METAJĘZYKI

2

mgr inż. Robert Berezowski

e-mail: beny@ie.tu.koszalin.pl

Ul. Śniadeckich 2

Pokój 223A

Politechnika Koszalińska

Wydział Elektroniki i Informatyki

Katedra Inżynierii Komputerowej

XML – wstęp

- XML to zbiór zasad do definiowania znaczników „w miarę potrzeb”;
- Znacznik – dodatkowa informacja do tekstu (znaków), pozwalająca go przetworzyć później w określony sposób;
- Znaczniki w XML można nazywać i używać w zależności od potrzeby budowanej aplikacji.
- Znaczniki tworzą języki znacznikowe (metajęzyki) dla konkretnych zastosowań;
- Duża grupa ludzi używająca jeden język może doprowadzić do rozwoju aplikacji z nim związanych (MathML, SVG, CML);
  - aplikacje;
  - przeglądarki;
  - wtyczki;

XML – wstęp

- Początek znacznika oznaczany jest znakiem mniejszości < ;
- Symbole:
  - większości >,
  - cudzysłów pojedynczy ‘,
  - cudzysłów podwójny ” ,

    traktowane są jako elementy znaczników
- Przykłady znaczników: **<strona>, <list>, <mecz>, <numer.strony>**
- Samoopisujące się dokumenty;
  - **samodokumentujące się;**
- Dane i struktura danych;

Przykładowy dokument XML

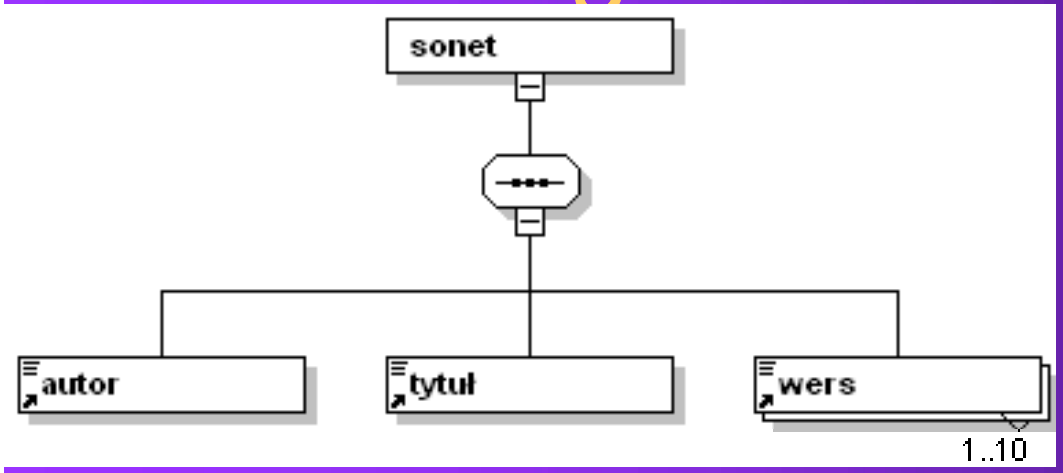
```
<?xml version="1.0" standalone="yes" encoding="UTF-16"?>
<sonet>
    <autor plec="M">Adam Mickiewicz </autor>
    <linia/>
    <!-- komentarz Sonet XIII  -->
    <tytuł>Sonet XIII</tytuł>
    <zwrotka numer="1">
      <wers>Drżąc muślimin całuje stopy twej opoki,</wers>
      <wers>Maszcie krymskiego statku, wielki  Czatyrdahu!</wers>
      <wers> O minarecie świata! O gór padyszachu!</wers>
      <wers> Ty, nad skały poziomu uciekłszy w obłoki.</wers>
      <wers> Siedzisz sobie pod bramą niebios, jak wysoki </wers>
      <wers> Gabryjel pilnujący edeńskiego gmachu; </wers>
    </zwrotka>
</sonet>
```

