Technologie pro web a multimedia

6. přednáška

XML, DTD, namespaces

Martin Klíma, Miroslav Bureš

Computer Graphics Group

XML – úvod

XML

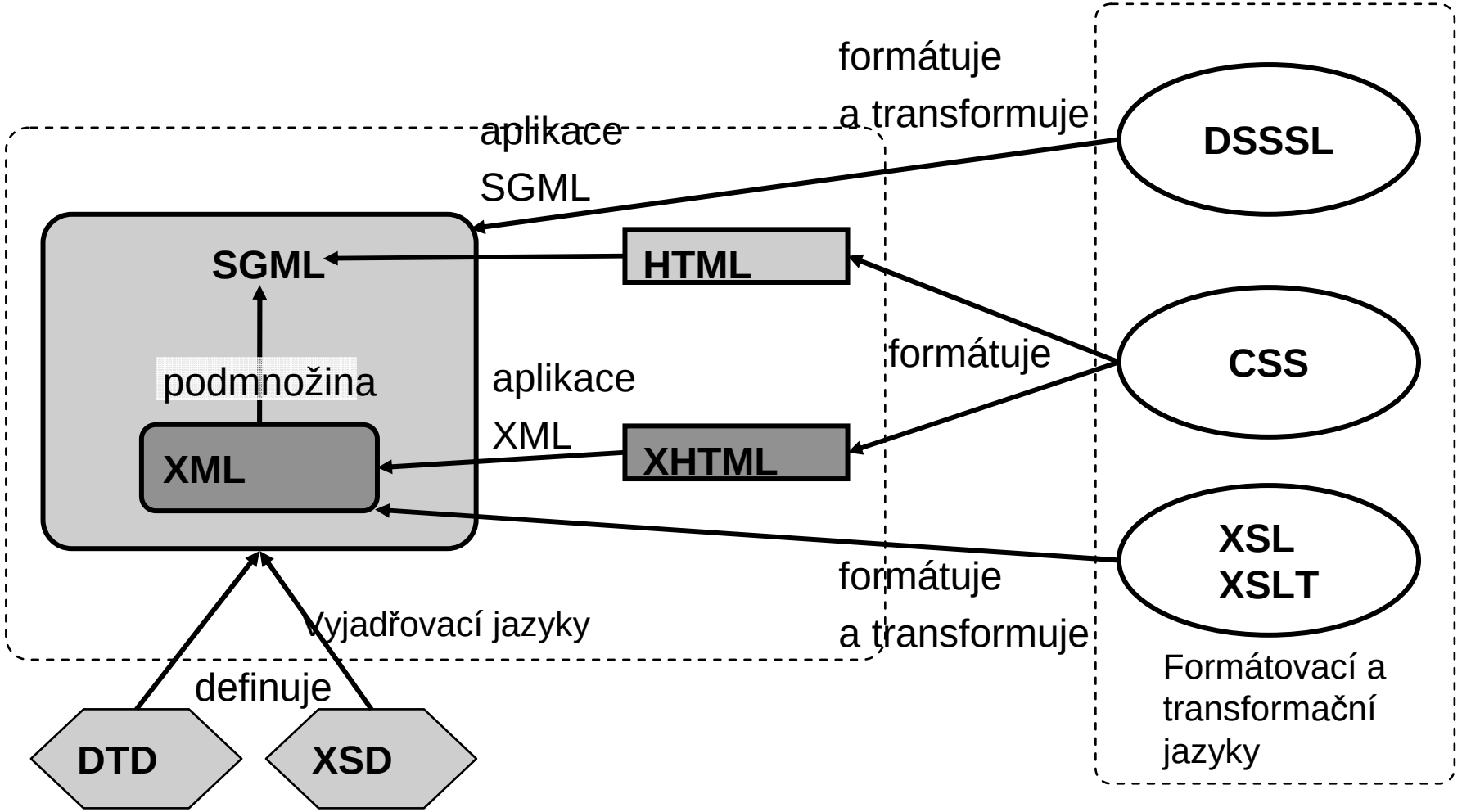
eXtensible Markup Language,

česky *rozšiřitelný značkovací jazyk*

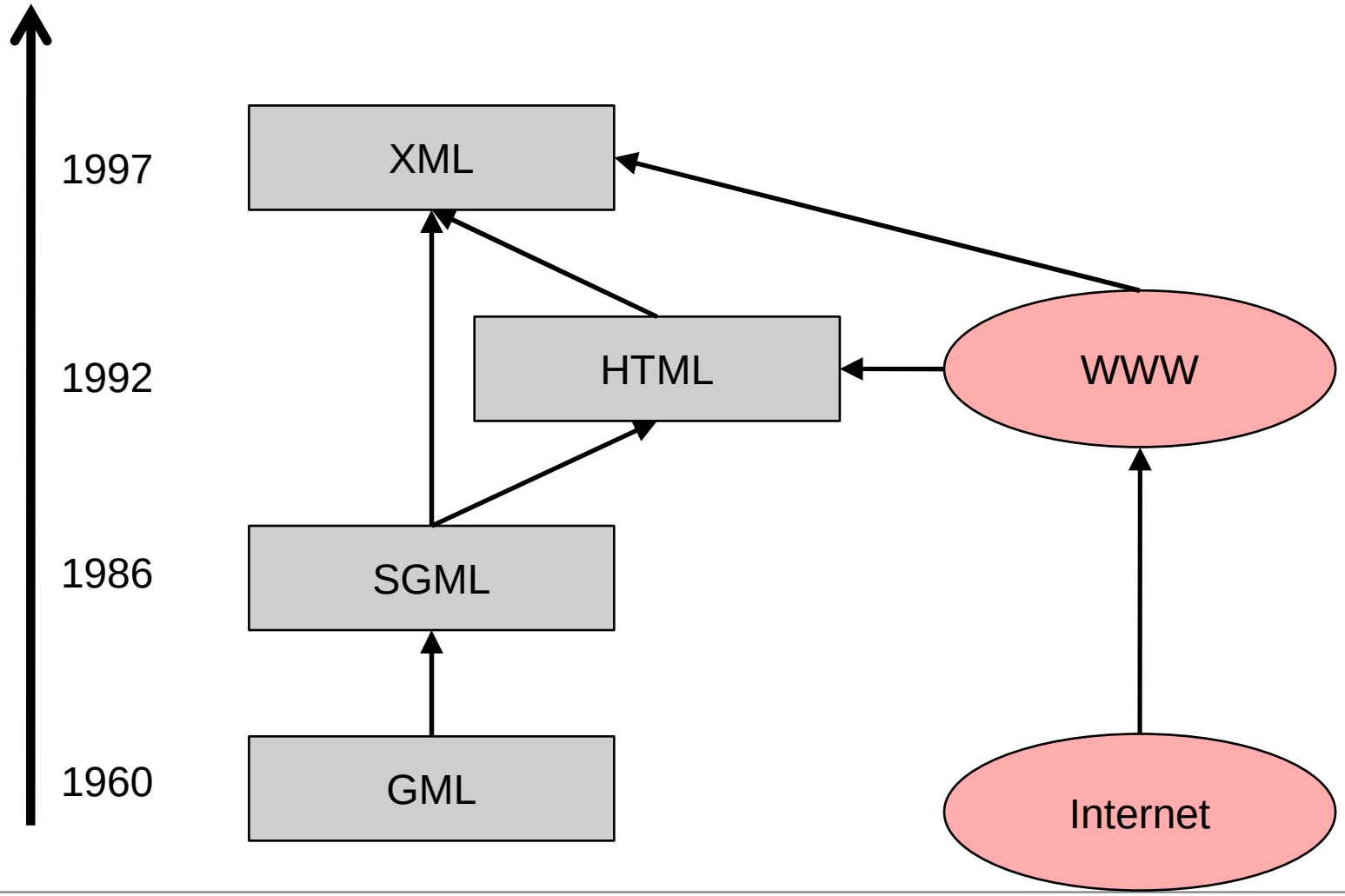
obecný značkovací jazyk, který byl vyvinut a standardizován konsorciem W3C.

Umožňuje snadné vytváření konkrétních značkovací jazyků pro různé účely a široké spektrum různých typů dat.

HTML a jeho vztah k ostatním jazykům



Historie XML



Technologie související s XML

- 1960 GML *(General Markup Language)*, vyvinut v IBM pro přenos dokumentů mezi různými platformami
- 1986 SGML *(Standard General Markup Language)*, přijato jako ISO standard. Umí reprezentovat téměř všechny dokumenty, značně složitý
- 1992 HTML *(Hypertext Markup Language)* vyvinuto v CERNu, je to aplikace SGML (definováno pomocí DTD)
- 1997 XML *(eXtensible Markup Language)* - zjednodušení SGML pro praktické použití konzorciem W3C

XML - vlastnosti

- Platformově nezávislý
  - textový dokument s pevně definovanými vlastnostmi
- Mezinárodní podpora
  - Lze použít formát Unicode, může tedy obsahovat všechny znaky všech národních abeced
  - XML dokument může obsahovat více různých jazyků najednou
- Tagy mohou mít názvy přímo vyjadřující jejich význam
  - XML dokumenty jsou proto dobře čitelné i pouhým okem
- Široká podpora ve většině moderních jazyků
  - parsery, kontrola správnosti podle gramatiky

