

Programowanie komunikacji człowiek-komputer

dr inż. Joanna Ochelska-Mierzejewska

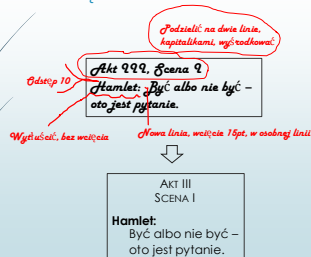
2

Znakowanie tekstu (ang. text markup) – koncepcja

- the process of marking manuscript copy for typesetting with directions for use of type fonts and sizes, spacing, indentation, etc.
(The Chicago Manual Of Style)
- Odrębne pismo a znakowanie tekstu
 - Ustalenie odstępów, przecinków, dużych liter, podkreśleń itp.
- Znakowanie tekstu a struktura dokumentu
 - Ustalenie wyglądu dokumentu, tabel, list, numeracji itp.
- Znakowanie tekstu w epoce komputerów
 - Często czysty tekst (email, forum, SMS itp.)

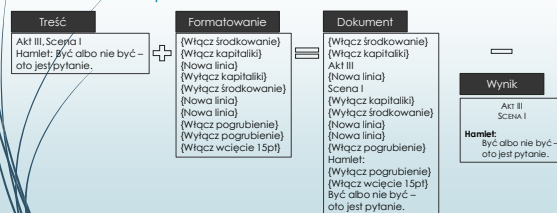
3

Początki znakowania tekstu



4

Znakowanie tekstu w epoce komputerów



5

Znakowanie tekstu

- Znakowanie tekstu dla prezentacji
- Opis wyglądu dla fragmentu tekstu (czcionka, kolor, ...)
- Przykłady:
 - Postscript, PDF, TeX
 - HTML:

 - Formatowanie w edytorach tekstu
 - XSL-FO
- Semantyczne znakowanie tekstu
- Opisanie roli/znaczenia fragmentu tekstu
- Przykłady:
 - LaTeX (człecłowa)
 - HTML: <Q> <CITE> <VAR>
 - Style w edytorach tekstu
 - Większość aplikacji opartych na SGML i XML

6

Potrzeba standardu

- Większość aplikacji wprowadza własny wewnętrzny format
- Nowe wersje aplikacji często wraz z rozbudową funkcjonalności wprowadzają zmiany do używanego formatu – występują problemy ze zgodnością wstecz (często brak możliwości zapisu do formatu poprzednich wersji)
- Aplikacje dostarczają konwerterów, ale
 - tylko do najpopularniejszych formatów
 - możliwość utraty danych podczas konwersji

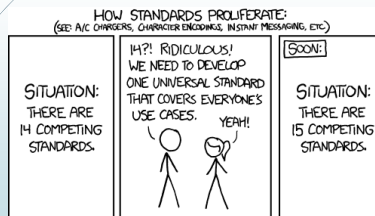
7

Potrzeba standardu

- Chcielibyśmy mieć format
 - Na tyle elastyczny, by mógł reprezentować wiele formatów dokumentów
 - Na tyle jednorodny i prosty, by można było łatwo budować przetwarzającego go dokumenty
- Standard musi być
 - Własnością publiczną
 - Otwarty i jawny
 - Niezależny od producenta oprogramowania

8

Kolejny standard



xkcd.com

9

Droga do XML

- 1967 – William Tunnicliffe, **GenCode**
- 1969 – IBM – SCRIPT projekt, eksperyment INTIME
 - Charles Goldfarb, Edward Mosher, Raymond Lorie
 - Generalized Markup Language (**GML**)
- 1974–1986 – Standard Generalized Markup Language (**SGML**)
 - ISO 8879:1986
- 1990s – Extensible Markup Language (**XML**)
 - W3C Recommendation 1998
 - Simplification(!) and subset of SGML

10

Przykład składni GML

```
:h1.Chapter 1: Introduction
:p.GML supported hierarchical containers, such as
:ol
:li.Ordered lists (like this one),
:li.Unordered lists, and
:li.Definition lists
:eol. as well as simple structures.
:p.Markup minimization (later generalized and formalized in
SGML), allowed the end-tags to be omitted for the "h1" and
"p" elements.
```

