

# XML

I.Savnik, FAMNIT

# Ozadje

- 1978 - ANSI (*American National Standards Institute*) - ustanovljena skupina za pripravo standarda jezika za opis besedil.
- 1984 - ANSI in ISO (*International Standards Organization*) - priprava mednarodnega standarda. Osnutek je bil objavljen leta 1985, naslednje leto pa tudi sam standard SGML.

# Ozadje

- SGML je bil že med nastajanjem in ob samem začetku podprt z dvema večjima projektoma:
  - 1983-1987 je delovna skupina pri *Association of American Publishers* pripravila v SGML opise zvrsti *knjiga*, *časopis* in *članek*
    - **EMP** (Electronic Manuscript Project)
  - 1987 *US DoD* zahteva enotno tehnično dokumentacijo vseh naročenih izdelkov – **CALS** (Continuous Acquisition and Life-cycle Support)
- 1990 – nov projekt – **HTML** (Tim Berners-Lee uporabi SGML in zgradi DTD (specifična množica označb) za hipertekst – *podobno lahko obravnavamo LaTeX, pdflatex, html2latex,... kot ~SGML/XML s specifičnim DTD*)

# SGML - HTML

- HTML – specifična fiksna množica značk
- HTML – opis prezentacije
- SGML - ločuje med **obliko in vsebino**
- SGML - objavljanje (*publishing*)
- SGML – elektronski dokumenti, arhivi, knjižnice

# Zakaj XML ?

- Kompleksnost SGML
- Definicija strukture dokumenta/podatkov
  - HTML ne omogoča
- Jezik za opis podatkov
- Prenos podatkov med aplikacijami  
(v organizacijah, med organizacijami)
- Shranjevanje/prenos podatkov na spletu

# XML – zahteve

- XML mora biti neposredno uporaben na spletu
  - XML mora podpirati veliko množico uporab
  - XML mora biti skladen s SGML
  - Pisanje programov, ki procesirajo XML dokumente, mora biti preprosto
  - Število dodatnih (opcijskih) elementov XML naj bo minimalno, idealno nič
1. XML dokumenti morajo biti berljivi in jasni
  2. Priprava modela XML jezika mora biti hitra
- XML mora biti formalen in jedrnat Priprava XML dokumentov mora biti preprosta
1. Razumljivost XML dokumentov je bistvenega pomena (in zgoščenost minimalnega pomena)

# <Tag – značka>

- Očitno: označevalni jeziki uporabljajo značke
- Ideja označevanja je zelo stara: presledki (*white spaces*), označevanje začetka in konca stavka, odstavki, ..., uredniški popravki,

Cap i have learned to look on nature not as in the Bf  
hour of thoughtless youth but hearing oftentimes the less #  
still, sad music of humanity, nor harsh nor grating, ital  
though of ample power to chasten and subdue. Bf  
ital





# <Tag – značka>

- Značke:
  - Postopkovno označevanje: značka določa **kaj se zgodi** z označenim delom besedila (WordStar, WorfPerfect, Word, PS, RTF,...)
  - Vsebinsko označevanje:
    - opisno – oblikovanje glede na izbran stil ((La)TeX, Word, ...)
    - posplošeno – določitev značilnih sestavin in njihove medsebojne povezanosti – tvorimo *znakovno podatkovno bazo* (SGML, XML)

# <Tag – značka>

- SGML navodilo: značka se naj začne z znakom < in konča z znakom >
- Zakaj tekstovne značke? Večja prenosljivost med različnimi okolji! Boljša berljivost!