XML - vlastnosti 2

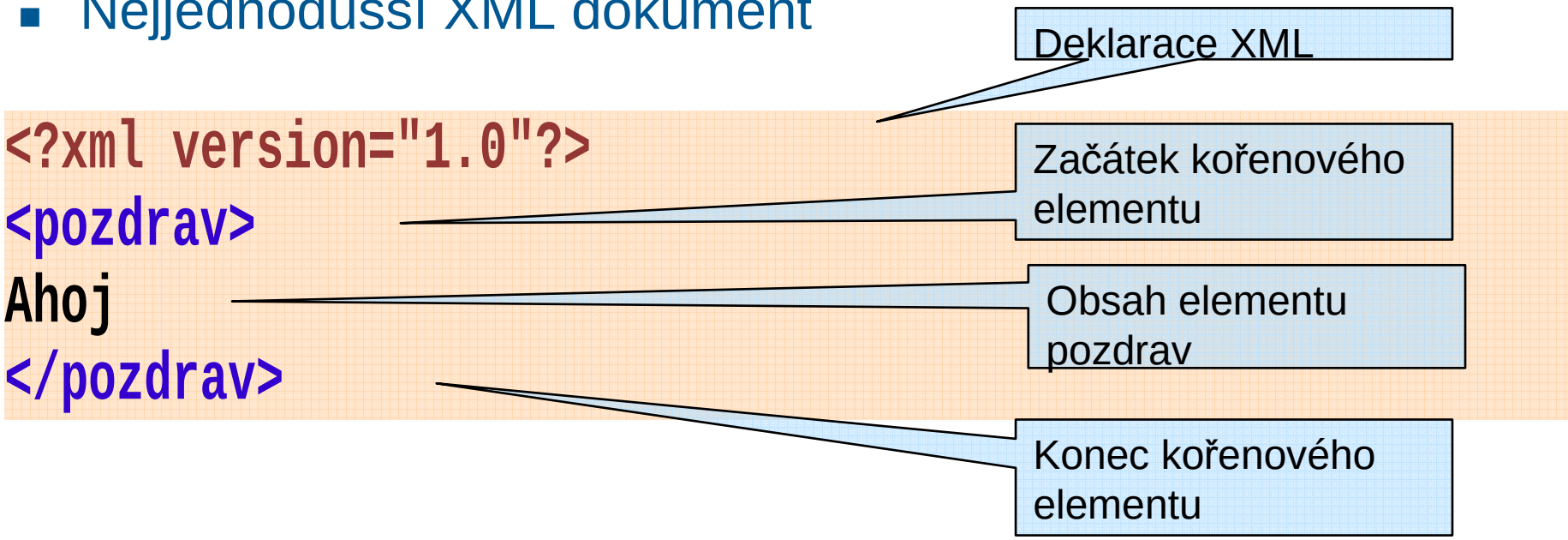
- Vytváření linků a adresace místa v XML dokumentu:
  - XLink, XPointer, XPath
- XSL (XML Stylesheet Language) je jazyk, který definuje, jak se má daný XML dokument transformovat do jiné podoby. Výsledná podoba nemusí být XML.
- Možné použít CSS (Cascading Style Sheets), jazyk pro formátování dokumetů:
  - CSS se stará o vzhled, zatímco informační obsah a struktura dokumentu je uložena v XML
  - CSS není XML formát
- Automatická kontrola správnosti dokumentů
  - jazyky pro definici struktrury dokumentu: DTD, XSD

Použití XML například:

- Dokumenty
  - docx, pptx, xlsx, …
  - XHTML
  - Konverze dokumentů, formáty pro import a export dat
- Integrace mezi systémy
- B2B (Business to Business)
  - Výměna dat mezi obchodními partnery
  - Požadavek platformové nezávislosti
- Webové technologie
- Konfigurační soubory
- …