11

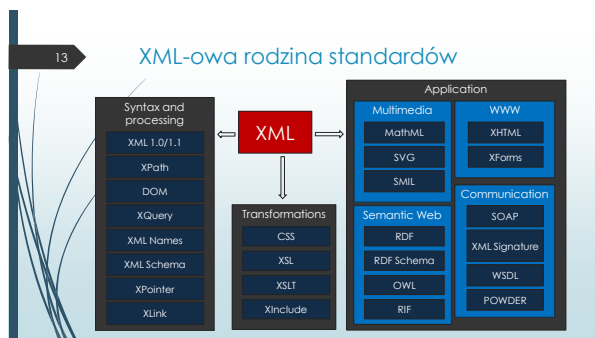
SGML – Standard Generalized Markup Language

- Trzy najważniejsze cechy SGML-a
 - Deskryptywność opisu
 - Typizacja dokumentu
 - Maksymalna przenośność
- Wady SGML-a
 - Konieczność definiowania struktury
 - Duża złożoność, zbyt wiele opcji
 - Trudność tworzenia narzędzi (drogą!)

12

Czym jest XML?

- **Standard** – Extensible Markup Language World Wide Web Consortium (W3C) Recommendation
 - version 1.0 – 1998
 - version 1.1 – 2004
- **Język** – format pisania struktury dokumentów dla plików tekstowych
- **Metajęzyk** – rozszerzalna i rozwijana lista rodzin konkretnych języków (XHTML, SVG, itp.)
 - Wspólny
 - model modelu dokumentu
 - syntaktyka
 - techniczne wsparcie (parsery, biblioteki, narzędzia, standard)



14 Czym XML nie jest?

- Językiem programowania
- Rozszerzeniem HTML
- Formą prezentacji
 - Mówimy "dane prezentowane w formacie XML", a nie "XML prezentuje dane"
- Web-only, WebServices-only, database-only, *-only technology
- Zła zasada
 - XML nie jest dobry na wszystko!!!

15 Struktura dokumentu XML

- Element
 - Dokument zawiera dokładnie jeden element główny
 - Elementy mogą zawierać podelementy i węzły tekstowe

```
<student>Jan Kowalski</student>
<koniec_rejestracji_studentów />
```

16 Struktura dokumentu XML

- Nazwa elementu
 - Wrażliwa na wielkość liter
 - Może zawierać
 - Znaki alfanumeryczne i cyfry
 - Znaki hardowe (np. `<`, `>`, `&`)
 - Znaki interpunkcyjne (.-)
 - Może zaczynać się od litery lub podkreślenia

```
<student>Jan Kowalski</student>
<STUDENT>Jan Kowalski</STUDENT>
<student>Jan Kowalski</student>
<1student>Jan Kowalski</1student>
<_student>Jan Kowalski</_student>
<stud.ent>Jan Kowalski</stud.ent>
```

17 Struktura dokumentu XML

- Atrybuty
 - Wartości muszą być ujęte w cudzysłowy lub apostrofy
 - Dany element nie może posiadać dwóch atrybutów o tej samej nazwie
 - Wartość atrybutu nie może zawierać znaku `<`

```
<student semestr="5">Jan Kowalski</student>
<rejestracja naSemestr="zimowy" rokImmatrykulacji="08/09"/>
```

18 Struktura dokumentu XML

- Komentarz
 - Mogą być wieloliniowe
 - Mogą zawierać dowolne znaki `<`, `>`, `&`
 - Nie mogą zawierać w treści po sobie dwóch minusów
 - Nie mogą być wstawiane w etykiety znaczników
 - Mogą wystąpić na zewnątrz elementu głównego

```
<!-- To mój komentarz i nikt tego nie przetworzy -->
<student>Jan Kowalski</student>
```

19

Struktura dokumentu XML

- Deklaracja XML-owa
 - Zawsze na początku dokumentu (w pierwszej linii, od pierwszego znaku)
 - Deklarację można omijać, jeśli dokument jest w wersji 1.0, a jego kodowanie to UTF-8 lub UTF-16
- Document type declaration (DTD)
 - Opcjonalnie

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE document SYSTEM "doc.dtd">
```

20

Struktura dokumentu XML

- Instrukcje sterujące/przetwarzania
 - Mogą wystąpić na zewnątrz elementu głównego
 - Nazwa instrukcji sterującej musi być różna od ciągu „XML” (i wariantów z różną wielkością liter)

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<?xml-stylesheet href="style.css" type="text/css" ?>

<student>Jan Kowalski</student>
```