# XML

- XML je namenjen
  - Strukturiranju,
  - shranjevanju in
  - prenosu podatkov
- XML opisuje podatke in zveze med podatki
- XML nima definiranih značk
  - značke definiramo sami - slovnica
  - XML je množica podobnih jezikov (prim. kontekstno neodvisni jeziki)
- XML uporablja DTD ali XML Schema za opis strukture podatkov - **slovnice**

# XML – zgradba jezika

- **Struktura + Vsebina**
- DTD (*Document Type Definition* – zvrst spisa) definira **strukturo** zvrsti dokumenta
  - slovnica  $G$  in organizacija značk
- XML dokument, ki uporablja te značke za označevanje **vsebine**
  - beseda  $w$ , ki je v jeziku  $L(DTD)$

# XML – struktura dokumenta

- Uvod (*Prolog*) – ukazi za definiranje razpoznavalnika (*parser*)
  - slovnica, *G*
- *Telo* – vsebina, pomembna za ljudi
  - beseda *w*
- Zaključek (*Epilog*) – zaključni komentarji

# XML - jezik

- XML dokument je beseda  $w$  v jeziku, ki ga definira slovnica DTD  $G=(\Sigma, \Gamma, S, P)$ :
  - vsak dokument ima korenski element ( $S$ )
  - med elementi veljajo odnosi kot jih definirajo produkcije  $P$  (drevo izpeljave)
  - simbole ( $X \in \Gamma$ ) imenujemo tudi imenujejo **veje**, črke ( $a \in \Sigma$ ) pa **listi**
  - elementi lahko vsebujejo attribute ali prilastke, ki jim dodajo funkcionalnost (npr. podajajo opise, omogočajo iskanje ipd.)
- Napake v besedilu bodo programsko opremo ustavile
  - prevajanje
  - do napake pride, če  $w \notin L$  (DTD)

# XML - osnovni gradniki

- Elementi
  - Značke – opisujejo podatkovni objekt
- Atributi (prilastki)
  - Opisujejo lastnosti objektov - elementov
- Entitete
  - Deli skupnega teksta
- Komentarji

# XML - primer

```
<?xml version="1.0"?>
```

```
<!-- File Name: Inventory.xml -->
```

```
<inventory>
```

```
  <book genre="comp" id="b724">
```

```
    <title>Database Management Systems</title>
```

```
    <author><firstname>Raghu</firstname>
```

```
      <lastname>Ramakrishnan</lastname>
```

```
    </author>
```

```
    <publisher>McGraw Hill</publisher>
```

```
    <year>2000</year>
```

```
  </book>
```

```
  ...
```

```
</inventory>
```



# XML - osnovna pravila

- Vsebuje deklaracijo dokumenta
- Vsaka začetna značka mora imeti tudi pripadajočo končno značko
- Strukture morajo biti pravilno gnezdene
- Vrednosti atributov so navedene v narekovajih
- Na najvišjem nivoju je lahko le en element – korenski element (S)

# XML - veljaven dokument

- XML dokument je **dobro oblikovan**, če veljajo v njem osnovna pravila
  - Ima pravo obliko in strukturo
  - Preverjanje vsebine (telo XML dokumenta)
- XML dokument je **veljaven**, če:
  - je dobro oblikovan in
  - njegova struktura ustreza slovnici (DTD definiciji)
    - Preverjanje DTD in XML
    - Beseda jezika  $L$  (DTD)

# XML - atribut

- Alternativni način opisa informacij
  - Opis lastnosti elementa
- Vrednost atributa je v narekovajih “”

```
<book genre="comp">
```

# XML - identifikatorji

- Zgolj oblikovni gradnik
- Veljajo znotraj celotnega dokumenta

```
<person id="o555"> <name>Janez</name> </person>
```

```
<person id="o456"> <name>Marija</name>
```

```
    <children idref="o123 o555"/>
```

```
</person>
```

```
<person id="o123" mother="o456"><name>Tone</name>
```

```
</person>
```

# DTD

- Document Type Descriptor
- Podedovan od SGML
- Podoben DB shemi, čeprav ni zares ...
- BNF slovnica (prim. kontekstno neodvisni jezik)
- Definicija elementov specifičnega XML jezika

# DTD - primer

```
<!element book (title,author*,publisher,year) >  
<!element title #PCDATA >  
<!element publisher #PCDATA >  
<!element year #PCDATA >  
<!element author (firstname,lastname,address?,age?) >  
<!attlist book id ID #required >  
<!attlist book genre CDATA #required >  
...
```

