

ՅՈՒՆԵՍԿՈ ԵՎ ՆԻԾՔԻՆԵՐԻ ՎԵՐԱԿՈՆԿՐԱԾ

Ժամ

ՌԻՆԶՅԱ ՄԻՆԻՍԿԻՆ

Otvoreno računarstvo

XML

- Uvod
- Pravila XML-a
- Primjene XML-a
- Prostori imena

Mario Žagar



Problem podataka

- Operacije nad podacima
(Open Information Management):
 - Interpretacija
 - Prezentacija
 - Interoperabilnost
 - Prenosivost
 - Pretraživanje

HTML i opis podataka



```
<p>
<b>00364244545 Marko Ferković - Enter</b><br>
<i>Student</i><br>
Sveučilište u Zagrebu<br>
<u>Fakultet elektrotehnike i računarstva</u>
Unska 3<br>
HR-10000 ZAGREB<br>
</p>
```

- HTML definira:
 - konačan broj oznaka i njihovu semantiku
 - strukturu podataka unutar dokumenta
 - oznake i struktura prilagođene su domeni prikaza podataka
- Jezik prilagođen autorima i korisnicima (ljudima)
- Računala služe samo za objavljivanje i prikaz, nema obrade!

Ograničenja HTML-a:

- Ne postoji čvrsta, formalna struktura dokumenta koje se autor mora držati, a koja bi omogućila verifikaciju
- Preglednici nastoje sami popraviti pogreške programera
 - jednostavnost stvaranja stranica plaćena složenošću pripreme prikaza
- Ne postoji fleksibilnost za opis podataka iz različitih domena:
 - proširivost oznaka kao mehanizma uvođenja novih pojmova
 - prilagodljivost strukture opisivanim podacima
- Problem uočen pojavom prvih strojnih “korisnika” web stranica
 - tumačenje značenja podataka na web stranici – čovjeku jednostavno, računalu vrlo teško (*common sense* je za današnju UI nedostižan cilj)

SGML

- Sveobuhvatan, složen standard za opis tipova dokumenata
- Vrlo opširan, velik broj neobaveznih elemenata
 - cijena implementacije
 - problemi u kompatibilnosti različitih implementacija
- HTML je tip dokumenta definiran SGML-om
- Za široku uporabu potrebna jednostavnija inačica SGML-a
 - jednostavna, učinkovita i prenosiva implementacija za što veći broj platformi i programskih jezika
 - mali, ali čvrsti skup pravila, podrška internacionalizaciji

XML - eXtensible Markup Language



- Mlađi (i manji) brat SGML-a
 - Primjena u opisu, pohrani i razmjeni različitih podataka u tekstualnom obliku zapisa
- Nema predefiniranih oznaka
 - Definiraju se po potrebi
- Formalan i sažet oblik
 - Mali broj jednostavnih, ali strogih pravila
- Jednostavnost obrade
 - Uvjetuje veće zahtjeve pri rukovanju sa sadržajem dokumenta (suprotno u odnosu na HTML)
- Orijentiran prema strojnoj obradi

Svojstva XML-a

- Proširivost:
 - oznake se definiraju po potrebi
- Odjeljivanje podataka od prezentacije:
 - oznake **opisuju podatke, ne njihov izgled**
- Validacija:
 - stroga pravila - strojna provjera valjanosti i strukture
- Internacionalizacija:
 - XML izvorno koristi Unicode (UTF-8)
- Prenosivost:
 - XML dokument je obična tekst datoteka
- Rasprostranjenost:
 - neovisan o platformi i podržan od većine prog. jezika

Primjer XML dokumenta:

```
<?xml version="1.0" encoding="UTF-8" ?>

<student jmbag="00364244545">
  <ime>Marko</ime>
  <prezime>Ferković - Enter</prezime>
  <institucija>
    <sveučilište>Sveučilište u Zagrebu</sveučilište>
    <fakultet id="0036">
      <ime>Fakultet elektrotehnike i računarstva</ime>
      <adresa>
        <ulica>Unska</ulica>
        <broj>3</broj>
        <pbr>HR-10000</pbr>
        <grad>Zagreb</grad>
      </adresa>
    </fakultet>
  </institucija>
</student>
```

XML i strukturiranje podataka

- XML ne propisuje oznake, već pravila definiranja
 - da li je XML meta-jezik?
- Korisnici dokumenta XML tumače značenje oznaka
 - što npr. znači oznaka `<ime>` ?
 - koje su sve oznake potrebne za opis podataka?
- Oznake definiraju hijerarhijske odnose podataka unutar dokumenta (odnos roditelj – dijete):

```
<student jmbag="00364244545">
  <ime>Marko</ime>
  <prezime>Ferković - Enter</prezime>
</student>
```

XML i opis podataka

- Jezik za opis podataka:
 - konačan broj oznaka (rječnik)
 - struktura podataka (tvorba rečenica)
 - semantika (značenje riječi)
- Konačan broj oznaka i struktura
 - određuju se tipom dokumenta,
 - tipovi se definiraju jezicima DTD ili XML Schema
- Semantika pojedinih oznaka definirana implicitno
 - mora biti dijeljena između svih entiteta koji koriste pojedini tip dokumenta (shared semantics)
- Eksplicitna definicija semantike – semantički web

XML i opis podataka

- Standardizacija jezika po domenama primjene
 - horizontalne domene
 - razmjena podataka između aplikacija u istoj domeni (npr. CAD sustavi, zdravstvo, e-business, logistika ...)
 - onemogućuje tzv. vendor-locking
 - vertikalne domene
 - opis podataka prisutnih u svim domenama ili neovisnih o domenama primjene (npr. stylesheets, SOAP ...)
 - razmjena podataka između različitih domena
 - npr. između zdravstva, osiguranja, MUP-a, biračkih popisa ...

