

# Historia rozwoju technik znakowania tekstu

Patryk Czarnik

Instytut Informatyki UW

XML i nowoczesne technologie zarządzania treścią – 2007/08

- 1 Idea znakowania
  - Prehistoria
  - Kategorie znakowań

- 2 Historia
  - Historyczne pomysły
  - Droga do SGML
  - Droga do XML

- 3 Idee SGML i XML
  - SGML
  - XML

# To jest XML (1)

```
<wykład tytuł="Historia technik znakowania tekstu">  
  <slajd tytuł="To jest XML (1)">  
    <par>  
      XML jest standardem do zapisywania danych  
      tekstowych <ważne>wraz z ich strukturą</ważne>.  
    </par>  
  </slajd>  
</wykład>
```

## To jest XML (2)

```
<?xml version="1.0" encoding="iso-8859-2"?>
<!-- Komentarz -->
<element_glowny>
  <podelement atrybut='Wartość atrybutu'
    inny-atrybut="123">
    Zawartość tekstowa <elem>i zanurzony element</elem>.
    <element_pusty moze_miec="atrybut"/>
  </podelement>
  Zawartość tekstowa &encja; &#502;
  <![CDATA[x &lt; 5 & x > -5]]>
</element_glowny>
```

# Plan

- 1 Idea znakowania
  - Prehistoria
  - Kategorie znakowań

- 2 Historia
  - Historyczne pomysły
  - Droga do SGML
  - Droga do XML

- 3 Idee SGML i XML
  - SGML
  - XML

# Znakowanie tekstu – źródła

- *Markup* – znakowanie.
- Źródła: ręczne znakowanie tekstu przeznaczonego do druku.

# Znakowanie tekstu – źródła

- *Markup* – znakowanie.
- Źródła: ręczne znakowanie tekstu przeznaczonego do druku.

*wytfuścić*

Hamlet

Być albo nie być, oto jest pytanie.

*wcięcie*

# Znakowanie tekstu – kategorie

- Znakowanie interpunkcyjne i prezentacyjne.
- Znakowanie proceduralne.
- Znakowanie opisowe.
- Znakowanie referencji.

# Znakowanie interpunkcyjne i prezentacyjne

- Nawet pisząc „płaski tekst” (np. nieformatowany email) używamy znaczników:
  - znaki interpunkcyjne,
  - wielkość liter,
  - odstępy w poziomie i pionie,
  - „ręczne” wypunktowania, numeracje itp.
- Informacja ta może być wykorzystana do odtworzenia (także automatycznego) struktury tekstu.

# Znakowanie proceduralne

- Znaczniki powodują określone zachowanie programu czytającego, zazwyczaj zastosowanie określonego formatowania.
- Przykłady:
  - Postscript, TeX,
  - częściowo LaTeX, np. `\tt`, `\center`,
  - bezpośrednie formatowanie w Wordzie,
  - znaczniki HTML takie jak `<B>` czy `<BR>`.

# Wady bezpośredniego formatowania

- Dokument w danej chwili posiada tylko jedno formatowanie, zmiana formatowania wymaga zmian w dokumencie.
- Brak pewnego sposobu na automatyczne rozróżnienie różnych znaczeniowo fragmentów tekstu o tym samym formatowaniu.
  - Przykład: zarówno cytaty jak i wyróżnienia oznaczamy kursywą, następnie chcemy wszystkie wyróżnienia oznaczyć pogrubieniem.

# Znakowanie opisowe (strukturalne, semantyczne)

- Znaczniki opisują rolę fragmentów tekstu:
  - struktura dokumentu (rozdział, paragraf, lista),
  - znaczenie fragmentów tekstu (definicja, cytat, osoba).
- Przykłady:
  - częściowo LaTeX, np. `\section`, `\theorem`,
  - style w Wordzie,
  - znaczniki HTML takie jak `<H1>`, `<P>`, `<Q>`, `<DFN>`.