21

Struktura dokumentu XML

- Sekcja CDATA
 - Jedynym zabronionym ciągiem znaków jest]]> (również w zawartości tekstowej – w sekcji CDATA można cytować dokument XML nie zawierający sekcji CDATA)
 - Sekcji CDATA nie można zagnieżdżać

```
<student>
  <![CDATA[
    <svg xmlns="http://www.w3.org/2000/svg"
      width="12cm" height="10cm">
      <ellipse rx="110" rx="130"/>
      <rect x="4cm" y="1cm" width="6cm"
        height="3cm" />
    </svg>
  ]]>
</student>
```

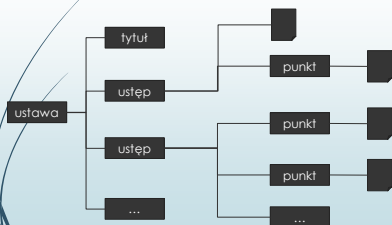
22

Jak wygląda XML?

```
<?xml version="1.0" encoding="UTF-8"?>
<ustawa>
  <tytuł> o realizacji przedpłat na samochody </tytuł>
  <ustęp nr="1">
    Osoby, które w 1981 r. wniosły przedpłaty na zakup
    samochodów i dotychczas samochodu nie odebrały, mają
    prawo do otrzymania rekompensaty w wysokości:
    <punkt nr="1">
      5.930 zł, jeśli przedpłata została wniesiona na
      samochód marki Fiat 126p,
    </punkt>
    <punkt nr="2">
      8.400 zł dla samochodu marki FSO 1500.
    </punkt>
  </ustęp>
</ustawa>
```

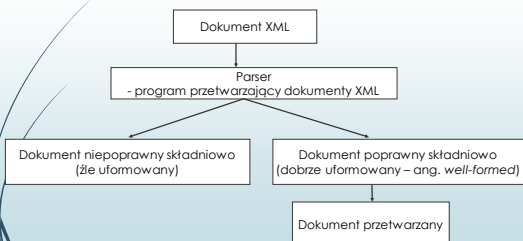
23

Dokument XML – Drzewo



24

Poprawność składniowa



25

Poprawność składniowa

```

<?xml version="1.0"
<?xml>
<student wydzial="FTiMS" nr_albumu="220220">
  <rejestracja kierunku="informatyka">pełna</rejestracja>
  <rejestracja kierunku="fizyka">pełna</rejestracja>
</student>
<student wydzial="FTiMS" nr_albumu="220110">
  <rejestracja kierunku="matematyka">pełna</rejestracja>
  <rejestracja kierunku="fizyka">warunkowa</rejestracja>
</student>
</pl>

```

This page contains the following errors:

error on line 11 at column 6: Opening and ending tag mismatch: pl line 0 and pl

Below is a rendering of the page up to the first error.

26

Struktura XML

Element główny (root)

Znacznik początkowy

Atrybut

Znacznik końcowy

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<?xml>
<pl>
  <student wydzial="FTiMS" nr_albumu="220220">
    <rejestracja kierunku="informatyka">pełna</rejestracja>
    <rejestracja kierunku="fizyka">pełna</rejestracja>
  </student>
  <student wydzial="FTiMS" nr_albumu="220110">
    <rejestracja kierunku="matematyka">pełna</rejestracja>
    <rejestracja kierunku="fizyka">warunkowa</rejestracja>
  </student>
</pl>

```

Element

Zawartość tekstowa

27

Rozdzielenie treści od wyglądu

- Zgodnie z ideą XML w dokumentach tylko
 - Treść/dane
 - Znaczniki struktury, znaczenia („znakowanie semantyczne”)
- Brak informacji o wyglądzie
- Jak prezentować?
 - Interpretacja znanych typów dokumentów
 - Zewnętrzne arkusze stylu (dowolne typy dokumentów)

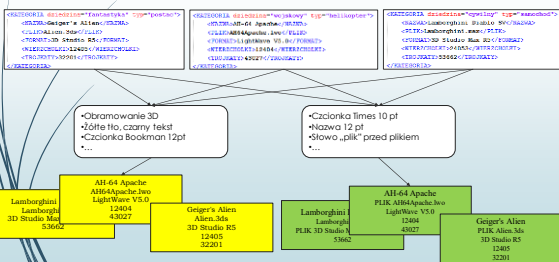
28

Zalety rozdzielenia treści od wyglądu

- Łatwiejsza analiza danych źródłowych i inne już znane zalety znakowania semantycznego
- Możliwość ponownej prezentacji
 - Porównanie danych
 - Działanie dokumentu o takiej samej strukturze
- Zmiana wyglądu dokonywana poza dokumentem źródłowym, raz dla całej klasy dokumentów
- Możliwość definiowania wielu arkuszy stylu dla każdej klasy dokumentów w zależności od
 - Porozumienia
 - Medium (tekst, wydruk, dźwięk)
 - Szczegółowości (pełen raport, podsumowanie, itp.)
 - Preferencji czytelnika (rozmiar czcionki, kolory, ...)

29

Idea arkusza stylów



30

Standardy związane z prezentacją XML

- Przypisanie arkusza stylu do dokumentu
 - Associating Style Sheet with XML documents
- Języki do zapisywania arkusza stylu
 - DSSSL (historyczny, stosowany dla SGML) Document Style Semantics and Specification Language
 - CSS Cascading Style Sheet
 - XSL Extensible Stylesheet Language

31

Przypisanie stylu do dokumentu

- Za pomocą instrukcji przetwarzania `<xml-stylesheet>`
- Opisane w rekomendacji W3C
Associating Style Sheet with XML documents

■ Jeden styl