Povijest XML-a

- HTML – beznadno izgubio prvobitnu namjenu
 - odvajanja prikaza od sadržaja, više narječja HTML-a (Netscape, Microsoft ...)
- Kasne 90-te
 - Bosak, Bray, Clark i drugi - prijedlog XML-a
- 1998-02-10 W3C preporuka, verzija 1.0
- Eksplozija tehnologija i primjena XML-a
- Aktualno:
 - četvrto izdanje XML preporuke iz kolovoza 2006.
 - postoji i XML 1.1 ! (nije u širokoj upotrebi)

Svijet XML-a

- Prateći jezici (preporuke i norme):
 - DTD, XML Schema, XSL, XSLT, XPath, XQuery, DOM, SAX, Namespaces, XLink, XPointer, ...
 - XHTML, WAP, ...
 - ODF, SMIL, SVG, CML, MathML, MusicML, OTA, ebXML, ...
 - XML-RPC, SOAP, WSDL, ...

Pravila XML-a

XML - primjer sadržaja dokumenta



procesna naredba

komentar

element korijen (root)

element dijete (child)

prazan element

atribut

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

```
<!-- komentar -->
```

```
<radnik>
```

```
  <ime> Marko </ime>
```

```
  <prazno/>
```

```
  <plaća valuta="kn">5000</plaća>
```

```
</radnik>
```


Element

`<ime_elementa> sadržaj </ime_elementa>`

- Element - osnovna gradivna komponenta XML dokumenta, sastoji se od:
 - početne oznake (`<ime_elementa>`),
 - završne oznake (`</ime_elementa>`) i
 - sadržaja omeđenog oznakama.
- Elementi se imenuju po entitetima domene čiji se podaci opisuju npr.:
 - jmbag, ime, prezime, predmet, ocjena...
 - iz domene poslovnih procesa na sveučilištu
- Imena elemenata su *case-sensitive*:
 - `<jmbag>` \neq `<JMBAG>`

Imenovanje elementa

- Valjana imena elementa:
[slova][_], [0-9][slova][znakovi]*
- Razmak se ne smije koristiti u tvorbi imena
- Ime elementa ne smije započeti s xml
 - u imenu se može koristiti većina Unicode znakova
- Prvi znak imena odmah nakon oznake "<"
- Primjeri dobro oblikovanih imena elemenata:
 <jmbag>, <first_name>, <first-name>, <šifra>, <_note>,
 <mbr >, <velika.seoba.naroda>, <صفٹ>
- primjeri krivih imena elemenata:
 <boja očiju>, <1.ples>, < jmbag>, <-HT-R-A>

↑

↑

↑

↑

Sadržaj elementa

```
<ime_elementa>
    ...sadržaj ...
</ime_elementa>
```

- Sadržaj elementa:
 - sve između početne i završne oznake
- Vrste sadržaja:
 - prazan
 - jednostavan
 - elementi djeca
 - miješani

Prazan sadržaj

```
<ime_elementa></ime_elementa>  
  
<ime_elementa/>
```

- Element nema sadržaja
- Moguće koristiti standardan ili skraćeni način zapisa praznog elementa

Jednostavan sadržaj



```
<ime_elementa>
```

Sadržaj u obliku teksta bez oznaka...

```
</ime_elementa>
```

- Podrazumijeva samo tekst
 - bez oznaka ili specijalnih karaktera
- Obraduje se od strane parsera
 - ne smije sadržavati znakove od značenja za parser (početak oznake, entiteta ...)
- Posebni znakovi:
<, >, &, " i ', **moraju**/mogu se nadomjestiti entitetima
<, **>**, **&**, **"** i **'**;

`<constr>x< 2*y</constr>`

`<constr>x< b> 2*y</constr>`

Elementi djeca

```
<vanjski_element>
    <unutarnji_element>
        ...
    </unutarnji_element>
    <unutarnji_element>
        ...
    </unutarnji_element>
</vanjski_element>
```

- Sadržaj
 - unutar elementa ugnježdjeni isključivo jedan ili više pod-elemenata (čije vrste sadržaja također mogu biti ...)
 - Očuvanje redosljeda otvaranja i zatvaranja elemenata
- `<element1><element2></element1></element2>` **KRIVO!**

