

# METAJEZYKI.ie.tu.koszalin.pl

dr inż. Robert Berezowski

e-mail:

**robert.berezowski@ie.tu.koszalin.pl**

ul. Śniadeckich 2

Pokój 223A

Konsultacje:

wtorek 11.00 – 13.00

piątek 16.00 – 17.00

Politechnika Koszalińska

Wydział Elektroniki i Informatyki

Katedra Inżynierii Komputerowej

# Zaliczenie wykładu

- Aplikacja składająca się z dwóch części:
  - moduł administratora do wprowadzania, edycji danych
    - Zbudowany w dowolnym języku programowania (C#, VB, C++, PHP, Delfi, Java...) korzystająca z wbudowanych bibliotek XML'owych
  - Moduł użytkownika do przeglądania danych
    - Zbudowany w oparciu o arkusze stylów XSL (ewentualnie XSL w połączeniu z innymi językami programowania)
- Dane przechowywane w plikach XML o ściśle określonej strukturze (plik xmlSchema).
- Projekt zrealizowany w grupach dwuosobowych.
- Ostateczny termin oddania: 4 czerwiec 2006
  - Etapy pośrednie:
    - 14 marca – określenie tematu pracy
    - 21 marca – prezentacja struktury danych - dtd
    - 4 kwietnia – prezentacja struktury danych – xmlSchema
    - 9 maja – Wstępna prezentacja projektu

# Program wykładów

- Wprowadzenie – ogólne informacje.
- Podstawy XML:
  - Budowa dokumentów;
  - Programy do tworzenia dokumentów XML;
- Walidacja dokumentów XML – gramatyka:
  - DTD – document type definition;
  - XML Schema;
- Style dokumentów XML (widok dokumentów):
  - CSS;
  - XSL;
- Analiza dokumentów XML (parsery):
  - SAX (Simple Api for XML) – analizator zdarzeniowy;
  - DOM (Document Object Model) – struktura drzewiasta;
- Transformacje dokumentów XML:
  - XML – HTML
  - XML – PDF;
  - XML – SVG;
- Tworzenie i przekształcanie dokumentów XML z pomocą różnych języków programowania (C#, Java)

# Znakowanie tekstu - markup

Proces znakowania dokumentów z użyciem czcionki, rozmiaru, odległości, akapitami itd

Hamlet:  
Spać albo nie spać.  
Oto jest pytanie.

← pogrubić  
← odstęp  
↑ wcięcie

**Hamlet:**  
Spać albo nie Spać.  
Oto jest pytanie.

Frame (MIF) `<Font <FTag `B'>><String `Hamlet'>`

QuarkXPress `<B>Hamlet<B>`

RTF `{\b\f5\cf1 Hamlet}`

TeX/LaTeX `\textbf{Hamlet}`

HTML `<B>Hamlet</B>`

# Programy (edytory tekstu) i ich formaty – do dziś

- Brak przenośności oraz możliwości wspólnej pracy
- Prawie każda aplikacja wprowadza swój wewnętrzny format.
- Nowe wersje tej samej aplikacji wprowadzają zmiany do używanego formatu:
  - wsteczna kompatybilność,
  - brak możliwości zapisu do formatu poprzednich wersji.
- Aplikacje dostarczają konwerterów:
  - tylko do najpopularniejszych formatów,
  - możliwość utraty danych podczas konwersji.

# Standardy

- Ciagle nie istnieją uznane standardy (?OfficeOpenXML?).
- Istnieją substandardy w różnych dziedzinach:
  - dokumenty biurowe: Microsoft Word,
  - teksty naukowe: Postscript, TeX,
  - Internet: HTML, GIF, JPG,
  - elektroniczna wymiana danych: EDIFACT.
- Standard musi być:
  - własnością publiczną,
  - otwarty i jawny,
  - niezależny od konkretnego producenta oprogramowania.
- Masa informacji cyfrowej powoduje potrzebę struktury:
  - jeden format dokumentu nie wystarczy dla 5 miliardów ludzi,
  - ale nie możemy operować milionami niekompatybilnych formatów.