```
<?xml-stylesheet type="text/css" href="style.css" ?>
```

■ Style alternatywne

```
<?xml-stylesheet title="niebieski" type="text/css" href="niebieski.css" ?>
```

```
<?xml-stylesheet title="złoty" alternate="yes" type="text/css" href="zolty.css" ?>
```

```
<?xml-stylesheet title="malutki" alternate="yes" type="text/css" href="maly.css" ?>
```

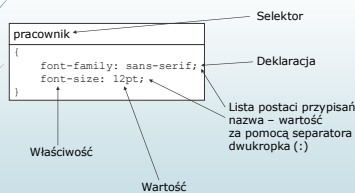
32

CSS

- Początki idei arkusza stylu: lata 70-te XX wieku
- Początki prac nad CSS sięgają roku 1994
 - Rekomendacja CSS Level1: grudzień 1996
 - Rekomendacja CSS Level2: maj 1998
 - Mniej więcej zaimplementowana we współczesnych przeglądarkach internetowych
 - CSS 2.1: czerwiec 2011
 - Bardziej restrykcyjny CSS 2 z większymi możliwościami
- CSS Level3: ciągle rozwijany
 - Modularyzacja (niektóre moduły już zatwierdzone)

33

Składnia



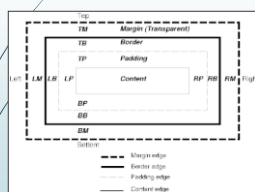
34

Tekst i czcionka

- `color, background-color, background-image` – kolor i tło
- `font-family` – nazwa czcionki oraz serif, sans-serif, monospace, ...
- `font-size` – rozmiar czcionki
- `font-style, font-weight` – styl czcionki
- `text-decoration` – dodatkowe style czcionki
- `text-align` – wyrównanie

35

Pudełka i wyrównanie



Źródło: <http://www.w3.org/TR/CSS2/box.html>

- Zagnieżdżenie bloków odpowiada zagnieżdżeniu elementów w dokumentcie
- Możliwe ręczne pozycjonowanie
- `margin, padding` – margines zewnętrzny i wewnętrzny
- `border-style, border-color, border-width` – obramowanie
- `position {static, relative, absolute, fixed}` – sposób pozycjonowania
- `left, right, top, bottom` – pozycja
- `width, height, min-width, max-width, ...` – rozmiar

36

Styl zależny od medium – przykład

```
@media print {
  pracownik {
    background-color: white;
    font-family: serif;
  }
}

@media screen {
  pracownik {
    background-color: yellow;
    font-family: sans-serif;
  }
}