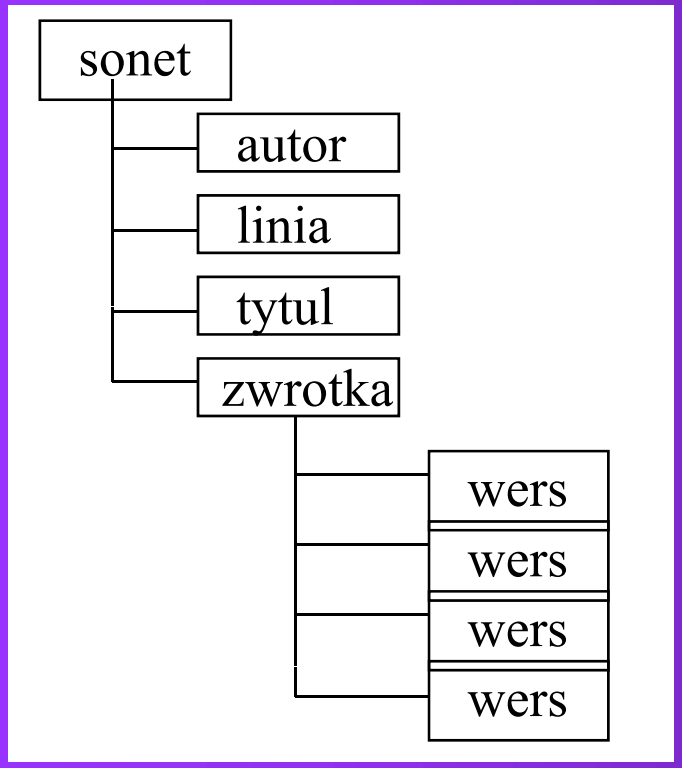
Obraz w przeglądarce IE5.0

```
<?xml version="1.0" standalone="yes" encoding="UTF-16"?>
– <sonet>
    <autor plec="M">Adam Mickiewicz</autor>
    <linia/>
    <!-- komentarz Sonet XIII  -->
    <tytul>Sonet XIII</tytul>
  – <zwrotka numer="1">
      <wers>Drzac muslimin caluje stopy twej opoki </wers>
      <wers>Maszcie krymskiego statku, wielki
    Czatyrdahu </wers>
      <wers>O minarecie swiata! O gor padyszachu </wers>
      <wers>Ty, nad skaly poziomu ucieklszy w obloki </wers>
      <wers>Siedzisz sobie pod brama niebios, jak
    wysoki</wers>
      <wers>Gabryjel pilnujacy edenskiego gmachu </wers>
    </zwrotka>
</sonet>
```

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

Struktura logiczna





- Korzeniem całej struktury jest element <sonet>;
- Wszystkie pozostałe elementy mieszczą się jedne w drugich;
- Element zawierający inny nazywamy rodzicem (np. <sonet>, <zwrotka>);
- Element znajdujący się wewnątrz nazywamy potomkiem (np. <autor>, <zwrotka>, <wers>);
- Elementy tego samego rodzica nazywamy pokrewnymi (<autor>, <linia>, <tytul>, <zwrotka>);

Przykładowy dokument XML

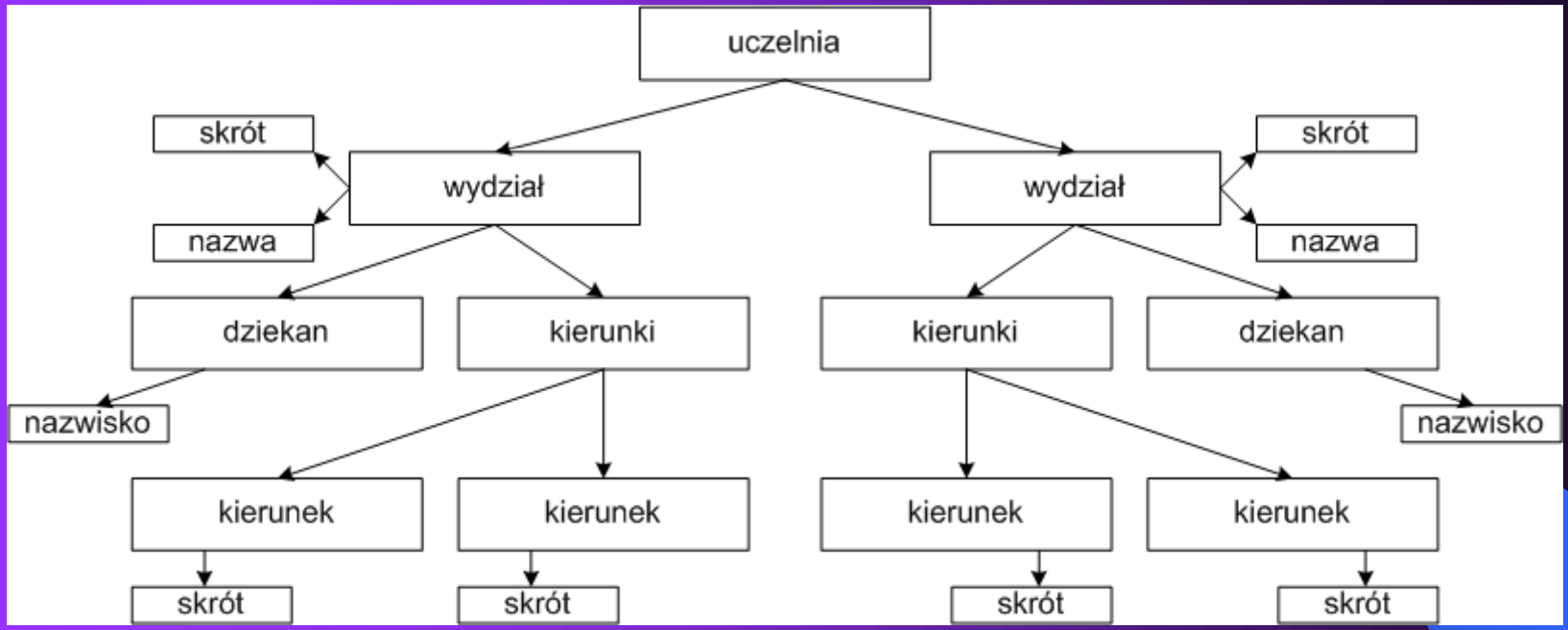
```
<?xml version="1.0" standalone="yes" encoding="UTF-16"?>
<uczelnia>
    <!- Wydziały uczelni  -->
    <wydział skrót="EII" nazwa="elektroniki i Informatyki">
    <dziekan nazwisko="Jacek Placek"/>
     <kierunki>
       <kierunek skrót="KIK">Inżynieria komputerowa</kierunek>
       <kierunek skrót="KSE">Systemy elektroniczne</kierunek>
     </kierunki>
    </wydział>
    <wydział skrót="EIZ" nazwa="Ekonomii i zarządzania">
    <dziekan nazwisko="Placek Jacek"/>
     <kierunki>
       <kierunek skrót="KME">Makroekonomia</kierunek>
       <kierunek skrót="KHZ">Handel zagraniczny</kierunek>
     </kierunki>
    </wydział>
</uczelnia>
```

