

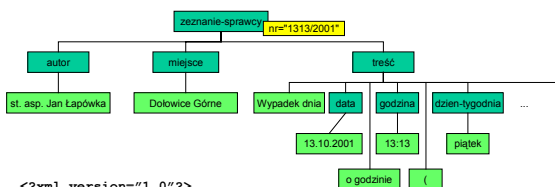
## Definiowanie typów dokumentów Część 1. DTD

## Jak wygląda XML?

```
<?xml version="1.0"?>
<zeznanie-sprawcy nr="1313/2001">
  <autor>st. asp. Jan Łapówka</autor>
  <miejsce>Dołowice Górne</miejsce>
  <treść>Wypadek dnia
  <data>13.10.2001</data>
  o godzinie <godzina>13:13</godzina>
  (<dzien-tygodnia>piątek
  </dzien-tygodnia>) miał miejsce nie
  z mojej winy. <poszkodowany>Alojzy
  M.</poszkodowany> nie miał żadnego
  pomysłu w którą stronę uciekać, więc
  go przejechałem.</treść>
</zeznanie-sprawcy>
```

Deklaracja XML  
Element główny  
Atrybut  
Element  
Znacznik początkowy  
Znacznik końcowy  
Zawartość tekstowa

## Struktura logiczna dokumentu XML



```
<?xml version="1.0"?>
<zeznanie-sprawcy nr="1313/2001">
  <autor>st. asp. Jan Łapówka</autor>
  <miejsce>Dołowice Górne</miejsce>
  <treść>Wypadek dnia <data>13.10.2001</data>
  o godzinie <godzina>13:13</godzina> (<dzien-tygodnia>piątek
  </dzien-tygodnia>) miał miejsce nie z mojej winy.
  <poszkodowany>Alojzy M.</poszkodowany> nie miał żadnego
  pomysłu w którą stronę uciekać, więc go przejechałem.</treść>
</zeznanie-sprawcy>
```

## Podstawy składni XML

- Deklaracja XML:  
`<?xml version="1.0" encoding="UTF-8" standalone="no"?>`
- Znaczniki:  
`<tag attributenam="attribute-value">  
</tag>`
- Znaczniki elementu pustego:  
`<br></br>  
<br/>`

## Definiowanie języków

- XML, SGML – metajęzyki.
- Definiowanie języków (zastosowań, struktury dokumentów, typów dokumentów):
  - określanie zestawu dopuszczalnych elementów, atrybutów, ...
  - definiowanie dopuszczalnej zawartości elementów (tekst, inne elementy),
  - przypisywanie atrybutów do elementów,
  - ...
- Metody definiowania struktury:
  - dokument XML bez formalnej definicji struktury,
  - DTD – Document Type Definition,
  - XML Schema (rekomendacja W3C z 2 maja 2001),
  - Relax NG.

## Poprawność dokumentów

- Dokument XML poprawny składniowo (ang. *well-formed*):
  - każdy element musi być zamknięty,
  - nie ma nakładających się elementów,
  - wartości atrybutów w apostrofach lub cudzysłowach,
  - ...
- Dokument XML poprawny strukturalnie (ang. *valid*):
  - struktura dokumentu zgodna ze strukturą zdefiniowaną w definicji typu dokumentu,
  - obecne wszystkie wymagane atrybuty.
- Dokument SGML: obowiązkowa definicja struktury – DTD.

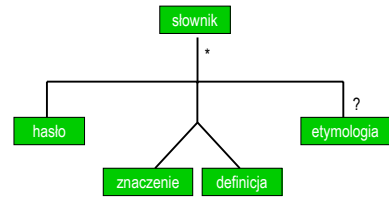


## Budowanie modeli zawartości

- Znana notacja:
  - + jeden lub więcej
  - \* zero lub więcej
  - , sekwencja
  - | alternatywa
  - ? opcjonalny
- Fragment zawartości (*content particle*) – wyrażenie zbudowane z połączonych spójnikami:
  - nazw elementów,
  - #PCDATA,
  - fragmentów zawartości.