## Rozwój języków uogólnionego znakowania tekstu

- 1969, IBM: GML – Generalized Markup Language (Goldfarb, Mosher, Laurie).
- 1986: SGML – Standard Generalized Markup Language, ISO 8879:1986.
- 1991: powstaje World Wide Web.
- 1994: HTML 2.0 zdefiniowany jako zastosowanie SGML-a.
- 1996: XML – Extensible Markup Language, World Wide Web Consortium.

# SGML

- Oparty na GML – Generalized Markup Language
- SGML – Standard Generalized Markup Language,
- Jest metajęzykiem służącym do opisywania innych języków znacznikowych
- Deklaracja SGML określa rodzaj tworzonego dokumentu: raport, broszura, dokumentacja, itd.;
- Deklaracja SGML określa znaki , które będą interpretowane jako dane lub jako znaczniki np.:  
<x>, {x};
- Rdzeniem aplikacji SGML jest DTD (definicja typu dokumentu), gdzie określone są poszczególne elementy i ich atrybuty;



# SGML

- SGML jest dobrym rozwiązaniem dla przemysłu – kosztowny, skomplikowany ale jednocześnie wyjątkowo potężny;
- Wykorzystanie SGML wymaga zbyt wiele czasu, narzędzi i szkoleń;
- Za pomocą SGML'a opisano m.in.:
  - HTML – Hyper Text Markup Language,
  - HTTP – Hyper Text Transfer Protocol,
  - CSS – Cascading StyleSheets,
  - XML – Extensible Markup Language,
  - HDML – język do urządzeń narecznych,
  - FAQ – język często zadawanych pytań,
  - KEDU – używany przez ZUS;

# HTML

- HTML służy do wyświetlania tekstu w przeglądarkach internetowych;
- Brak kontroli składniowej – przeglądarki działają tak by zinterpretować wszystko co choć z daleka przypomina kod HTML;
- HTML nie ma struktury;
- W HTML nie można określić informacji jaką zawiera dokument;
- HTML nie jest międzynarodowy;
- HTML nie ma możliwości kontroli połączeń (linków);
- Kod HTML trudno wykorzystać w innych tworzonych dokumentach;
- HTML ma ograniczoną ilość znaczników;
- Wystarczający na dziś, ale zbyt ograniczony na przyszłość;

# Podstawy XML

- XML – (eXtensible Markup Language – rozszerzalny język znaczników – 1996

twórcy: Tim Bray (Netscape), Jean Paoli (Microsoft), C.M. Sperberg-McQueen (University of Illinois).

- Jest językiem znacznikowym (jak HTML), ale może posiadać nieskończoną ilość znaczników
- Zestaw reguł do tworzenia znaczników dzielących identyfikujących dokument;
- Pozwala definiować strukturę znaczników dla dowolnych dziedzin zastosowań.

# Podstawy XML

- Tworzone znaczniki mogą być ściśle określone w DTD;
- Przeglądarka rozpozna znaczniki na podstawie DTD lub bezpośrednio z dokumentu;
- Znaczniki XML opisują strukturę i znaczenie dokumentu;
- Informacje o sposobie wyświetlania zawarte są w osobnym arkuszu stylów;

# HTML a XML

- Znaczenie elementów i ich atrybutów z góry określone.
- Interpretację elementów określa standard, a w praktyce przeglądarki internetowe.
- To, co jest poprawne również określają przeglądarki internetowe.
- Znaczenie elementów i ich atrybutów określa użytkownik lub aplikacja.
- `<p>` może w jednym dokumencie oznaczać **paragraf**, w drugim **pomoc**, a w trzecim **pismo odręczne**.
- Poprawność XML-a jest ściśle określona przez specyfikację.

# SGML a XML

- Filozofia: jeden duży system edytoryalny.
- Konieczność definiowania struktury.
- Skomplikowana składnia, wiele opcji.
- Trudność tworzenia parserów.
- Bardzo drogie narzędzia.
- Filozofia: wiele małych komunikujących się ze sobą modułów.
- Opcjonalne definiowanie struktury.
- Uproszczona składnia.
- Łatwość tworzenia parserów.
- Darmowe narzędzia.