# Zalety znakowania opisowego

- Znajomość struktury znacznie ułatwia analizę i wyszukiwanie (możliwe wyszukiwanie kontekstowe).
- Możliwość stosowania wielu formatowań do tego samego dokumentu.
- Możliwość wykorzystania danego formatowania do wielu dokumentów.

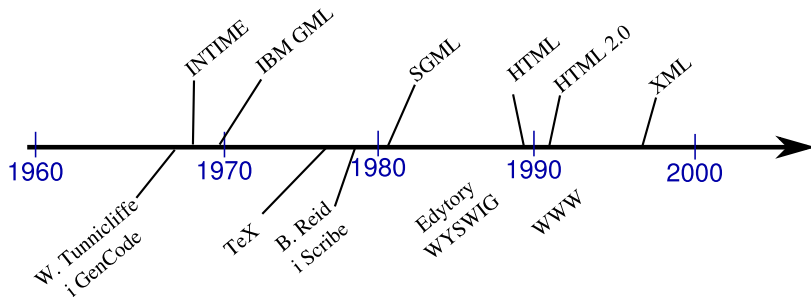
# Znakowanie referencji

- Zastosowania:
  - wstawianie znaków i fragmentów tekstu,
  - włączanie fragmentów z zewnętrznych źródeł bez kopiowania,
  - odnośniki do dokumentów i miejsc w dokumentach.
- Zalety:
  - możliwa zmiana wstawianych znaków i tekstów w zależności od kontekstu czy wybranego formatowania,
  - uaktualnianie fragmentów z zewnętrznych źródeł.

# Plan

- 1 Idea znakowania
  - Prehistoria
  - Kategorie znakowań
- 2 Historia
  - Historyczne pomysły
  - Droga do SGML
  - Droga do XML
- 3 Idee SGML i XML
  - SGML
  - XML

# Historia komputerowego znakowania tekstu



# Znakowanie zorientowane na wygląd

- Program RUNOFF (1964), następnie nroff i troff w pierwszych systemach UNIXowych.
- Edytory typu WYSWIG (tekstowe – koniec lat 60-tych, graficzne – lata 80-te).
- TeX (Donald Knuth, koniec lat 70-tych), LaTeX (początek lat 80-tych).
- Postscript (połowa lat 80-tych).
- RTF (1987).

# W. Tunncliffe i GenCode

## Pierwsze głosy dot. znakowania opisowego

- 1967 – William Tunncliffe, prezes Graphic Communications Association, podczas spotkania w Canadian Government Printing Office przedstawia ideę oddzielenia zawartości informacyjnej dokumentów od ich formatu,
- Stanley Rice proponuje użycie uniwersalnych znaczników do znakowania struktury tekstu.
- Projekt **GenCode** definiuje sposób oznaczania tekstu ukierunkowany na jego strukturę.

# B.Reid i Scribe

## Alternatywna droga

Brian Reid, pod koniec lat 70-tych opracowuje, a w 1981 przedstawia **Scribe**:

- język znaczników i system przetwarzania tekstu,
- wyraźne oddzielenie treści od formatu,
- idea arkusza stylu,
- podobno pomysły wzięte pod uwagę przy projektowaniu SGML.

# INTIME

## INteractive Textual Information Management Experiment

- Projekt badawczy Charlesa Goldfarba (IBM, koniec lat 60-tych XX wieku),
- Prototyp zintegrowanego systemu przetwarzania tekstu:
  - edycja tekstu,
  - repozytorium dokumentów,
  - wyszukiwanie.
- (Oczywiście) obsługa za pomocą konsoli tekstowej i ręcznie wpisywanych poleceń.

# Wnioski z projektu INTIME

- Wyszukiwanie jest efektywniejsze gdy znana jest struktura (znaczenie poszczególnych fragmentów) tekstu.
- Opracowano heurystykę odgadującą strukturę tekstu, ale zauważono potrzebę oznaczania struktury w źródle.
- Istniejące (wówczas) języki znakowania tekstu koncentrują się na wyglądzie, a nie strukturze czy znaczeniu.
- Charles Goldfarb bierze udział w pracach nad standardem GML a później SGML.