## Fragmety zawartości – przykład

```
<!ELEMENT słownik  
(hasło, (znaczenie | definicja), etymologia?)* >
```



## Typy atrybutów

- CDATA ciąg znaków
- NMTOKEN ciąg znaków mogących występować w nazwach atrybutów i elementów
- NMTOKENS ciąg NMTOKEN oddzielanych spacjami
- ID identyfikator unikalny w dokumencie
- IDREF wskaźnik do ID innego elementu
- IDREFS ciąg IDREF oddzielany spacjami
- ENTITY nazwa encji (musi być zadeklarowana)
- ENTITIES ciąg ENTITY oddzielany spacjami
- (a | b | c) typ wyliczeniowy

## Rodzaje atrybutów

- #REQUIRED
- #IMPLIED
- #FIXED  
<!ATTLIST NIP OPIS CDATA #FIXED  
"Numer Identyfikacji Podatkowej">
- Wartość domyślna  
<!ATTLIST wiersz biały (tak|nie) "nie">

## Normalizacja wartości atrybutów

- Upraszcza tworzenie dokumentów.
- Kroki normalizacji:
  - usunięcie otaczających cudzysłowów lub apostrofów,
  - zastąpienie referencji do znaków przez odpowiednie znaki,
  - rozwinięcie encji ogólnych,
  - zastąpienie znaków końca wiersza spacjami,
  - dla atrybutów typu NMTOKEN, NMTOKENS, ENTITY, ENTITIES, ID, IDREF, IDREFS:
    - usunięcie wiodących i końcowych spacji,
    - zastąpienie ciągów spacji pojedynczymi spacjami.

## Fizyczna struktura dokumentu: encje

- Encja (entity):
  - fizyczna reprezentacja obiektu informacyjnego w systemie, uogólnienie pojęcia *plik*,
  - jednostka fizycznej budowy dokumentu, uogólnienie pojęcia *makro*.
- Dokument ≠ plik ≠ encja:
  - encja dokumentu (*document entity*),
  - zawartość dokumentu może znajdować się w wielu encjach (reprezentowanych np. przez pliki).

## Encje predefiniowane

&amp;        &  
&lt;         <  
&gt;         >  
&apos;       '   
&quot;       "

## Encje wewnętrzne i zewnętrzne

- Encje wewnętrzne:
  - DTD:  
`<!ENTITY xml "<term>Extensible Markup Language</term>">`
  - Instancja dokumentu:  
**Metajęzyk &xml; wywodzi się z SGML-a.**
- Encje zewnętrzne:
  - DTD:  
`<!ENTITY intro SYSTEM "intro.xml">`  
`<!ENTITY chap1 SYSTEM "chapter1.xml">`  
`<!ENTITY chap2 SYSTEM "chapter2.xml">`
  - Instancja dokumentu:  
`<book>`  
    &intro;  
    &chap1;  
    &chap2;  
`</book>`

## Jak odwoływać się do encji zewnętrznych

- Identyfikatory zewnętrzne:
  - Identyfikator systemowy:  
`SYSTEM "docbook.dtd"`
  - Identyfikator publiczny:  
`PUBLIC "-//OASIS//DTD DocBook V3.1//EN"`
- Odzworowanie identyfikatorów publicznych na systemowe: plik catalog.

## Plik catalog

- Format OASIS (odziedziczony po SGML-u):  
`PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"`  
    "html40-1/html40-1.dtd"  
`PUBLIC "-//ArborText//DTD Article 950601//EN"`  
    "article/article.dtd"
- Format XCatalog:  
`<XMLCatalog>`  
    <Map PublicId= "-//W3C//DTD HTML 4.0 Transitional//EN"  
        HRef="html40-1/html40-1.dtd" />  
    <Map PublicId= "-//ArborText//DTD Article 950601//EN"  
        HRef="article/article.dtd" />  
`</XMLCatalog>`

## Encje parametryczne

- Wykorzystywane w DTD, np:  
`<!ENTITY % inline "(#PCDATA | emph | keyword | name)*">`  
`<!ELEMENT para %inline;>`  
`<!ELEMENT list (list-item)*>`  
`<!ELEMENT list-item %inline;>`  
`<!ELEMENT definition (%inline; | defined-word)*>`
- Zewnętrzne encje parametryczne – modularyzacja DTD, np:  
`<!ENTITY % calstbls PUBLIC`  
    "//ArborText//ELEMENTS CALS Table Structures//EN"  
`%calstbls;`

## Encje nieprzetwarzane

- Odwołania do obiektów nieprzetwarzanych przez parser XML (grafiki, dźwięki, plików binarnych, itp.)
- W DTD:
  - deklaracja notacji:  
`<!NOTATION GIF SYSTEM "gifmagic.exe">`
  - deklaracja atrybutu encyjnego:  
`<!ELEMENT pic EMPTY>`  
`<!ATTLIST pic name ENTITY #REQUIRED>`
  - deklaracja encji nieprzetwarzanej:  
`<!ENTITY logo SYSTEM "logo.gif" NDATA "GIF">`
- W instancji dokumentu:
  - odwołanie do encji:  
`<pic name="logo"/>`

## Encje – podsumowanie

	Encje ogólne		Encje parametryczne	
	Encje przetwarzane	Encje nieprzetwarzane	Encje przetwarzane	Encje nieprzetwarzane
Encje wewnętrzne	✓	✗	✓	✗
Encje zewnętrzne	✓	✓	✓	✗