@media all {
  pracownik {
    border-style: solid;
  }
}
```

37

Własność display

- Definiuje sposób wyświetlania/interpretowania elementu
- Możliwe wartości: `block`, `inline`, `list-item`, `table`, `table-cell`, ..., `none`
- Podstawowa własność w przypadku wizualizacji XML
- Rzadko stosowana dla HTML

38

Selektory

- | Selektory | Zależności selektorów |
|---|---|
| <ul style="list-style-type: none"> imię - nazwa elementu imię, nazwisko - oba elementy dział nazwisko - potomek dział > nazwa - dziecko A + B - następnik | <ul style="list-style-type: none"> dział nazwisko - potomek dział > nazwa - dziecko A + B - następnik A ~ B - wszystkie następni |

39

Pseudo klasy, pseudo elementy

- | Pseudo klasy | Pseudo elementy |
|--|--|
| <ul style="list-style-type: none"> <code>selector:pseudo-class</code> imię: <code>first-child</code> - pierwsze dziecko <code>p:last-child</code> - ostatnie dziecko <code>a:link</code>, <code>a:visited</code>, <code>a:hover</code>, <code>a:active</code> ... | <ul style="list-style-type: none"> <code>selector:pseudo-element</code> <code>pracownik::before</code> <code>pracownik::after</code> <code>p::first-letter</code> ... |

40

Selektory atrybutów

- `pracownik[stanowisko]` - test istnienia atrybutu
- `pracownik[stanowisko = 'kierownik']` - test wartości atrybutu
- `pracownik[funkcje ~ 'kierownik']` - wartość występuje na liście
- ...
- `pracownik#K12` - wartość atrybutu zadeklarowanego jako ID

41

Generowanie zawartości

- Wstawianie tekstów nie będących zawartością tekstową dokumentów
- Dostęp do wartości atrybutów
- Automatyczne numeracja

```
telefon[typ="biuro"]::before {
  content: 'tel. ';
}

pracownik::before {
  content: attr(stanowisko);
}

h2::before {
  counter-increment: section;
  content: counter(section) " ";
}
```

42

Przykładowy dokument XML

```
<?xml version="1.0" encoding="ISO-8859-2" standalone="no"?>
<?xml-stylesheet type="text/css" href="pracownik.css"?>
<firma>
  <nazwa>Księgowność</nazwa>
  <pracownik stanowisko="starszy referent" id="102103">
    <imie>Dawid</imie>
    <nazwisko>Duchowcy</nazwisko>
    <telefon typ="biuro">+48421234567</telefon>
    <telefon typ="kom">+48600777999</telefon>
    <email>dd@superfirma.pl</email>
  </pracownik>
  <pracownik stanowisko="kierownik" id="102104">
    <imie>Julian</imie>
    <nazwisko>Kowalski</nazwisko>
    <telefon typ="biuro">+48421355444</telefon>
    <telefon typ="kom">+48600222555</telefon>
    <email>j@superfirma.pl</email>
  </pracownik>
</firma>
```

43

Przykładowy dokument CSS

```

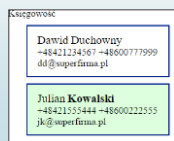
pracownik {
  display: block;
  margin: 10px auto 10px 30px;
  padding: 0.5em 1em;
  width: 200px;
  border-style: solid;
  border-width: 2px;
  border-color: #002288;
  background-color: #FFFFFF;
}

pracownik[stanowisko='kierownik'] {
  background-color: #C0FFD0;
}

imie, nazwisko {
  display: inline;
  font-size: larger;
}

pracownik[stanowisko='kierownik'] nazwisko {
  font-weight: bold;
}

```



44

Przykładowy dokument CSS – selektory

```

firma {
  display: block;
  background-color: #E0FFFF;
  color: rgb(0, 0, 33%);
  font-family: 'Bookman', serif;
  font-size: 14pt;
}

firma > nazwa {
  font-size: 1.5em;
  font-family: 'Verdana', 'Arial', sans-serif;
  font-weight: bold;
  text-align: center;
}

```



45

Możliwości i zalety CSS

- Ogromne możliwości wpływania na wygląd
- Rozróżnianie elementów ze względu na
 - Nazwę
 - Położenie w drzewie dokumentu
 - Występowanie atrybutów
 - Wartości atrybutów
- Szerokie wsparcie
 - Przeglądarki internetowe
 - Narzędzia do tworzenia arkuszy
- Łatwo pisać proste arkusze ©

46

Ograniczenia CSS

- Tylko wizualizacja danych (a nie np. przystawienie w innym formacie)
- Niemożliwe (w CSS Level 2)
 - Powtórzenie tego samego fragmentu wielokrotnie
 - Rozróżnienie elementów ze względu na ich zawartość
 - Zaawansowane warunki logiczne
 - Wyświetlenie (nieznanym a priori) nazw elementów i atrybutów
 - Przetwarzanie danych (np. zsumowanie wartości liczbowych)
 - Dostęp do wielu dokumentów na raz
- Trudne lub nienaturalne w CSS
 - Wyświetlenie wartości atrybutów
 - Wypisanie elementów w innej kolejności niż w dokumencie
 - Wyróżnienie elementów nie spełniających pewnego warunku