# GML – Generalized Markup Language

- Początki: 1969; Charles Goldfarb, Edward Mosher, Raymond Lorie.
- Powstał jako język makr do edytora IBM SCRIPT:
  - opisujących strukturę dokumentu,
  - zamienianych na znaczniki formatujące.
- Możliwe było rozszerzanie początkowego zbioru znaczników.
- Narzędzie pozwalało na definiowanie wielu „profilu” wizualizujących dokument.

# SGML – Standard Generalized Markup Language

- Pierwsze wersje robocze w 1980.
- Standard ISO w 1986.
- Rozwinięty potomek GML.
- Domyślnie w SGML znaczniki z trójkątnymi nawiasami.
- Dopuszczalne także znaczniki jak w GML:

```
:ol
  :li.Ordered lists (like this one),
  :li.Unordered lists, and
  :li.Definition lists
:eol.
```

# SGML – Ważne zastosowania

- Pierwsze szerzej znane zastosowania:
  - Electronic Manuscript Project, Association of American Publishers, 1987,
  - CALS – Computer-Aided Acquisition and Logistic Support, US Department of Defense, 1988.
- HTML 2.0 zdefiniowany jako zastosowanie SGML w 1991.
- DocBook (początki – 1991).
- Standardy związane z SGML:
  - DSSSL – Document Style Semantics and Specification Language,
  - HyTime – meta-notacja dla linków oraz opis struktur multimedialnych rozciągniętych w czasie.

# Ewolucja internetu

- 1 człowiek ↔ człowiek,
- 2 człowiek ↔ aplikacja,
- 3 aplikacja ↔ aplikacja.

# Ewolucja internetu

- 1 człowiek ↔ człowiek,
- 2 człowiek ↔ aplikacja,
- 3 aplikacja ↔ aplikacja.

Elektroniczna wymiana danych wymaga dobrze określonego, uniwersalnego i taniego w obsłudze standardu.

# Dlaczego nie SGML?

Spodziewane użycie SGML:

- duże scentralizowane systemy przetwarzające dokumenty,
- przetwarzanie tekstu i publikacje,
- człowiek ręcznie edytujący dokumenty.

# Dlaczego nie SGML?

## Spodziewane użycie SGML:

- duże scentralizowane systemy przetwarzające dokumenty,
- przetwarzanie tekstu i publikacje,
- człowiek ręcznie edytujący dokumenty.

## Nowe wyzwania (XML):

- architektura rozproszona, „lekkie” komponenty,
- sposób reprezentacji danych
  - elektroniczna wymiana danych,
  - bazy danych,
- dokumenty przetwarzane wyłącznie automatycznie.

# World Wide Web Consortium (W3C)

- Kuźnia standardów internetowych, np.:
  - HTML – Hyper Text Markup Language,
  - HTTP – Hyper Text Transfer Protocol,
  - CSS – Cascading StyleSheets,
- XML – Extensible Markup Language:
  - najważniejsza rekomendacja ostatnich lat,
  - twórcy: Tim Bray (Netscape), Jean Paoli (Microsoft), C.M. Sperberg-McQueen (University of Illinois).
- Obecnie głównie prace nad standardami związanymi z XML.

# Plan

- 1 Idea znakowania
  - Prehistoria
  - Kategorie znakowań
- 2 Historia
  - Historyczne pomysły
  - Droga do SGML
  - Droga do XML
- 3 **Idee SGML i XML**
  - **SGML**
  - **XML**

# Idea SGML i XML – znaczniki opisowe

- Znaczniki mówią o *znaczeniu* a nie wyglądzie tekstu.
- Znaczniki otwierające i zamykające.