Obraz w przeglądarce IE5.0

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <uczelnia>
    <!-- Wydzialy uczelni    -->
  - <wydział skrót="EII" nazwa="elektroniki i Informatyki">
      <dziekan nazwisko="Jacek Placek" />
    - <kierunki>
        <kierunek skrót="KIK">Inżynieria komputerowa</kierunek>
        <kierunek skrót="KSE">Systemy elektroniczne</kierunek>
      </kierunki>
    </wydział>
  - <wydział skrót="EIZ" nazwa="Ekonomii i zarządzania">
      <dziekan nazwisko="Placek Jacek" />
    - <kierunki>
        <kierunek skrót="KME">Makroekonomia</kierunek>
        <kierunek skrót="KHZ">Handel zagraniczny</kierunek>
      </kierunki>
    </wydział>
  </uczelnia>
```

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

Struktura logiczna



- Korzeniem całej struktury jest element <uczelnia>;
- Rodzic <uczelnia>, <wydział>, <dziekan>, <kierunki>;
- Potomek <wydział>, <dziekan>, <kierunki>, <kierunek>
- Pokrewne <dziekan>, <kierunki>

<?xml version="1.0" standalone="yes" encoding=„UTF-16"?>

- Deklaracja XML:
  - Określa co znajduje się dalej w kodzie XML;
  - Z którą wersją standardu XML dokument jest zgodny;
  - Czy dokument jest samodzielny czy wymaga dodatkowego pliku (DTD) w celu zinterpretowania jego zawartości (struktury);
  - Jakie kodowanie zostało użyte przy tworzeniu dokumentu
- Powyższa deklaracja jest instrukcją przetwarzania czyli instrukcją dla procesora XML;

Instrukcje przetwarzania

- Nie wchodzą w skład specyfikacji XML
  - Przeznaczone dla różnych procesorów
- Przykład:
  - <?xml-stylesheet… ?> - instrukcja łącząca dokument xml z arkuszem stylów, rozumiana przez procesory XML używane w IE, Netscape

<sonet> ... </sonet> <uczelnia> … </uczelnia>

- Element główny;
  - Każdy dokument XML może posiadać tylko jeden element główny;
  - Wszystkie pozostałe elementy muszą być w nim zawarte;
  - Jeżeli jeden element zawarty jest w drugim, to musi być w nim zawarty całkowicie;

```
<poezja>
  <sonet>
    <tytul>
      <autor>
        …
      </tytul>
    </autor>
  </sonet>
</poezja>
```