# DTD - zgradba

- Elementi
  - `<!ELEMENT element-name category>`
  - `<!ELEMENT element-name (element-content)>`
  - `E1,E2,... ; E* ; E+ ; E? ; E1 | E2`
  - Tipi: `#PCDATA`, `EMPTY`, `ANY`

# DTD - zgradba

- Atributi

- `<!ATTLIST el-name attr-name attr-type default-value>`
- Tipi: CDATA, (en1|en2|..), ID, IDREF, ENTITY, ...
- Določila: `#required`, `#implied`, `#fixed`



# DTD - zgradba

- Entitete

- `<!ENTITY entity-name "entity-value">`
- `<!ENTITY writer "Donald Duck."> -- uporaba:  
&writer;`

# DTD+XML vs. Relacije

- XML
  - Primeren bolj za dokumente kot podatke
  - Ime in tip elementov so povezani globalno
  - Ni podpore podatkovnim tipom
    - Ni validacije podatkov
  - Imamo lahko en sam ključ
    - Ni ključev z več atributi
    - Ni tujih ključev (reference na ključe)
    - Ni omejitev za IDREF
  - Ni podpore za ponovno uporabo strukture
    - OO strukture niso podprte
- Ni mogoče ohraniti informacije pri prevajanju iz relacijskega podatkovnega modela v XML

# XML Schema

- XML oblika DB sheme !
- Imena elementov in tipi so povezani lokalno
- Vsebuje osnovne podatkovne tipe (integer, string, date, itn.)
- Podpira omejitve vrednosti (integer > 100)
- Uporabniški podatkovni tipi
- Dedovanje (extension, restriction)
- Tuji ključi

# XML Schema - primer

```
<schema version="1.0"
  xmlns="http://www.w3.org/1999/XMLSchema">
  <element name="author" type="string" />
  <element name="date" type = "date" />
  <element name="abstract">
    <type>
      ...
    </type>
  </element>
  ...
  <element name="paper">
    <type>
      <attribute name="keywords" type="string"/>
      <element ref="author" minOccurs="0" maxOccurs="*" />
      <element ref="date" />
      <element ref="abstract" minOccurs="0" maxOccurs="1" />
      <element ref="body" />
    </type>
  </element>
</schema>
```

# XML – pomembni standardi

- CSS,XSL/XSLT:
    - Prezentacija in transformacija dokumentov in podatkov
  - RDF: resource description framework
    - Meta-info, kategorije, semantične mreže, ...
  - Xpath/Xpointer/Xlink:
    - standard za povezovanje dokumentov in elementov
  - Namespaces:
    - Delo z imeni
  - DOM: “Document Object Model”
    - Delo z XML dokumenti v prog. jezikih
  - SAX: Enostaven API za XML
  - XQL,Xquery, XML-QL: poizvedovalni jezik
- PMJ, 2011/12

# XQL

- Osnova
  - W3C standard
  - Rezultati izrazov so množice objektov
  - ./author, author, //author, /bookstore
- Dostop do otrok
  - Elementi: author/firstname, bookstore//title, bookstore/\*/title
  - Atributi: price/@exchange, price/@exchange/total, book[@style]
- Filtri
  - book[excerpt], book[author/degree]
  - author[degree \$and\$ award]
  - author[\$not\$ (degree \$or\$ award) \$and\$ publication]
- Funkcije
  - author[first-name!text() = 'Bob'], author[\*!text() = 'Bob']

# XML-QL – primer 1

- Izpiši avtorje in naslove vseh knjig založbe Addison-Wesley.

```
WHERE <book>
    <publisher>Addison-Wesley</publisher>
    <title>$t</title>
    <author>$a</title>
</book> IN www.bookstore.com/books.xml
CONSTRUCT <result>
    <author> $a</author>
    <title> $t</title>
</result>
```

# XML-QL – primer 2

- Izpiši leta in priimke avtorjev, ki so izdali knjigo na to leto.

```
WHERE <book>$e</book> IN www.bookstore.com/books.xml,  
    <author>$n</author>,  
    <year>$p</year> IN $e  
CONSTRUCT <result><year>$p</year>  
    WHERE <lastname>$l</lastname> IN $n  
    CONSTRUCT <lastname>$l</lastname>  
    </result>
```