# Znakowanie tekstu

- Stworzenie najodpowiedniejszego modelu dla naszych własnych dokumentów (różny dla różnych zastosowań).
- Języki znacznikowe: „rodzaj treści” + treść
- Przykłady:
  - encyklopedia: **<nazwisko>**, **<imie>**, **<ur>**, **<zm>**, **<wymowa>**, **<liczba-mieszk>**
  - Prawo (kodeks): **<rocznik>**, **<poz>**, **<art>**, **<sąd>**, **<sygn-wyroku>**, **<teza>**
  - dokument techniczny: **<part-number>**, **<function-name>**
  - patenty: **<wynalazca>**, **<nr-zgłoszenia>**
  - ubezpieczenia: **<data-polisy>**, **<wart-polisy>**
  - koszykówka: **<drużyna>**, **<gracze>**, **<punkty>**, **<faule>**

# Przykład dokumentu HTML

```
<html>
<body>
  <h1>Autor: Adam Mickiewicz</h1>
  <h2>Tytuł: Sonet XIII</h2>
  <p>Drząc muślinin całuje stopy twej opoki,</p>
  <p>Maszcie krymskiego statku, wielki Czatyrdahu!</p>
  <p>O minarecie świata! O gór padyszachu!</p>
  <p>Ty, nad skały poziomu uciekłszy w obłoki.</p>
  <p>Siedzisz sobie pod bramą niebios, jak wysoki </p>
  <p>Gabryjel pilnujący edeńskiego gmachu; </p>
</body>
</html>
```



# Przykład dokumentu XML

<sonet>

<autor>Adam Mickiewicz </autor>

<tytuł>Sonet XIII</tytuł>

<wers>Drząc muślinin 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 uciekłeś w obłoki.</wers>

<wers> Siedzisz sobie pod bramą niebios, jak wysoki </wers>

<wers> Gabryjel pilnujący edeńskiego gmachu; </wers>

</sonet>

# Zastosowania XML'a

## Zarządzanie dokumentami, treścią, wiedzą (dokument tekstowy):

- Pierwotne zastosowanie SGML-a.
- Dokumenty tworzone przez człowieka i przeznaczone dla człowieka.
- Długi czas życia dokumentów
- Typowy model mieszany zawartości.

### Dokument tekstowy:

```
<wiadomość>Dnia <data>21.02.2004</data>  
(<dzień> sobota </dzień>) zajęcia z  
<przedmiot> Metajęzyków </przedmiot> nie odbędą  
się z powodu braku studentów  
</wiadomość>
```

# Zastosowania XML'a

## Elektroniczna wymiana danych, integracja aplikacji: (baza danych)

- Nowa klasa zastosowań XML-a.
- Konieczność dokładnego kontrolowania struktury i zawartości.
- Dokumenty tworzone oraz przetwarzane automatycznie
- Dokumenty tworzone tylko na czas komunikacji.

### Baza danych:

```
<mecz>  
  <team1>Nabla</team1>  
  <team2>Format</team2>  
  <pkt1>112</pkt1>  
  <pkt1>46</pkt1>  
</mecz>
```

# Podstawy XML

- Znaczniki XML ułatwiają automatycznym programom przetwarzającym odnalezienie w dokumencie odpowiednich definicji;
- Nazwy elementów XML należy dobrać tak by nadać im jak najprecyzyjniejsze znaczenie;
- XML czyni znacznie prostszymi zadania projektantów stron internetowych. Można tworzyć dowolne znaczniki i nie ograniczać się do istniejących;
- XML jest prostym w zapisie językiem i nie potrzebne jest żadne dodatkowe oprogramowanie żeby dowiedzieć się o zawartości dokumentu;

# Podstawy XML

- XML dobrze nadaje się do pracy z bazami danych;
- XML jest formatem samoopisującym się i jest bardzo dobrze udokumentowany;
- XML nie jest niczyją własnością, nie jest objęty prawami autorskimi, patentowymi, tajemnicą handlową;
- XML jest idealny dla dużych i złożonych dokumentów, gdyż dane w nim są ustrukturyzowane;
- XML zawiera mechanizm pozwalający integrować ze sobą dane z różnych źródeł i wyświetlać je jako pojedynczy dokument;
- Dane można przekształcać nawet w biegu;
- Poszczególne części można pokazywać lub ukrywać w zależności od akcji podejmowanych przez użytkownika.

