<!ATTLIST referencia cap IDREF #REQUIRED>`
- Na instância (documento) XML
...
`<capitulo nr="A1">`
...
`</capitulo>`
O valor atribuído ao atributo `cap` deve ser válido, ou seja, deve haver um atributo do tipo ID com aquele valor
...
`<conclusao>` Enfim, deve-se ... Como apresentado inicialmente no capítulo `<referencia cap="A1"/>` deve-se tomar cuidado ao definir ...
`</conclusao>`
...

03/27

Outros tipos (vistos mais adiante)

- **ENTITY** – entidade
Definição:
`<ATTLIST figura arquivo ENTITY #REQUIRED>`
Uso:
`<figura arquivo="foto"/>`
- **ENTITIES** – várias entidades
Definição:
`<ATTLIST figura arquivo ENTITIES #REQUIRED>`
Uso:
`<figura arquivo="fotolano foto10anos foto15anos"/>`
- **NOTATION** – uma notação
Definição:
`<!NOTATION gif SYSTEM " ">`
`<ATTLIST figura tipo NOTATION #REQUIRED>`
Uso:
`<figura tipo="gif">`

03/28

Seções condicionais

☐ Porções da DTD podem ser consideradas opcionais

☐ As seções que serão consideradas pelo processador XML devem ser marcadas como uma **seção incluída**

```
<![INCLUDE[ .. .... . . . ]]>
```

☐ As seções que serão consideradas pelo processador XML devem ser marcadas como uma **seção ignorada**

```
<![IGNORE[ .. .... . . . ]]>
```

03/29

Documento sem DTD

☐ Documento XML sem DTD:

○ Atributos **não podem ter valor default**

○ Todos os atributos são considerados **CDATA**

○ Todos os atributos são considerados **opcionais**

○ **Entidades** (próxima seção) **não** podem ser definidas

03/30