## Gdzie umieścić DTD?

- W encji dokumentu:

```

<!DOCTYPE wiersz [
  <!ELEMENT wiersz (autor, tytuł, zwrotka*)>
  <!ATTLIST wiersz biały (tak|nie) "nie">
  <!ELEMENT autor (#PCDATA)>
  <!ELEMENT tytuł (#PCDATA)>
  <!ELEMENT zwrotka (wers)*>
  <!ELEMENT wers (#PCDATA)>
]>
<wiersz>
  <autor>William Shakespeare</autor>
  <tytuł>Sonet CCII</tytuł>
  <zwrotka>...</zwrotka>
</wiersz>
    
```

## Gdzie umieścić DTD?

- W zewnętrznej encji:

```

- wiersz.dtd
<!DOCTYPE wiersz [
  <!ELEMENT wiersz (autor, tytuł, zwrotka*)>
  <!ATTLIST wiersz biały (tak|nie) "nie">
  <!ELEMENT autor (#PCDATA)>
  <!ELEMENT tytuł (#PCDATA)>
  <!ELEMENT zwrotka (wers)*>
  <!ELEMENT wers (#PCDATA)>
]>

- sonet.xml
<!DOCTYPE wiersz SYSTEM "wiersz.dtd">
<wiersz>
  <autor>William Shakespeare</autor>
  <tytuł>Sonet CCII</tytuł>
  <zwrotka>...</zwrotka>
</wiersz>
    
```

## Gdzie umieścić DTD?

- Połączenie obu metod:

```

- wiersz.dtd
<!DOCTYPE wiersz [
  <!ELEMENT wiersz (autor, tytuł, zwrotka*)>
  <!ATTLIST wiersz biały (tak|nie) "nie">
  <!ELEMENT autor (#PCDATA)>
  <!ELEMENT tytuł (#PCDATA)>
  <!ELEMENT zwrotka (wers)*>
  <!ELEMENT wers (#PCDATA)>
]>
- sonet.xml
<!DOCTYPE wiersz SYSTEM "wiersz.dtd" [
  <ENTITY ws "William Shakespeare">
  <!ATTLIST wiersz rodzaj #IMPLIED>
]>
<wiersz rodzaj="sonet">
  <autor>ws</autor>
  <tytuł>Sonet CCII</tytuł>
  <zwrotka>...</zwrotka>
</wiersz>
    
```

Zewnętrzny podzbiór DTD

Wewnętrzny podzbiór DTD

## Zewnętrzny i wewnętrzny podzbiór DTD

- Zewnętrzny podzbiór DTD: deklaracje wspólne dla wszystkich dokumentów danego typu:
  - elementy, atrybuty,
  - encje parametryczne.
- Wewnętrzny podzbiór DTD: deklaracje lokalne dla dokumentu:
  - deklaracje encji,
  - deklaracje notacji.
- Zaawansowane możliwości wewnętrznego podzbioru DTD:
  - przedefiniowywanie encji parametrycznych,
  - przedefiniowywanie atrybutów,
  - dodawanie nowych atrybutów,
  - sekcje warunkowe.

## Zaawansowana składnia XML

- Komentarz:
 

```
<!-- komentarz -->
```
- Instrukcja przetwarzania:
 

```
<?target processing-instruction-body?>
```
- Sekcja CDATA:
 

```
<![CDATA[dowolny <tekst "nieprzetwarzany & przez [parser]]]>
```
- Odwolania do znaków:
 

```
&#161;
&#xA1;
```

kody zgodne ze standardem ISO/IEC 10646.

## Unicode

- Światowy standard kodowania narodowych znaków przy pomocy dwubajtowych par:
  - podzbiór ISO/IEC 10646.
- Odmiany:
  - UTF-7,
  - UTF-8 (pierwsze 128 - ASCII),
  - UTF-16.
- Obowiązkowy standard dla dokumentów XML:
  - każde narzędzie XML-owe musi wspierać przynajmniej UTF-8.

## Gdzie szukać dalej

- *DTD Tutorial*
  - 🔗 [www.xmlfiles.com/dtd](http://www.xmlfiles.com/dtd)
- Arbortext, *If You Can Name It, You Can Claim It!*
  - 🔗 [www.arbortext.com/html/issue\\_three.html](http://www.arbortext.com/html/issue_three.html)
- Megginson, D., *Structuring XML Documents*, Prentice Hall, 1998
- Dmoch, A., Ziolo, Sz., *Encje i pliki catalog, czyli fizjologia XML-a*, Software 2.0 nr 6/2004, Wydawnictwo Software