# Tworzenie dokumentów XML

- Dowolny edytor tekstowy – notatnik, vi;
- Edytor typu WYSIWYG (co widzisz to otrzymujesz) – AdobeFrameMaker;
- Edytor strukturalny wyświetlający XML w postaci drzewa – JUMBO, XMLViewer, InfoxFree, XMLShell;
- Zaawansowane oprogramowanie: Xselerator, TurboXml, Oxygen, XmlSpy;

# XML – od stworzenia do wyświetlenia

- Dokument XML jest tworzony w edytorze;
- Analizator składniowy XML (procesor lub parser XML) czyta dokumenty XML i sprawdza, czy jest on poprawnie sformułowany;
- Jeżeli wyniki testów będą pomyślne, procesor przekształca dokument na drzewko elementów;
- Parser przekazuje całe drzewko lub poszczególne jego węzły do aplikacji:
  - przeglądarka Mozilla, Amaya lub Internet Explorer – dane zostaną wyświetlone;
  - baza danych lub aplikacja - dane mogą też być dalej przetwarzane;

# Podstawy XML

- XML można łączyć z wieloma technologiami:
  - obsługa HTML w celu zapewnienia wstecznej kompatybilności z istniejącymi przeglądarkami;
  - arkusze stylów CSS i XSL;
  - adresy URL i URI;
  - język opisu łączy XLinks, XPointers;
  - zestaw znaków Unicode;



## Technologie związane – Arkusze stylów

- Określa sposób wyświetlenia dokumentu XML;
- Rodzaje arkuszy stylów:
  - CSS – cascading Style Sheet;
  - XSL – Extensible Style Language;

# Arkusze stylów - CSS

- Za pomocą CSS można określić:
  - wielkość, krój i grubość czcionki;
  - wcięcia, wyrównywanie akapitów;
  - różnym elementom można przypisywać różne style;
- CSS może zmienić jedynie sposób wyświetlania poszczególnych elementów, i tylko elementów
- CSS jest dość dobrze obsługiwany przez przeglądarki.

# Arkusze stylów - XSL

- Dokumenty XSL same w sobie są poprawnymi dokumentami XML;
- Dokumenty XSL zawierają ciągi reguł dotyczących pewnych układów elementów XML;
- Procesor XSL porównuje dokument XML ze wzorcami z arkusza stylów i po rozpoznaniu stosowany jest wybrany opis formatowania;
- Tekst wynikowy może być swobodniej kształtowany i uzupełniany, nie musi się ograniczać tylko do tekstu wejściowego wzbogaconego o formatowanie;

# Arkusze stylów - XSL

- Arkusze stylów XSL mogą zmienić układ i kolejność elementów;
- Niektóre elementy mogą ukryć, inne wyświetlić;
- Style można wyświetlić na podstawie:
  - elementów;
  - atrybutów znacznika;
  - położenia znacznika w dokumencie;
- Za pomocą arkusza XSL dokument XML można łatwo przekształcić na dokument HTML z arkuszami stylów CSS;

# Adresowanie w XML

- Dokumenty XML, HTML wskazywane są przez adresy URL;
  - Adresy te są dobrze rozumiane i prawidłowo obsługiwane przez przeglądarki;
- Specyfikacja XML dodatkowo używa ogólniejszych adresów URI;
  - Nie koncentrują się na lokalizacji lecz na zasobie;
  - Teoretycznie URI jest w stanie znaleźć najbliższą kopię dokumentu lub odnaleźć przeniesiony dokument;
  - Ten typ adresów jest na etapie badań;
  - W tej chwili jedynym rodzajem obsługiwanych przez oprogramowanie są adresy URL.

# Łącza w XML

- W dokumentach XML można użyć standartowych łączy dokumentów HTML - (`<a href="..."></a>`);
- XML pozwala użyć XLinks do łączenia dokumentów
- XML pozwala użyć XPointers do wskazywania poszczególnych części dokumentów.

