

Engenharia de Ontologias (Ontology Engineering)

Universidade Federal de Uberlândia

Faculdade de Computação

Programa de Pós-Graduação em Ciência da
Computação

Prof. Fabiano Azevedo Dorça

XML DTD

- Document Type Definition (DTD)
 - Define as **regras de formação** dos elementos e atributos
 - **Quais os elementos** podem aparecer em um documento
 - Em que **ordem** eles podem aparecer
 - Qual a **hierarquia** permitida para os elementos
 - Quais **atributos um elemento** pode conter

XML DTD

- Considere o seguinte XML

```
<?xml version="1.0"?>
```

```
<recado>
```

```
  <de>Maria</de>
```

```
  <para>José</para>
```

```
  <mensagem>Reunião amanhã as 14hs</mensagem>
```

```
</recado>
```

XML DTD

- DTD correspondente
- `<!ELEMENT recado (de, para, mensagem)>`
- `<!ELEMENT de (#PCDATA)>`
- `<!ELEMENT para (#PCDATA)>`
- `<!ELEMENT mensagem (#PCDATA)>`
- Obs.: O conteúdo texto de um elemento pode ser PCDATA apenas.

XML DTD

- #PCDATA - Parsable Character Data
 - conteúdo do elemento será **analisado pelo processador** que está lendo o arquivo XML.
- Texto encontrado entre a marca inicial e a marca final de um elemento XML.
- As *tags* dentro do texto **serão tratadas como marcação** e as entidades **serão expandidas**.
- Ex.: <nome>João da Silva</nome>

XML DTD

- CDATA também significa dados de caracteres.
- CDATA é texto que **não será analisado pelo parser**.
 - As *tags* dentro do texto **NÃO serão tratadas como marcação** e as entidades não serão expandidas.
- Usado como tipo caracter em atributos
<!ELEMENT empregado (nome, sobrenome)>
<!ATTLIST empregado
 cod CDATA #REQUIRED
 dept CDATA #REQUIRED>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT sobrenome (#PCDATA)>
- XML: <empregado cod="XX" dept="YY">....</empregado>

XML DTD

- DTD Interno

```
<?xml version="1.0"?>
<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to    (#PCDATA)>
  <!ELEMENT from  (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body  (#PCDATA)>
]>
<note>
  <to>João</to>
  <from>Maria</from>
  <heading>Reunião</heading>
  <body>Reunião agendada para amanhã as 14hs</body>
</note>
```

XML DTD

- DTD externo

DTD privada
Use PUBLIC para
DTDs publicas

URL relativa ou absoluta

```
<?xml version="1.0"?>  
<!DOCTYPE note SYSTEM "note.dtd">  
<note>  
  <to>João</to>  
  <from>Maria</from>  
  <heading>Reunião</heading>  
  <body>Reunião agendada para amanhã as 14hs</body>  
</note>
```

XML DTD

- Conteúdo do arquivo "note.dtd":

```
<!ELEMENT note (to,from,heading,body)>
```

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

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

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

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

- Possibilidades:
 - Usar uma DTD comum para troca de dados.
 - Usar DTD padrão para verificar se os dados recebidos do mundo externo são válidos.
 - Usar um DTD para verificar seus próprios dados.

XML DTD

- DTD público
 - DTDs públicos externos são identificados pela palavra-chave PUBLIC e destinam-se a um uso amplo.
 - O "DTD_location" é usado para localizar o DTD público se ele não pode ser localizado pelo "DTD_name".

XML DTD

- `<!DOCTYPE root_element PUBLIC "DTD_name" "DTD_location">`
 - onde:
 - DTD_location: URL relativa ou absoluta
 - DTD_name: segue a sintaxe:
 - "prefix//owner_of_the_DTD//
description_of_the_DTD//ISO
639_language_identifier"

XML DTD

- Os seguintes prefixos são permitidos no nome DTD:
 - **ISO** O DTD é um padrão ISO. Todas as normas ISO são aprovadas.
 - **+** A DTD é uma norma não-ISO aprovada.
 - **-** O DTD é um padrão não-ISO não aprovado.

XML DTD

- Exemplo:

```
<?xml version="1.0" standalone="no"?>  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0  
Transitional//EN"  
  "http://www.w3.org/TR/REC-html40/loose.dtd">
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>A typical HTML file</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

This is the typical structure of an HTML file. It follows the notation of the HTML 4.0 specification, including tags that have been deprecated (hence the "transitional" label).

```
</BODY>
```

```
</HTML>
```

XML DTD

- Elementos vazios
- Elementos vazios são declarados com a palavra-chave EMPTY dentro dos parênteses:
- `<! ELEMENT nome-elemento EMPTY >`
- Exemplo:
- `<!ELEMENT br EMPTY >`
- XML válido:
- `
`

XML DTD

- Cardinalidade: + * ?
 - + um ou mais
 - * zero ou mais
 - ? zero ou um
- Se não houver símbolo de cardinalidade ao lado do elemento, ele é obrigatório.

XML DTD

- `<!ELEMENT empregados (empregado+)>`
-
- Existe um elemento `empregados`, cujo conteúdo é constituído de um ou mais elementos `empregado`
- No documento XML

```
<empregados>  
  <empregado>...</empregado>  
  <empregado>...</empregado>  
  <empregado>...</empregado>  
</empregados>
```

XML DTD

- `<!ELEMENT empregados (empregado*)>`
-
- Existe um elemento `empregados`, cujo conteúdo é constituído de zero ou mais elementos `empregado`
- Possíveis XML válidos