# Uporaba Cascading Style Sheet

```
<?xml version="1.0"?>
```

```
<!-- File Name: Inventory02.xml -->
```

```
<?xml-stylesheet type="text/css" href="Inventory02.css"?>
```

```
<INVENTORY>
```

```
<BOOK>
```

```
<TITLE>The Adventures of Huckleberry Finn</TITLE>
```

```
<AUTHOR>Mark Twain</AUTHOR>
```

```
<BINDING>mass market paperback</BINDING>
```

```
<PAGES>298</PAGES>
```

```
<PRICE>$5.49</PRICE>
```

```
</BOOK>
```

```
<BOOK>
```

```
<TITLE>Leaves of Grass</TITLE>
```

```
<AUTHOR>Walt Whitman</AUTHOR>
```

```
<BINDING>hardcover</BINDING>
```

```
<PAGES>462</PAGES>
```

```
<PRICE>$7.75</PRICE>
```

```
</BOOK>
```

```
... PMJ, 2011/12  
</INVENTORY>
```

procesni ukaz



```
/* File Name: Inventory02.css */
```

```
BOOK
```

```
{display:block;  
margin-top:12pt;  
font-size:10pt}
```

```
TITLE
```

```
{display:block;  
font-size:12pt;  
font-weight:bold;  
font-style:italic}
```

```
AUTHOR
```

```
{display:block;  
margin-left:15pt;  
font-weight:bold}
```

```
BINDING
```

```
{display:block;  
margin-left:15pt}
```

*The Adventures of Huckleberry Finn*

**Mark Twain**  
mass market paperback  
\$5.49

*Leaves of Grass*

**Walt Whitman**  
hardcover  
\$7.75

*The Legend of Sleepy Hollow*

**Washington Irving**  
mass market paperback  
\$2.95

```
PAGES
```

```
{display:none}
```

```
PRICE
```

```
{display:block;  
margin-left:15pt}
```

# eXtensible Stylesheet Language

```
<?xml version="1.0"?>
<!-- File Name: XslDemo02.xml -->
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <H2>Book Inventory</H2>
    <xsl:for-each select="INVENTORY/BOOK">
      <SPAN STYLE="font-style:italic">Title: </SPAN>
      <xsl:value-of select="TITLE"/><BR />
      <SPAN STYLE="font-style:italic">Author: </SPAN>
      <xsl:value-of select="AUTHOR"/><BR />
      <SPAN STYLE="font-style:italic">Binding type: </SPAN>
      <xsl:value-of select="BINDING"/><BR />
      <SPAN STYLE="font-style:italic">Number of pages: </SPAN>
      <xsl:value-of select="PAGES"/><BR />
      <SPAN STYLE="font-style:italic">Price: </SPAN>
      <xsl:value-of select="PRICE"/><P />
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

XSL ID  
Definicija  
imenskega prostora

vzorec se nanaša na cel  
XML dokument

upoštevanje XML pravil

```
...
<?xml-stylesheet type="text/xsl" href="XslDemo02.xml"?>
...
```

## Book Inventory

*Title:* The Adventures of Huckleberry Finn  
*Author:* Mark Twain  
*Binding type:* mass market paperback  
*Number of pages:* 298  
*Price:* \$5.49

*Title:* The Adventures of Tom Sawyer  
*Author:* Mark Twain  
*Binding type:* mass market paperback  
*Number of pages:* 205  
*Price:* \$4.75

- Omogoča izločanje in sortiranje podatkov
- Dosegljiv je vsak podatek, četudi je podan s prilastkom

# Zaključek

- Podjetja uvajajo XML kot medij za prenos podatkov med oddelki
- Sistemi (distribuirani) uporabljajo XML kot medij za prenos podatkov
- Konfiguracijske datoteke, logi, ... sistemov so pogosto v XML
- Večina DBMS ima XML vmesnik

# Viri

- Peter Peer, Andrej Brodnik
  - prosojnice za 2003/04
- Raghu Ramakrishnan
  - Knjiga: Database Management Systems
- <http://www.w3.org/XML>  
<http://www.w3schools.com/>