Základy jazyka XML

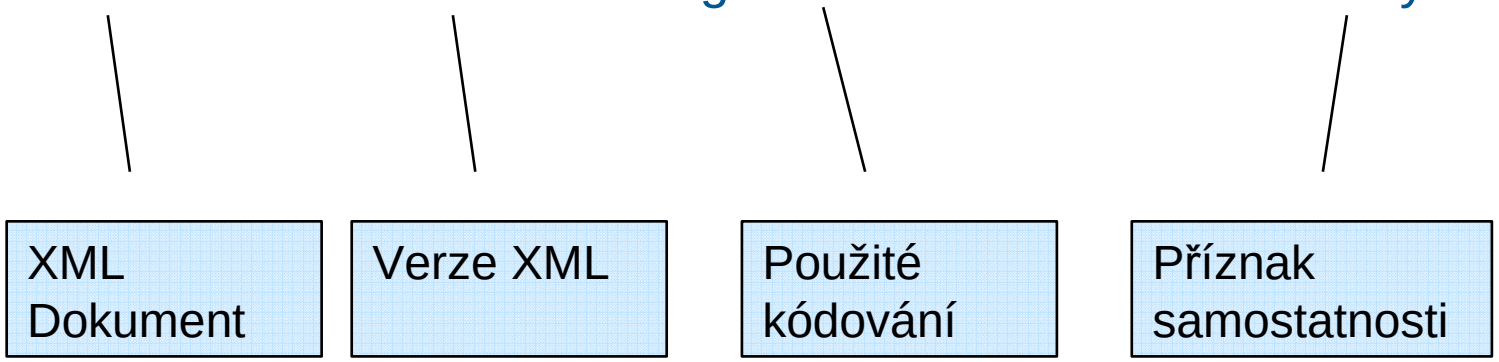
- Nejjednodušší XML dokument



Části XML dokumentu

- Deklarace XML

```
<?xml version="1.0" encoding="ISO-8859-2" standalone="yes"?>
```



- Deklarace je povinná

Části XML dokumentu

Každý XML dokument MUSÍ mít právě jeden kořenový element

- nesmí tedy mít žádný nebo více než jeden

```
<?xml version="1.0"?>
<pozdrav>
Ahoj
</pozdrav>
```

Kořenový element

```
<pozdrav>
Ahoj podruhé
</pozdrav>
```

Neplatné XML, 2 kořenové elementy

Elementy

```
<?xml version="1.0"?>
<dokument>
   <paragraf>
   text text text
    blablabla
   </paragraf>
   <oddelovac />
   <paragraf>
   další text
   blablabla
   </paragraf>
</dokument>
```

Začátek (kořenového) elementu **dokument**

Začátek elementu **paragraf**

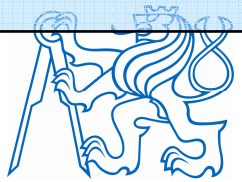
Konec elementu **paragraf**

Prázdný element **oddelovac**

Začátek elementu **paragraf**

Konec elementu **paragraf**

Konec (kořenového) **elementu** dokument



Elementy

- Elementy musí být ukončeny

  <body> </body>
- Pokud je element prázdný, je hned ukončen

- Elementy se mohou vnořovat ale ne křížit

```
<?xml version="1.0"?>
<dokument>
   <strong>
      <paragraf>
      text text text
      </strong>
   </paragraf>
</dokument>
```

! křížení tagů

Komentáře

Komentáře mají tento tvar:

```
<!--  zde je text
   komentráře -->
```

Komentáře musí následovat až za XML deklarací

Komentáře nesmí být uvnitř tagů

```
<dokument <!--
komentář 1 --> >
</dokument>
```

```
<?xml version="1.0"?>
<!-- komentář 1 -->
<dokument>
   <paragraf>
   text text text
    blablabla
   </paragraf>
   <oddelovac />
   <paragraf>
   <!-- komentář 2 -->
   další text
   blablabla
   </paragraf>
</dokument>
```


Znaky

- Pokud chceme použít v datech uložených v XML souborech znaky, které mají řídící význam, musíme je kódovat:

&amp;	&
&lt;	<
&gt;	>
&quot;	“
&apos;	‘

Znaky II

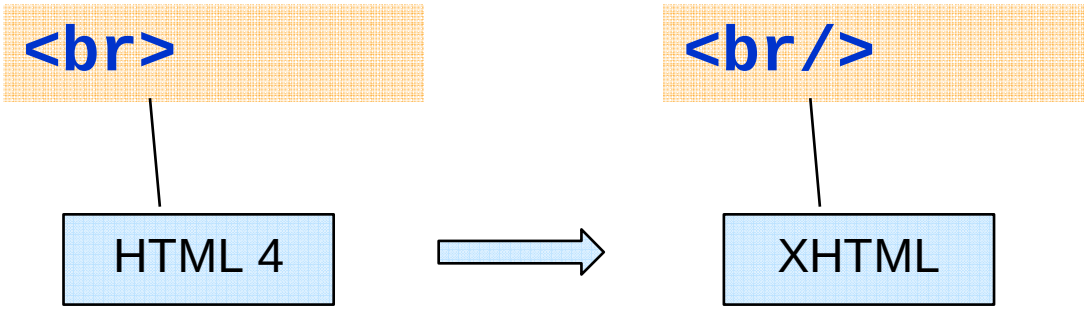
- Při vkládání velkých objemů dat může být kódování nepohodlné
- Řešení: CDATA

```
<dokument>
<![CDATA[
tady si můžu dělat co chci a je to ok < blabla>
]]>
</dokument>
```

Tagy

- Tagy = značky mají jméno
- Jméno musí začínat písmenem nebo znakem _
- Jsou case-sensitive!