# Idea SGML i XML – znaczniki opisowe

- Znaczniki mówią o *znaczeniu* a nie wyglądzie tekstu.
- Znaczniki otwierające i zamykające.

```
<OSOBA MÓWIAĆA>Hamlet</OSOBA MÓWIAĆA>  
<WYPOWIEDŹ>Być albo nie być.  
    Oto jest pytanie.</WYPOWIEDŹ>
```

# Idea SGML i XML – formatowanie

Informacja o wyglądzie poza treścią dokumentu.

- OSOBA MÓWIĄCA – Arial 14 pogrubiony,
- WYPOWIEDŹ – Times 12 normaly, wcięcie 1.5 cm.

# Idea SGML i XML – formatowanie

Informacja o wyglądzie poza treścią dokumentu.

- OSOBA MÓWIĄCA – Arial 14 pogrubiony,
- WYPOWIEDŹ – Times 12 normaly, wcięcie 1.5 cm.

## Hamlet

Być albo nie być.

Oto jest pytanie.

# Idea SGML i XML – model

- Można i należy dostosować zestaw znaczników do problemu.
- **Model** definiuje słownik pojęć i zależności między nimi (analogia: klasy w UML).
- SGML i XML pozwala na ograniczenie struktury dokumentu zgodnie z opracowanym modelem (Definicja Typu Dokumentu).

# Idea SGML i XML – model

- Można i należy dostosować zestaw znaczników do problemu.
- **Model** definiuje słownik pojęć i zależności między nimi (analogia: klasy w UML).
- SGML i XML pozwala na ograniczenie struktury dokumentu zgodnie z opracowanym modelem (Definicja Typu Dokumentu).

```
<OSOBA MÓWIAĆA>Hamlet</OSOBA MÓWIAĆA>  
<WYPOWIEDŹ><NUDA>Być albo nie być.  
    Oto jest pytanie.</NUDA></WYPOWIEDŹ>
```

# Najodpowiedniejszy model

- **encyklopedia:** <nazwisko>, <imie>, <ur>, <zm>, <wymowa>, <etymologia>, <liczba-mieszk>
- **prawo:** <promulgator>, <rocznik>, <poz>, <art>, <sąd>, <sygn-wyroku>, <teza>
- **dokument techniczny:** <part-number>, <function-name>
- **patenty:** <wynalazca>, <nr-zgłoszenia>

# Język – metajęzyk

- Stan wyjściowy: Wieża Babel (brak wspólnego języka).
- Wspólny metajęzyk:
  - znana gramatyka,
  - jednolita metodologia,
  - takie same narzędzia.
- Dowolnie wiele języków specyficznych dla zastosowań.

# SGML/XML a HTML

## HTML

- Zestaw znaczników określony przez standard i zamknięty.
- Znaczenie znaczników określone przez standard.
- Wygląd znaczników określony przez standard lub przeglądarki.
- Poprawność w praktyce weryfikowana przez przeglądarki (ile stron się waliduje?...).

## XML/SGML

- Możliwość definiowania nowych znaczników.
- Ta sama nazwa znacznika może mieć różne znaczenie w różnych zastosowaniach, np. kod – kod źródłowy programu, kod pocztowy.
- Można przypisać dowolny wygląd do określonych znaczników (jeśli w ogóle potrzeba).
- Poprawność ściśle określona przez standard.

# SGML/XML a HTML

## HTML

- Zestaw znaczników określony przez standard i zamknięty.
- Znaczenie znaczników określone przez standard.
- Wygląd znaczników określony przez standard lub przeglądarki.
- Poprawność w praktyce weryfikowana przez przeglądarki (ile stron się waliduje?...).

## XML/SGML

- Możliwość definiowania nowych znaczników.
- Ta sama nazwa znacznika może mieć różne znaczenie w różnych zastosowaniach, np. kod – kod źródłowy programu, kod pocztowy.
- Można przypisać dowolny wygląd do określonych znaczników (jeśli w ogóle potrzeba).
- Poprawność ściśle określona przez standard.