```
<sonet>
…
</sonet>
<sonet>
…
</sonet>
```

```
<poezja>
  <sonet>
    …
  </sonet>
  <sonet>
    …
  </sonet>
</poezja>
```

<linia/>

- Element pusty
- Dwa sposoby oznaczania:
  - <linia/>
  - <linia></linia>

<!-- komentarz Sonet XIII -->
<!--Wydziały uczelni -->

- Znak <!-- oznacza początek, a znak --> koniec komentarza;

XML – wstęp

- Nazwy znaczników muszą zaczynać się literą, znakiem podkreślenia lub dwukropkiem, a dalej mogą występować cyfry, litery, podkreślenia, znak minus i kropka
- W przypadku użycia jednego z wcześniej podanych znaków jako znaku zwykłego należy go odpowiednio „zacytować”:
  - & – &
  - ‘ – '
  - > – >
  - < – <
  - „ – "

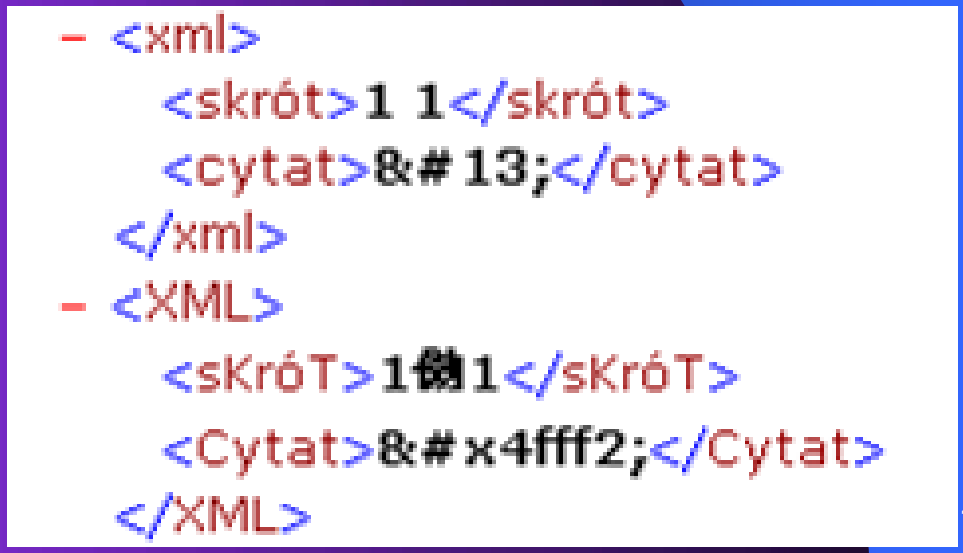
```
<skrót>&amp;</skrót>
 <cytat>&amp;amp;</cytat>
<skrót>&apos;</skrót>
 <cytat>&amp;apos;</cytat>
<skrót>&gt;</skrót>
 <cytat>&amp;gt;</cytat>
<skrót>&lt;</skrót>
 <cytat>&amp;lt;</cytat>
<skrót>&quot;</skrót>
 <cytat>&amp;quot;</cytat>
```

```
- <uczelnia>
  - <xml>
      <skrót>&</skrót>
      <cytat>&amp;</cytat>
    </xml>
  - <xml>
      <skrót>'</skrót>
      <cytat>&apos;</cytat>
    </xml>
  - <xml>
      <skrót>></skrót>
      <cytat>&gt;</cytat>
    </xml>
  - <xml>
      <skrót><</skrót>
      <cytat>&lt;</cytat>
    </xml>
  - <xml>
      <skrót>"</skrót>
      <cytat>&quot;</cytat>
    </xml>
</uczelnia>
```

XML – wstęp

- Odwołania do znaków:
  - `&#13; (spacja)`
  - `&#xD;`
- Znaczniki w XML zorientowane są na zawartość, a nie na wygląd;
- Wszystko co nie jest znacznikiem jest treścią;
- Ostateczny wygląd określony jest przez dołączone style;
- XML rozróżnia małe i duże litery;