- Všechny tagy musí být uzavřené
- Pokud tag není párový, musí se uzavřít sám v sobě



Atributy

- Tagy mohou mít atributy
- Tag může mít 0 a více atributů

```
<?xml version="1.0"?>
<dokument>
   <paragraf jazyk="CZ" zarovnani="vlevo">
   text text text
   </paragraf>
   <oddelovac />
   <paragraf jazyk="EN">
   další text blablabla
   </paragraf>
</dokument>
```

Atributy II

- Pravidla pro jména atributů jsou stejná jako pro jména tagů
- Hodnoty atributů jsou uzavřeny v “ nebo v ‘
- Hodnoty nemají typ, jsou to prostě řetězce

Well – formed dokumenty

XML dokument je well – formed, pokud:

- Má na začátku XML deklaraci
- Elementy obsahující data mají začáteční i koncový tag
- Nepárové elementy jsou ukončeny ***/>***
- Elementy se nesmí křížit
- Hodnoty atributů musí být uzavřeny v “ nebo ‘
- Znaky < a & mohou být použity jenom jako významové znaky
- Nevýznamové znaky musí být zakódovány

Formátování - CSS

- XML definuje obsah, ne formátování
- Formátování je obsaženo v jiné struktuře: CSS

```
<?xml version="1.0"?>
<?xml-stylesheet type=”text/css” href=”formatovani.css”?>
<dokument>
   <paragraf>
   text text text
   </paragraf>
   <oddelovac />
   <paragraf>
   další text blablabla
   </paragraf>
</dokument>
```

Computer Graphics Group

viz soubor.xml,
formatovani.css

Zobrazení

Prohlížeč neví, jak dokument zobrazit

```
<?xml version="1.0" ?>
- <dokument>
    <paragraf>text text text</paragraf>
    <oddelovac />
    <paragraf>dalsi text blablabla</paragraf>
  </dokument>
```

S CSS

text text text
dalsi text blablabla

```
paragraf {
  font-size: 3em;
  display: block;
  background-color:
yellow;
}
```

Výsledek vykreslení XML dokumentu

Pravidla pro zobrazení v souboru formatovani.css

Definice struktury dokumentu - DTD

Definuje se pomocí pravidel napsaných v DTD souboru

DTD = Document Type Declaration

DTD specifikuje gramatiku XML souboru

Parsery umí kontrolovat proti této gramatice

XML je well formated pokud neporušuje základní pravidla formátování

XML je valid pokud splňuje pravidla příslušné gramatiky

DTD ELEMENT

ELEMENT definuje strukturu a pořadí, počty elementů v dokumetu

obecná definice

<!ELEMENT jméno_elementu (obsah_elementu)>

př.:

<!ELEMENT paragraf (#PCDATA)>

Element paragraf obsahuje volný text

př.:

<!ELEMENT dokument (paragraf+, oddelovac*)+>

Element dokument obsahuje jeden nebo více paragrafů a 0 nebo více oddělovačů

DTD

Tato direktiva říká, že definice struktury tohoto dokumentu je v souboru dokument.dtd

XML

```
<?xml version="1.0"?>
<!DOCTYPE dokument
SYSTEM "dokument.dtd">
<dokument>
    <paragraf>
    text text text
    </paragraf>
    <oddelovac />
    <paragraf>
    blablabla
    </paragraf>
</dokument>
```

DTD (soubor dokument.dtd)

```
<!ELEMENT dokument
   (paragraf+, oddelovac*)+>

<!ELEMENT paragraf (#PCDATA)>

<!ELEMENT oddelovac EMPTY>
```

paragraf obsahuje volný text, žádný element

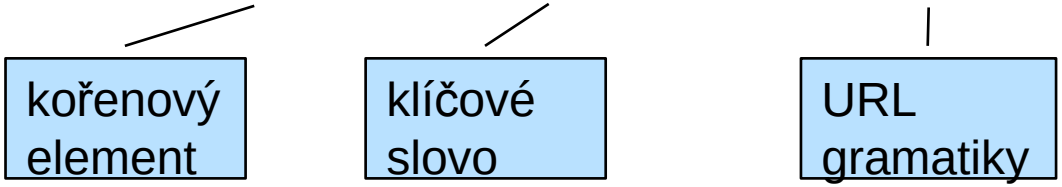
oddělovač neobsahuje nic, je prázdný (empty)