# Łącza XLinks

- Standard XLinks umożliwia uczynienie łączem dowolnego dokumentu, a nie tylko elementu <a>
- Łącza te mogą być:
  - dwukierunkowe;
  - wielokierunkowe;
- Łącza mogą wskazywać wiele kopii dokumentu docelowego, z których wybrana będzie kopia najbliższa;
- Obecnie standard XLinks używa do określenia przyłączanej strony zwykłego adresu URL;

# Łączy XPointers

- Standard XPointers umożliwia łączom wskazywanie nie tylko konkretnego dokumentu w konkretnym miejscu, ale nawet jego dowolnego fragmentu;
- Wskaźnik XPointer odnosić się może do:
  - konkretnego dokumentu,
  - jego kolejnego wystąpienia;
  - pierwszego elementu zawartego w danym elemencie, itd.;



# Łącza XPointers

- Wskaźniki XPointers dają możliwość wyjątkowo elastycznego łączenia dokumentów, które przy tym nie wymaga dodawania specjalnych znaczników w celu zrealizowania takiego złączenia;
- W przeciwieństwie do stosowanych w HTML znaczników <a>, wskaźniki XPointers nie muszą wskazywać punktu w dokumencie, mogą wskazywać zakres lub obszar;
- XPointers można użyć do wybierania części dokumentu, tak aby tę część skopiować lub załadować do programu;

# UNICODE

- Większość stron internetowych napisano w języku angielskim (HTML).
- XML w pełni obsługuje dwubajtowy zestaw znaków Unicode, a także jego bardziej zwarte postacie.
- Unicode pozwala obsłużyć niemalże wszystkie współcześnie używane języki.

# UNICODE

- Aby projektować i odczytać napis w jakimś egzotycznym (polskim) języku należy posiadać trzy elementy:
  - zestaw znaków tego języka;
  - czcionkę odpowiednią do tego zestawu znaków;
  - system operacyjny i oprogramowanie, które będą potrafiły z tego zestawu znaków skorzystać.
    - InfoxFree potrafi obsługiwać UNICODE;

# Podsumowanie

- XML określa gramatykę znaczników, których można użyć do oznakowania dokumentu;
- Dokument XML używa znaczników XML, a domyślnym zestawem znaków jest Unicode;
- Dokument XML może między innym zawierać łącza hipertekstowe łączące go z innymi dokumentami i zasobami;
- Łącza tworzone są zgodnie ze specyfikacją XLinks;
- Łącza identyfikują dołączany dokument przez adres URI (teoretycznie) lub URL (praktycznie);

# Podsumowanie

- W łańcuchu XLinks może wystąpić dokładniejsza specyfikacja miejsca docelowego za pomocą adresacji XPointers;
- Jeżeli dokument XML ma być czytany (oglądany) przez człowieka, użycie arkusza stylów pozwala określić sposób formatowania poszczególnych elementów;
- Arkusz stylów może być napisany w różnych językach stylów, z których najpopularniejsze są CSS i XSL;

# Problemy z XML

- XML nie jest jeszcze całkiem ukończony. W obsługującym go oprogramowaniu jest jeszcze szereg niedoróbek.
- Mozilla, Internet Explorer, Amaya, Netscape tylko w pewnym zakresie obsługują XML;
- Technologie związane z XML tylko w pewnym zakresie są obecnie obsługiwane przez przeglądarki;

# Jeszcze tylko kilka adresów

- SGML Home Page:
  - [www.sgmlsource.com](http://www.sgmlsource.com)
- Rekomendacja XML:
  - [www.w3.org/TR/REC-xml](http://www.w3.org/TR/REC-xml)
- W3C – The World Wide Web Consortium:
  - [www.w3.org](http://www.w3.org)
- XML
  - [www.xml.com](http://www.xml.com)
  - [www.xml.org](http://www.xml.org)
  - [www.oasis-open.org/cover/xml.html](http://www.oasis-open.org/cover/xml.html)
- XML/SGML w Polsce
  - [www.xml.pl](http://www.xml.pl)
  - [pl.comp.xml](http://pl.comp.xml)

# I kilka książek

- Harold, E. R., „XML Księga eksperta”,
- Marchal, B., „XML w przykładach”,
- Young, M. J., „XML krok po kroku”,
- Kaziienko, P.; Gwiazda, K. „XML na poważnie” Wydawnictwo Helion, 2002
- van Otegem, M. „XSLT dla każdego” Wydawnictwo Helion, 2003