```
<xml>
 <skrót>1&#13;1</skrót>
 <cytat>&amp;#13;</cytat>
</xml>
<xml>
 <skrót>1&#x4ff2;1</skrót>
 <cytat>&amp;#x4fff2;</cytat>
</xml>
```



<autor plec="M">
<zwrotka numer="1">

- Znaczniki (autor, zwrotka) mogą zawierać jeden lub więcej atrybutów (plec, numer);
  - para: nazwa i wartość
  - nazwie znakiem = przypisana wartość
    - <element atrybut(nazwa)=”wartość”>
  - Użycie cudzysłowów jest obowiązkowe;
  - Cudzysłowy mogą być pojedyncze ’ lub podwójne ”
- Atrybuty można podawać jedynie w znaczniku początkowym; Atrybut
- Atrybuty mogą być (przy projektowaniu struktury):
  - Opcjonalne;
  - Obowiązkowe;

<sonet numer=14 numer=1>

<sonet numer=14>

<sonet numer=”14’dec’ ”>

<sonet numer(dec)=”14”>

<sonet numer=”14”>
…
</sonet numer=”14”>

ATRYBUTY

- Atrybuty – Elementy mogą być stosowane wymiennie
- Duża liczba atrybutów utrudnia czytanie dokumentu
- Trudność utworzenia struktury
- Nazwy atrybutów nie mogą się powtarzać

Dokumenty XML

- **Dokumentem XML może być:**
  - Plik;
  - Zbiór przechowywany w bazie danych;
  - Zbiór tworzony w pamięci, przez aplikację;
  - Zestaw kilku odrębnych plików;
- **Ogólnie dokument XML składa się z tak zwanych encji**

Dokument XML poprawnie sformułowany (ang. well-formed)

- **Dokument XML poprawny składniowo**
  - ~~deklaracja xml (1 linia) – łączenie dokumentów w 1~~
  - istnienie co najmniej jednego elementu w strukturze (główny),
  - każdy element musi być zamknięty,
  - nie ma nakładających się elementów,
  - nazwy atrybutów nie mogą się powtarzać,
  - wartości atrybutów w apostrofach lub cudzysłowach,
  - czy encje są prawidłowo zdefiniowane
    - predefiniowane
    - zewnętrzne (w DTD)

Dokument XML poprawnie sformułowany (ang. well-formed)

- Parsery i przeglądarki (np.IE6.0) błędnie sformułowany kod XML natychmiast zgłosi jako błąd składniowy;
- Pisanie kodu XML podobne jest do pisania w jakimś języku programowania:
  - Pisanie kodu;
  - „kompilowanie” kodu;
  - Znajdowanie i poprawianie błędów;

Zastosowania XML (definiowanie języków)

- **Definiowanie języków (zastosowań, struktury dokumentów – encyklopedia, kodeks, raport):**
  - określanie zestawu dopuszczalnych elementów, atrybutów,
  - definiowanie dopuszczalnej zawartości elementów (tekst, inne elementy),
  - przypisywanie atrybutów do elementów,
- **Metody definiowania struktury:**
  - dokument XML bez określonej struktury,
  - DTD – Document Type Definition,
  - XML Schema

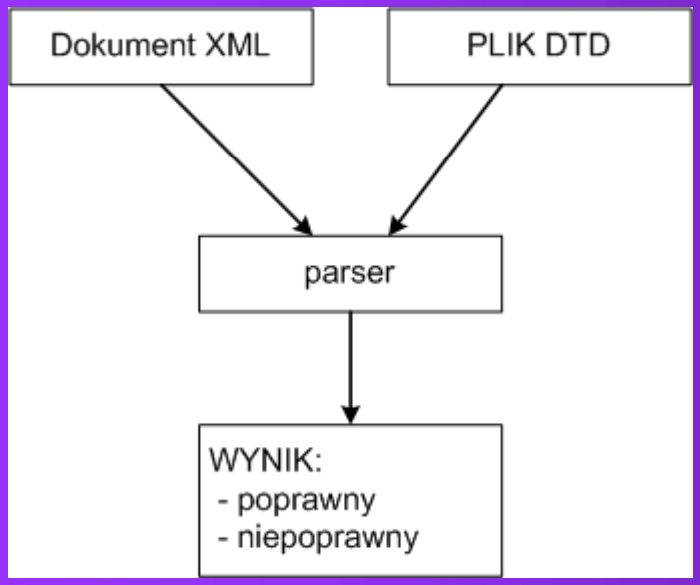
Dokument XML poprawny strukturalnie (ang. valid)

- **Dokument XML poprawny strukturalnie:**
  - struktura dokumentu zgodna ze strukturą zdefiniowaną w definicji typu dokumentu / XMLSchema,
  - obecne wszystkie wymagane atrybuty.
  - Zawartość elementów zgodna z założeniem.
- **Programy do walidacji:**
  - DTD – xmlvalid.exe, XMLShell, InfoxFree,
    - http://www.stg.brown.edu/service/xmlvalid/
  - XMLSchema – ScheamtronValidator, XMLShell, InfoxFree

Zalety istnienia DTD (struktury)

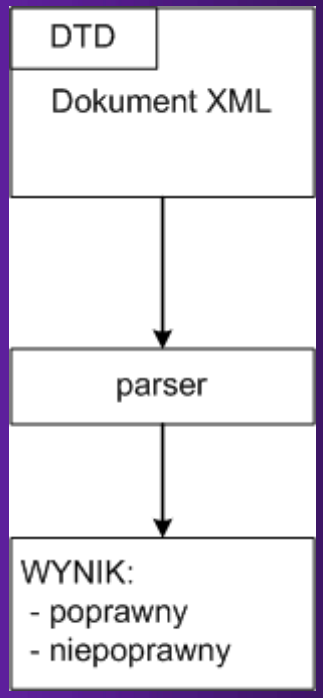
- DTD może sprawdzać poprawność dokumentu XML co pozwala unikać błędów w fazie tworzenia dokumentu
- DTD jest opisem dokumentu dzięki czemu różne osoby mogą określać jakie dane będą im potrzebne i jak je pobierać;
- Można automatycznie konwertować jedne formaty dokumentów na inne;
- Są pewnego rodzaju dokumentacją dokumentów XML
- DTD może wspomagać twórców dokumentów, np. Mogą podpowiadać użycie odpowiednich znaczników w odpowiednim miejscu dokumentu.
- Pozwala wielu osobom tworzyć wiele jednakowych (strukturalnie) dokumentów

Proces sprawdzania poprawności struktury



DTD w pliku zewnętrznym

DTD wewnątrz dokumentu XML



Deklaracja typu dokumentu

- DTD zawiera opis dopuszczalnych struktur dokumentu - język opisu dokumentu;
- DTD ma postać:
  - <!DOCTYPE nazwa [podzbiór.wewnętrzny]>
  - <!DOCTYPE nazwa SYSTEM [plik.zewnętrzny]>
    - nazwa powinna być taka sama jak nazwa elementu głównego;