```
<empregados>  
  <empregado>...</empregado>  
  <empregado>...</empregado>  
</empregados>
```

ou

```
<empregados></empregados>
```

XML DTD

- `<!ELEMENT empregado (nome, inicial-meio?, sobrenome)>`
- O elemento `empregado` é composto de 3 sub elementos: `nome` (obrigatório), `inicial-meio` (opcional), e `sobrenome` (obrigatório)
- XML válido
`<empregado>`
 `<nome>João</nome>`
 `<sobrenome>Santos</sobrenome>`
`</empregado>`

XML DTD

- Sequência e Escolha
- `<!ELEMENT empregado (nome, inicial-meio, sobrenome)>`
- O elemento empregado é composto de 3 sub-elementos: nome, inicial-meio, e sobrenome, nessa ordem

```
<empregado>  
  <nome>João</nome>  
  <inicial-meio>S.</inicial-meio>  
  <sobrenome>Santos</sobrenome>  
</empregado>
```

XML DTD

- `<!ELEMENT cliente (cpf | rg | código)>`
- O elemento cliente é composto de 1 sub-elemento: ou cpf ou rg ou código

```
<cliente>
```

```
  <cpf>111.111.111-11<\cpf>
```

```
<\cliente>
```

XML DTD

- Lista de atributos - ATTLIST
 - Nome do elemento que contém o atributo
 - Lista de atributos
 - Para cada atributo da lista, devem ser declaradas as seguintes informações
 - nome do atributo
 - tipo do atributo (CDATA, ID, etc.), ou lista de valores
 - possíveis, separados por “|”
 - obrigatoriedade (#REQUIRED, #FIXED, #IMPLIED, ou um valor *default* para o atributo, informado entre aspas)

XML DTD

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

```
<!ATTLIST empregado
```

```
  cod CDATA #REQUIRED
```

```
  dept CDATA #REQUIRED
```

```
>
```

← Obrigatório fornecer o valor do atributo

No documento XML válido:

```
<empregado cod="E01" dept="D01">João</empregado>
```

XML DTD

- Atributos *opcionais*: utiliza-se a palavra reservada #IMPLIED
<!ATTLIST empregado dept CDATA #IMPLIED>
- Atributos com *valores fixos*: #FIXED valor
<!ATTLIST empregado empresa CDATA #FIXED "ABC">
- Atributos podem ter valores *default*
<!ATTLIST empregado sexo (F | M) "F" >
<!ATTLIST empregado sexo CDATA "F" >
 - Neste caso, se o documento XML não fornecer o valor do atributo, o parser o adiciona automaticamente na hora do processamento.

XML DTD

- Tipo ID – identificador - deve ter valor único no documento
- `<ATTLIST capitulo Nr ID #REQUIRED>`
 - XML: `<capitulo Nr="A1">...</capitulo>`
 - Observação: Os valores para o tipo ID seguem as mesmas regras de construção de nomes de elementos (devem começar por letra, “:” ou “_”)

XML DTD

- Elemento Misto

```
<!ELEMENT endereco (#PCDATA | cidade)*>
```

- Elemento endereço pode possuir texto e subelemento(s) cidade
- No documento XML:

```
<endereco>Rua das Flores, 45  
<cidade>Rio de Janeiro</cidade>  
CEP 24220-260  
</endereco>
```

XML DTD

- Regra importante que deve ser seguida quando existe escolha entre texto e elementos filho:
 - #PCDATA deve ser obrigatoriamente o primeiro *token* no grupo o grupo deve ser de escolha e com repetição.

XML DTD

- Conteúdo livre
 - `<!ELEMENT element-name ANY>`
- Podem conter qualquer combinação de parsable data (elementos, #PCDATA)

XML DTD

Exemplo:

```
<!ELEMENT collection (description,recipe*)>

<!ELEMENT description ANY>

<!ELEMENT recipe (title,ingredient*,preparation,comment?,nutrition)>

<!ELEMENT title (#PCDATA)>

<!ELEMENT ingredient (ingredient*,preparation)?>
<!ATTLIST ingredient name CDATA #REQUIRED
                    amount CDATA #IMPLIED
                    unit CDATA #IMPLIED>

<!ELEMENT preparation (step*)>

<!ELEMENT step (#PCDATA)>

<!ELEMENT comment (#PCDATA)>

<!ELEMENT nutrition EMPTY>
<!ATTLIST nutrition protein CDATA #REQUIRED
                    carbohydrates CDATA #REQUIRED
                    fat CDATA #REQUIRED
                    calories CDATA #REQUIRED
                    alcohol CDATA #IMPLIED>
```

XML DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<collection>
  <description>
    Some recipes used for the XML tutorial.
  </description>
  <recipe>
    <title>Beef Parmesan with Garlic Angel Hair Pasta</title>
    <ingredient name="beef cube steak" amount="1.5" unit="pound"/>
    ...
    <preparation>
      <step>
        Preheat oven to 350 degrees F (175 degrees C) .
      </step>
      ...
    </preparation>
    <comment>
      Make the meat ahead of time, and refrigerate over night, the acid in the
      tomato sauce will tenderize the meat even more. If you do this, save the
      mozzarella till the last minute.
    </comment>
    <nutrition calories="1167" fat="23" carbohydrates="45" protein="32"/>
  </recipe>
  ...
</collection>
```

XML DTD

- Validação do XML
 - Browsers recentes só fazem a verificação do XML (sintaxe).
 - Não validam docs XML frente a DTD's e XML Schemas.
 - Ferramentas para verificação.
 - On-line
 - Off-line

XML DTD

- Prática 01
- Fim