Doctype – externí definice DTD

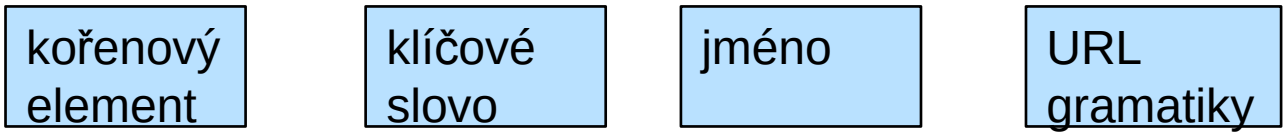
- DTD může být referována jako externí zdroj
- K určení umístění se používá URL
  - může být relativní či absolutní
  - SYSTEM - pro použití jedním autorem nebo lokálním kolektivem
  - PUBLIC – pro veřejné použití, definuje navíc ještě jméno pro DTD
- V definici je jméno kořenového (root) elementu

DTD

```
<?xml version="1.0"?>
<!DOCTYPE dokument SYSTEM "dokument.dtd">
```




```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```



DTD – další možnosti

Element může obsahovat buď jeden, druhý nebo třetí element

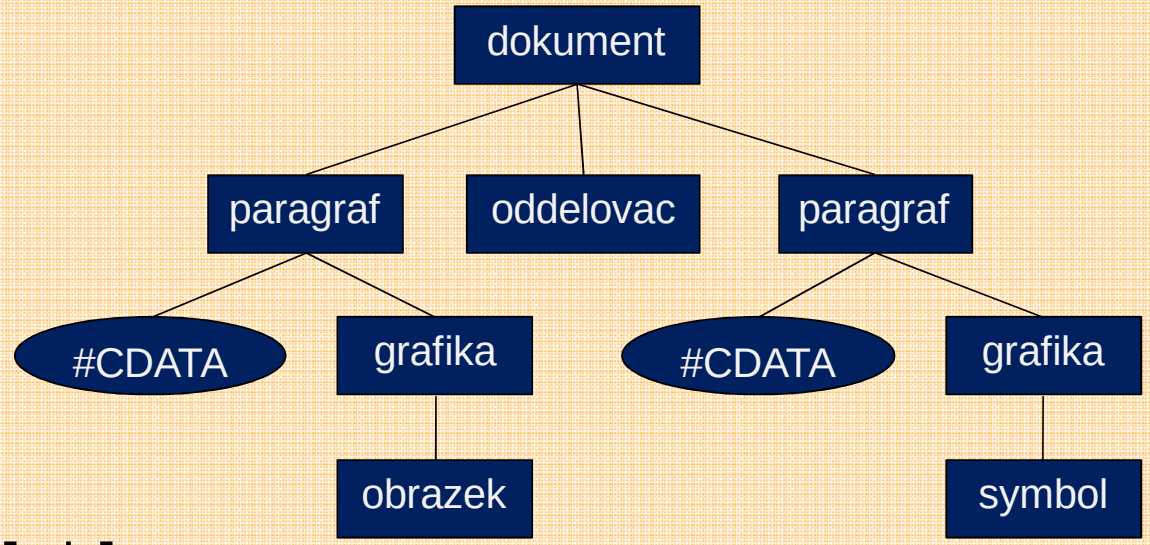
Př. element **`grafika`** obsahuje jeden z těchto elementů:
`obrazek` nebo **`symbol`** nebo **`animace`**

V DTD to vyjádříme takto:

```
<!ELEMENT grafika (obrazek|symbol|animace)>
<!ELEMENT obrazek EMPTY>
<!ELEMENT symbol EMPTY>
<!ELEMENT animace EMPTY>
```

DTD – další možnosti pokačování XML dokument