```
<?xml version=„1.0”?>
<!DOCTYPE list [...]>
<list>.....</list>
```

```
<?xml version=„1.0”?>
<!DOCTYPE list SYSTEM ”listdtd.dtd”>
<list>.....</list>
```

Przykład DTD

```
<?xml version=”1.0” standalone="yes" encoding=”UTF-16”?>
    <!DOCTYPE list SYSTEM ”sonet.dtd”> -->
<!DOCTYPE sonet [
    <!ELEMENT sonet (autor, linia*, tytul,zwrotka+)>
    <!ELEMENT autor (#PCDATA)>
      <!ATTLIST autor plec (M | K) "M">
    <!ELEMENT linia EMPTY>
    <!ELEMENT tytul (#PCDATA)>
    <!ELEMENT zwrotka (wers+)>
    <!ELEMENT wers (#PCDATA)>                    ]>

<sonet>
    <autor plec="M">Adam Mickiewicz </autor>
    <linia/>
    <!-- komentarz Sonet XIII   -->
    <tytul>Sonet XIII</tytul>
    <zwrotka>
        <wers>Drzac muslimin całuje stopy twej opoki,</wers>
        <wers>Maszcie krymskiego statku, wielki Czatyrdahu!</wers>
    </zwrotka>
</sonet>
```

Przykład DTD

```
<?xml version=”1.0” standalone="yes" encoding=”UTF-16”?>
    <!-- DOCTYPE list SYSTEM ”sonet.dtd”> -->
<!DOCTYPE sonet [
    <!ELEMENT sonet (autor, linia*, tytul,zwrotka+)>
    <!ELEMENT autor (#PCDATA)>
    <!ATTLIST autor plec (M | K) "M">
    <!ELEMENT linia EMPTY>
    <!ELEMENT tytul (#PCDATA)>
    <!ELEMENT zwrotka (wers+)>
    <!ELEMENT wers (#PCDATA)>                ]>
<sonet>
    <autor plec="M">Adam Mickiewicz </autor>
    <linia/>
    <!-- komentarz Sonet XIII  -->
    <tytul>Sonet XIII</tytul>
    <zwrotka>
        <wers>Drzac muslimin całuje stopy twej opoki,</wers>
        <wers>Maszcie krymskiego statku, wielki Czatyrdahu!</wers>
    </zwrotka>
</sonet>
```

DTD – Deklaracje

- Deklaracja DOCTYPE:
  - <!DOCTYPE nazwa zawartość>
  - <!DOCTYPE nazwa SYSTEM „plik”>
    - plik – zawiera deklaracje DTD
      - Dysk:/katalogi/plik
      - Plik
      - Adres URL
  - <!DOCTYPE nazwa PUBLIC ”opis” ”adres”>
    - Uznany przez określoną społeczność standard dokumentów XML
    - „opis” to dodatkowe informacje o osobach odpowiedzialnych za standard

DTD – Deklaracje

- Deklaracja elementów:
  - <!ELEMENT nazwa zawartość>
- Deklaracja atrybutów:
  - <!ATTLIST nazwa.elem nazwa.atryb zawartość>
- Deklaracja encji:
  - <!ENTITY rozdzial1 SYSTEM "rozdzial1.xml">

DTD – Elementy

- Typy elementów:
  - Element tekstowy:
    - <!ELEMENT autor(#PCDATA)>
  - Element pusty:
    - <!ELEMENT linia EMPTY>
  - Element dowolny:
    - <!ELEMENT spacja ANY>

DTD – Elementy

- Listy elementów:
  - <<!ELEMENT sonet (autor, linia, tytuł, zwrotka)>