# SGML/XML a HTML

## HTML

- Zestaw znaczników określony przez standard i zamknięty.
- Znaczenie znaczników określone przez standard.
- Wygląd znaczników określony przez standard lub przeglądarki.
- Poprawność w praktyce weryfikowana przez przeglądarki (ile stron się waliduje?...).

## XML/SGML

- Możliwość definiowania nowych znaczników.
- Ta sama nazwa znacznika może mieć różne znaczenie w różnych zastosowaniach, np. kod – kod źródłowy programu, kod pocztowy.
- Można przypisać dowolny wygląd do określonych znaczników (jeśli w ogóle potrzeba).
- Poprawność ściśle określona przez standard.

# SGML/XML a HTML

## HTML

- Zestaw znaczników określony przez standard i zamknięty.
- Znaczenie znaczników określone przez standard.
- Wygląd znaczników określony przez standard lub przeglądarki.
- Poprawność w praktyce weryfikowana przez przeglądarki (ile stron się waliduje?...).

## XML/SGML

- Możliwość definiowania nowych znaczników.
- Ta sama nazwa znacznika może mieć różne znaczenie w różnych zastosowaniach, np. `kod` – kod źródłowy programu, kod pocztowy.
- Można przypisać dowolny wygląd do określonych znaczników (jeśli w ogóle potrzeba).
- Poprawność ściśle określona przez standard.

# XML – uproszczenie SGML

- Każdy *poprawny strukturalnie* dokument XML jest też dokumentem SGML.
- Uproszczenie składni i uproszczenie procesu parsowania.
- Nie są dopuszczalne pewne konstrukcje ułatwiające ręczne tworzenie dokumentów w SGML.
- Ten sam dokument zapisany w składni XML może zajmować więcej miejsca niż w składni SGML.

# Dwie twarze XML

## Dokumenty, publikacje...

- Pierwotne zastosowanie SGML.
- Człowiek tworzy i czyta dokument.
- Typowy mieszany model zawartości z wieloma opcjami.
- Dokumenty przechowywane przez dłuższy czas.

## Elektroniczna wymiana danych

- Nowe zastosowanie XML.
- Dokumenty przetwarzane automatycznie.
- Precyzyjna „bazodanowa” struktura.
- Dokumenty tworzone na czas komunikacji.

# Dwie twarze XML

## Dokumenty, publikacje...

- Pierwotne zastosowanie SGML.
- Człowiek tworzy i czyta dokument.
- Typowy mieszany model zawartości z wieloma opcjami.
- Dokumenty przechowywane przez dłuższy czas.

## Elektroniczna wymiana danych

- Nowe zastosowanie XML.
- Dokumenty przetwarzane automatycznie.
- Precyzyjna „bazodanowa” struktura.
- Dokumenty tworzone na czas komunikacji.

# Dwie twarze XML

## Dokumenty, publikacje...

- Pierwotne zastosowanie SGML.
- Człowiek tworzy i czyta dokument.
- Typowy mieszany model zawartości z wieloma opcjami.
- Dokumenty przechowywane przez dłuższy czas.

## Elektroniczna wymiana danych

- Nowe zastosowanie XML.
- Dokumenty przetwarzane automatycznie.
- Precyzyjna „bazodanowa” struktura.
- Dokumenty tworzone na czas komunikacji.

# Dwie twarze XML

## Dokumenty, publikacje...

- Pierwotne zastosowanie SGML.
- Człowiek tworzy i czyta dokument.
- Typowy mieszany model zawartości z wieloma opcjami.
- Dokumenty przechowywane przez dłuższy czas.

## Elektroniczna wymiana danych

- Nowe zastosowanie XML.
- Dokumenty przetwarzane automatycznie.
- Precyzyjna „bazodanowa” struktura.
- Dokumenty tworzone na czas komunikacji.