```
<?xml version="1.0"?>
<!DOCTYPE dokument SYSTEM "dokument_graficky.dtd">
<dokument>
    <paragraf>
      text text text
      <grafika>
        <obrazek/>
      </grafika>
    </paragraf>
    <oddelovac/>
    <paragraf>
      další text blablabla
      <grafika>
      <symbol/>
      </grafika>
    </paragraf>
</dokument>
```



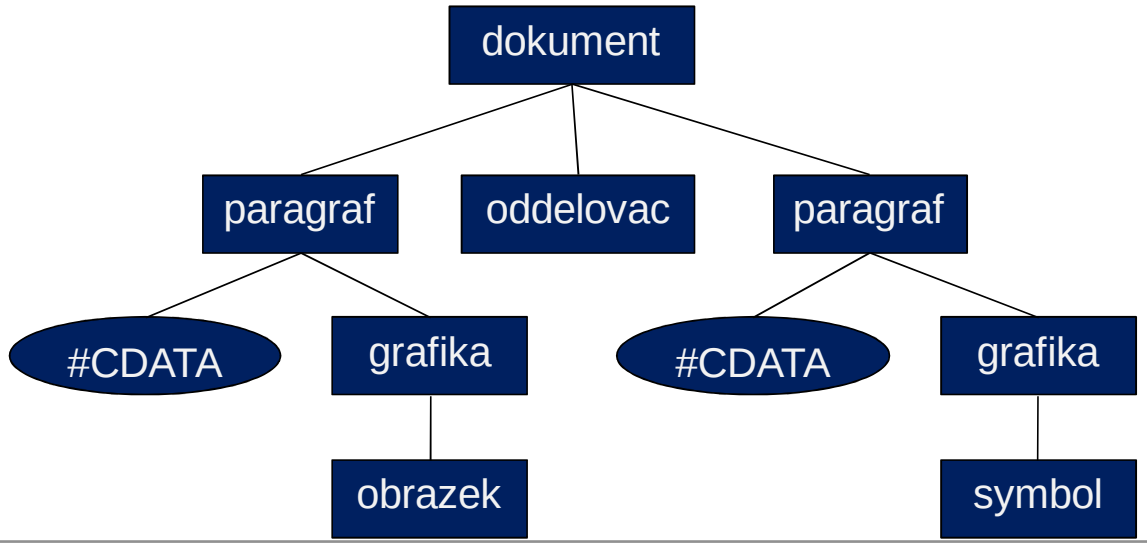
Computer Graphics Group

DTD – další možnosti pokačování DTD

```
<!ELEMENT dokument (paragraf+, oddelovac*)+>
<!ELEMENT paragraf (#PCDATA | grafika)*>
<!ELEMENT oddelovac EMPTY>
<!ELEMENT grafika (obrazek | symbol | animace)>
<!ELEMENT obrazek EMPTY>
<!ELEMENT symbol EMPTY>
<!ELEMENT animace EMPTY>
```

Mix #PCDATA a element

Možnosti



DTD a entity

- Entita je nějaká jednotka, která má svůj identifikátor v XML dokumentu a je možné jí rozvinout

Př.:

```
<!ENTITY CVUT "České vysoké učení technické">
```

jméno

rozvinutí

```
<?xml version="1.0"?>
<!DOCTYPE dokument SYSTEM "entity.dtd">
<dokument>
        <paragraf>
                text text text &CVUT;
                <grafika>
                        <obrazek/>
                </grafika>
        </paragraf>
</dokument>
```

použití entity

Computer Graphics Group

DTD – předdefinované entity

&amp;	&
&lt;	<
&gt;	>
&quot;	“
&apos;	‘

```
<!ENTITY lt "&#38;#60;">
<!ENTITY gt "&#62;">
<!ENTITY amp "&#38;#38;">
<!ENTITY apos "&#39;">
<!ENTITY quot "&#34;">
```

DTD – externí entity

Za entity můžeme prohlásit celý XML soubor.

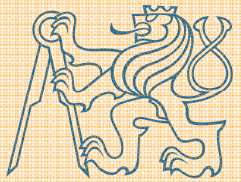
Př.: (soubor podpis.xml)

```
<?xml version="1.0"?>
<podpis>
    <copyright>Martin Klima</copyright>
</podpis>
```

V DTD pak můžeme tento soubor nazvat entitou

```
<!ENTITY SIG SYSTEM "podpis.xml">
```

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE dokument SYSTEM "entity_podpis.dtd">
<dokument>
        <paragraf>
                text text text &SIG;
                <grafika>
                        <obrazek/>
                </grafika>
        </paragraf>
</dokument>
```

DTD a atributy

Předpokládejme, že chceme odstavci přiřadit nějaké atributy a jejich možné hodnoty. Zeleně je označena **default** hodnota. Řekněme, že atribut zarovnání je **povinný**.

- zarovnání: **vlevo**, vpravo, doprostred
- id: identifikátor obsahující jakýkoli řetězec
- radkovani: **1**, 2, 3, ….
- odsazeni: jakékoli číslo, nepovinný atribut

```
<?xml version="1.0"?>
<!DOCTYPE dokument SYSTEM "atributy.dtd">
<dokument>
      <paragraf zarovnani="vlevo" id="p1" radkovani="1">
            text text text
      </paragraf>
</dokument>
```

Atributy musí být zapsány v DTD

Zápis v DTD

```
…
…
…
<!ATTLIST paragraf
    zarovnani CDATA "vlevo"
    id        CDATA  #REQUIRED
    radkovani CDATA "1"
    odsazeni  CDATA #IMPLIED
  >
```

Atributy elementu **paragraf**

Atribut zarovnani má default hodnotu “vlevo”