```
<sonet>
    <autor> Adam Mickiewicz </autor>
    <linia/>
    <tytuł> Sonet XIII </tytuł>
    <zwrotka> a tu tekst </zwrotka>
</sonet>
```

  - !ELEMENT auto (fabryka, nazwa, rok.produkcji)>

```
<auto>
    <fabryka> Ford </ fabryka >
    <nazwa> Sierra </nazwa>
    <rok.produkcji>1991</rok.produkcji>
</auto>
```

DTD – podelementy

- Wybór elementów:
  - <!ELEMENT poezja (wiersz | esej | powiesc)>

```
<poezja><wiersz> jakis wierszyk </wiersz></poezja>
<poezja><wiersz> jakis inny wierszyk </wiersz></poezja>
<poezja>
    <esej> a teraz sonet </esej>
<poezja>
```

  - <!ELEMENT adres (ip | nazwa | skrót) >

```
<adres>
    <ip>12.12.12.12</ip>
</adres>
<adres>
    <skrót>termit</skrót>
</adres>
```

DTD – podelementy

- Wybór elementów:
  - <!ELEMENT poezja ((wiersz | sonet), autor, tytuł)>

```
<poezja>
    <wiersz> jakis wierszyk </wiersz>
    <autor> Adaś Mickiewicz</autor>
    <tytuł> kotek</tytuł>
</poezja>
```

- Lista z wyborem elementów:
  - <!ELEMENT adresFTP ((ip | nazwa), user, (pass | anonym))>

```
<adresFTP>
    <ip>12.12.12.12</ip>
    <user>beny</user>
    <anonym/>
</AdresFTP>
```

DTD – indeks wystąpień

- Indeks wystąpień elementów:
  - + - element występuje co najmniej raz
    - <!ELEMENT multilotek (liczba+)>

<multilotek>

<liczba>5</liczba><liczba>43</liczba><liczba>55</liczba>

</multilotek

<multilotek>

<liczba>5</liczba>

</multilotek

DTD – indeks wystąpień

- Indeks wystąpień elementów:
  - ? - element występuje raz lub wcale;
    - <!ELEMENT multilotek (kupon.wygrany?)>

```
<multilotek>
   <kupon.wygrany>tak</kupon.wygrany>
</multilotek
<multilotek>
</multilotek>
```

DTD – indeks wystąpień

- Indeks wystąpień elementów:
  - * - element występuje dowolną ilość razy od 0 zaczynając;
    - <!ELEMENT mulilotek (główna.wygrana*)>

```
<multilotek>
  <główna.wygrana>jacek placek</główna.wygrana>
  <główna.wygrana>zenek i romek</główna.wygrana>
</multilotek
<multilotek>
</multilotek
```

DTD – indeks wystąpień

- Indeks wystąpień elementów:
  - <!ELEMENT sonet (autor+, linia?, tytuł, zwrotka*)>

```
<sonet>
    <autor>Adam Mickey </autor>
    <autor>Julek Czech </autor>
    <tytul>Sonet XIII</tytul>
    <zwrotka>
       <wers>Drzac muslimin całuje stopy twej opoki,</wers>
    </zwrotka>
    <zwrotka>
       <wers>Maszcie krymskiego statku, wielki Czatyrdahu!</wers>
    </zwrotka>
</sonet>
```

DTD – Deklaracja atrybutów

- Deklaracja atrybutów:
  - <!ELEMENT nazwa.elementu EMPTY>
  - <!ATTLIST nazwa.elementu definicje.atrybutów>
- Rodzaje atrybutów:
  - Tekstowy – ciąg znaków
    - <!ATTLIST team nazwa CDATA>
  - Wyliczane – pobierane z zadeklarowanej listy;
    - <!ATTLIST farba kolor (biały | zielony | żółty) „biały”>
  - Atomizowane – wartości rozpoznawalne przez XML;
    - <!ATTLIST team id ID>

DTD – ATRYBUTY

- Atrybuty atomizowane:
  - ID – identyfikator elementu w którym dany element wystąpił, nie może być dwóch takich samych wartości w całym dokumencie;
    - <!ATTLIST team id ID>
      - <team id=„a01”>gwardia</team>
      - <team id=„a02”>lech</team>
      - <team id=„a03”>kotwica</team>
  - IDREF – wskaźnik na wartość ID;
    - <!ATTLIST kapitan drużyna IDREF>
      - <kapitan drużyna=„a02”>roninio<./kapitan>
      - <kapitan drużyna=„a01”>dodi</kapitan>

DTD – ATRYBUTY

- IDREFS – wskaźnik na większą liczbę wartości IDREF;
  - <!ATTLIST drużyna pilkarz IDREFS>
    - <team drużyna=„a22,a24,a44,a11”>śląsk</team>
    - <team drużyna=„a2,a4,a43,a1”>ślęza</team>
- ENTITY – Wskaźnik encji zadeklarowanej w DTD
  - Wskaźnik na plik, obraz, tekst;
  - Zmienna kryjąca dalszą informację

DTD – ATRYBUTY

- Domyślne wartości atrybutów:
  - #REQUIRED – atrybut musi być ustawiony i powinien wystąpić;
    - <!ATTLIST team id ID #REQUIRED>
  - #IMPLIED – XML ma poinformować aplikacje, że nie podano żadnej wartości, a aplikacja ma zdecydować, co dalej;
    - <!ATTLIST pilkarz numer #IMPLIED>
  - #FIXED – każda wartość ma być zgodna z wartością domyślną
    - <!ATTLIST team trener #FIXED „Boniek”>

```
<mecz>
  <druzyna>
    <gracz id="g1" pseudo="Kowal"/>
    <gracz id="g2" pseudo="Miszczu"/>
    <gracz id="g3" pseudo="Balcer"/>
  </druzyna>
  <pierwsza5 kto="g1 g3"/>
  <kwarta>
    <element czas="1:12" kto="g1" typ="zbiorka"/>
    <element czas="2:23" kto="g2"/>
    <element czas="4:02" kto="g3" typ="asysta"/>
  </kwarta>
</mecz>
```

```
<!ELEMENT mecz (druzyna, pierwsza5, kwarta*)>
<!ELEMENT druzyna (gracz*)>
<!ELEMENT gracz EMPTY>
<!ATTLIST gracz id ID #REQUIRED>
<!ATTLIST gracz pseudo CDATA #REQUIRED>
<!ELEMENT pierwsza5 EMPTY>
<!ATTLIST pierwsza5 kto IDREFS #REQUIRED>
<!ELEMENT kwarta (element*)>
<!ELEMENT element EMPTY>
<!ATTLIST element czas CDATA #IMPLIED>
<!ATTLIST element kto IDREF #REQUIRED>
<!ATTLIST element typ (pkt | zbiorka | asysta) "pkt">
```

Encje

- Umożliwiają nazywanie fragmentów języka XML, następnie pozwalają wielokrotnie użyć poprzez ich nazwy
- Wewnętrzne (nazwa + ciąg znaków):
  - Deklaracja:
    - **<!ENTITY** nazwa „dowolny tekst”>
    - **<!ENTITY** pytanie „wciśnij T lub N”>
  - Wywołanie encji:
    - &nazwa;
    - Np.:**<tekst>**Grasz dalej? &pytanie; **</tekst>**
  - deklaruje się w DTD;
  - Widziane tylko w ramach tego samego dokumentu

Encje

- Zewnętrzne (nazwa + wskazanie na plik zewnętrzny):
  - Wskazują na fragmenty kodu XML lub elementy DTD

```
<!DOCTYPE ksiazka [
        <!ENTITY rozdzial1 SYSTEM "rozdzial1.xml">
        <!ENTITY rozdzial2 SYSTEM "rozdzial2.xml">
        <!ENTITY rozdzial3 SYSTEM "rozdzial3.xml"> ]>
<ksiazka>
                &rozdzial1;
                &rozdzial2;
                &rozdzial3;
</ksiazka>
```

Zewnętrzny i wewnętrzny DTD

- Zewnętrzny podzbiór DTD:
  - deklaracje wspólne dla wszystkich dokumentów danego typu:
    - elementy,
    - atrybuty,
    - encje.
- Wewnętrzny podzbiór DTD:
  - deklaracje lokalne dla dokumentu:
    - deklaracje encji,

Ograniczenia DTD

- Ograniczenie treści elementów do typu tekstowego (#PCDATA)
  - Brak innych typów danych (int, float, …)
- Brak wartości domyślnych elementów
- Małe możliwości kontroli typów atrybutów
  - Tylko typ wyliczeniowy
- Brak możliwości definiowania podstruktur wykorzystywanych wielokrotnie
- Brak dziedziczenia, rozszerzania i ograniczania
- Brak możliwości zmiany kolejności elementów
- Brak możliwości ustalenia kolejności atrybutów
- Ograniczenia ID dla atrybutów (zawsze dotyczy całego dokumentu bez rozróżnienia różnych elementów)
- Syntaktyka DTD nie jest podobna do zapisu XMLowego
- Rozszerzonie Struktury - XMLSchema