Atribut id je povinný

Atribut odsazeni je nepovinný a nemá default hodnotu

viz atributy.dtd

DTD – typy atributů

CDATA	jakýkoli text
Výčet	výčet možných hodnot
ID	jednoznačný identifikátor v rámci dokumentu
IDREF	hodnota ID atributu nějakého elementu
IDREFS	více ID elementů oddělených čárkami
ENTITY	jméno entity deklarované v DTD
ENTITIES	jména více entit oddělená čárkami
NMTOKEN	XML jméno (NameChar)+
NMTOKENS	více XML jmen oddělených čárkou

DTD Atributy - výčet

Zpět k požadavku na atribut **zarovnani**

zarovnání: **vlevo**, vpravo, doprostred

Zde nechceme, aby byl přípustný jiný atribut než vyjmenovaný a default je vlevo

```
...
...
<!ATTLIST paragraf
    zarovnani (vlevo | vpravo | doprostred) "vlevo"
    id         CDATA  #REQUIRED
    radkovani CDATA "1"
    odsazeni   CDATA #IMPLIED
  >
```

Výčet možných hodnot atributu zarovnani

Default hodnota atributu zarovnani

viz atributy_lepsi.dtd

Jmenné prostory (namespaces)

- Umožňují používat několik druhů značek v jednom dokumentu
- Značky mohou mít stejná jména, ale díky namespace je dokážeme rozlišit

Namespace - příklad

- Dvě různé tabulky
- Každá vychází z jiné definice
- Liší se tedy strukturou

```
<table>
    <tr>
        <td>Apples</td>
        <td>Bananas</td>
    </tr>
</table>
```

```
<table>
    <name>African Coffee Table</name>
    <width>80</width>
    <length>120</length>
</table>
```

Mohu tyto tabulky dostat do jednoho dokumetu?

- 2 možnosti
  - použít prefix
  - použít namespace

```
<h:table>
   <h:tr>
   <h:td>Apples</h:td>
   <h:td>Bananas</h:td>
   </h:tr>
</h:table>
```

```
<f:table>
   <f:name>African Coffee Table</f:name>
   <f:width>80</f:width>
   <f:length>120</f:length>
</f:table>
```

Namespaces - zápis

- pomocí atributu xmlns:*prefix*
- hodnota je URI jmenného prostoru

Prefix xml a xmlns jsou rezervované

```
<h:table xmlns:h="http://www.w3.org/TR/html4/">
   <h:tr>
   <h:td>Apples</h:td>
   <h:td>Bananas</h:td>
   </h:tr>
</h:table>
```

```
<f:table xmlns:f="http://www.w3schools.com/furniture">
   <f:name>African Coffee Table</f:name>
   <f:width>80</f:width>
   <f:length>120</f:length>
</f:table>
```

Default namespace

- Pokud nadřazenému elementu řeknu, v jakém je jmenném prosturu, jeho potomci jsou v něm také

```
<table xmlns="http://www.w3.org/TR/html4/">
   <tr>
       <td>Apples</td>
       <td>Bananas</td>
   </tr>
</table>
```

```
<table xmlns="http://www.w3schools.com/furniture">
   <name>African Coffee Table</name>
   <width>80</width>
   <length>120</length>
</table>
```

Namespace – definice na začátku souboru

```
<?xml version=”1.0”?>
<xsl:stylesheet
      xmlns:xsl="http://www.w3.org/XSL/Transform/1.0"
      xmlns:html="http://www.w3.org/TR/REC-html40">
  <xsl:template match="PERIODIC_TABLE">
    <html:html>
      <xsl:apply-templates/>
    </html:html>
  </xsl:template>
  <xsl:template match="ATOM">
    <html:p>
      <xsl:apply-templates/>
    </html:p>
  </xsl:template>
</xsl:stylesheet>
```

Definuje použité namespace

Návrh XML dokumentů

Používat atributy nebo elementy?

- Samotná specifikace XML toto neurčuje
- V atributu i elementu - stejná informace
- Platí zde analogie jako při návrhu databází (entita, atribut)?

Doporučení – v XML používejte elementy:

- *Případná rozšíření XML dokumentu v budoucnu:*
- Atributy nemohou obsahovat více hodnot (rozlišitelných přímo na úrovni XML)
- Atributy se nedají rozšiřovat
- Pomocí atributů se nedají popisovat struktury

Návrh XML dokumentů - pokračování

- Hodnoty atributů ke obtížnější testovat proti DTD
- Velké množství atributů ztěžuje přehlednost a údržbu dokumentu

- ID datových objektů, případně metadata, která nemají přímý vztah k samotným datům je vhodné ukládat v atributech

- Záleží na konkrétním dokumentu

Reference

http://www.xml.com/

http://www.biztalk.org/

http://www.xml.org/

http://www.oasis-open.org/cover/

http://zvon.vscht.cz/

http://www.xmlsoftware.com/

http://www.w3.org/XML/

http://www.wapserver.cz/