Miješani sadržaj



```
<content>
```

```
    Predavanje će obraditi kvantitativnu metodu  
                                verifikacije homogenosti XML dokumenta  
u prisustvu                    ionizirajućeg zračenja, a koju su  
razvili
```

```
<name>Cholsky</name> i <name>Rangapathra</name> sa  
    sveučilišta <institution>UC Berkeley</institution>.
```

```
</content>
```

- Sadržaj se sastoji od teksta i elemenata u bilo kojem redoslijedu
- Za elemente unutar miješanog sadržaja vrijede ista pravila kao i za ostale elemente

Atributi



```
<ime_elementa ime_atributa="vrijednost" >  
    ...  
</ime_elementa>  
  
<prazan_element ime_atributa="vrijednost"/>
```

- Atribut mora pripadati elementu
- Element može imati jedan ili više atributa
- Atribut čini uređeni par ime="vrijednost"

Pravila oblikovanja atributa

- Pravila imenovanja atributa ista kao za elemente
- Unutar istog elementa atributi moraju imati različita imena
- Vrijednost atributa mora biti navedena unutar navodnika (jednostrukih ili dvostrukih)
- Sadržaj atributa može se sastojati samo od teksta
- Sadržaj atributa se obrađuje od strane parsera
- Razmak između riječi u sadržaju može se tretirati kao odjeljivanje dviju vrijednosti u nizu vrijednosti pojedinog atributa

`<flag colors="red white blue"/>`

Kada elementi, a kada atributi ?

- Sadržaj elementa
 - strukturirani podaci, jednostavno proširenje ili dodavanje višestrukih vrijednosti, jednostavniji za obradu
- Atribut pridijeljen elementu
 - podatak o elementu (meta podatak), primjer:


```
<lecture>
      <desc lang="en"> ..... </desc>
</lecture>
```
- Preporuke:
 - kad god je moguće, za zapis podataka koristiti elemente
 - attribute koristiti za meta podatke i za definiranje logičke strukture podataka

Naredbe obrade

```
<?ime_naredbe lista_atributa ?>
```

- eng. processing instruction
- Naredbe aplikacijama za obradu xml dokumenata
- Ime i atributi ovisni kojoj aplikaciji je naredba namijenjena
- Ako aplikacija ne razumije naredbu, ignorira je
- Primjer:
 - naredba za primjenu stylesheeta za formatiranje xml dokumenta

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

Komentari

```
<!-- ovo je jedan redak komentara -->  
  
<!-- komentar se proteže preko više redaka  
    ...  
-->
```

- Oznaka početka komentara **<!--**
- Oznaka kraja komentara **-->**
- Sadržaj omeđen početnom i krajnjom oznakom komentara je ignoriran od strane parsera
- Nema gniježđenja komentara
 - prva detekcija **-->** od strane parsera terminira komentar

CDATA blokovi

- Parser ignorira CDATA blokove
 - nema prepoznavanja elemenata, ekspanzije entiteta ...
- Označavanje blokova koji sadrže veći broj nedozvoljenih znakova
 - npr. dijelovi programskog kôda, formule ...
- Podaci unutar bloka ne smiju sadržavati slijed znakova `]]>`

```
<scenario-desc>
<![CDATA[
  action Seller in processCFP() {
    msgRcv : (CFP <- @Buyer1);
    msgSnd : ((PROPOSE | #REFUSE -> @Buyer1) &
              (REQUEST -> Supervisor));
  }
  action Buyer in processProposeRefuse() {
    msgRcv : (PROPOSE | #REFUSE <- <Seller>);
    msgSnd : (ACCEPT | #REJECT -> <Seller>);
  }
]]>
</scenario-desc>
```

Entiteti

- Entitet:
 - dio XML dokumenta
- XML dokument sastoji se od jednog ili više entiteta
- Vrste entiteta:
 - **parsirani**: datoteka s xml sadržajem
 - **neparsirani**: binarna datoteka
- Ugrađeni i naknadno definirani entiteti
- Unutarnji i vanjski entiteti
 - **unutarnji**: definirani unutar tipa dokumenta
 - **vanjski**: zaseban resurs, mora se prilikom parsiranja dohvatiti (slično `#include` direktivama u C/C++)

Referenciranje entiteta

- Referenciranje entiteta unutar dokumenta:
`&ime_entiteta;`
- Referenca se u postupku parsiranja zamjenjuje sa sadržajem entiteta
 - dobiva se parsirani dokument
- Dodavanje znakova u dokument,
 - npr. Unicode znak π : `π`
- Korištenje **naknadno** definiranog entiteta **π**
 entitet mora biti definiran u sklopu definicije tipa dokumenta

Ugrađeni entiteti

| | |
|---|------|
| < | lt |
| > | gt |
| & | amp |
| ' | apos |
| " | quot |

Bijeli znak (*Whitespace*)

- Različitost terminiranja retka teksta:
 - DOS & Windows: **CR/LF**, UNIX: **LF**, Mac: **CR**
- Preporučeno korištenje LF za terminiranje retka
- Kao bijeli znakovi tretiraju se skupno:
 - razmak (' '), tab ('\t'), CR(ASCII 13, '\r'), LF (ASCII 10, '\n')
- Dvije vrste bijelih znakova:
 - **važan**: dio sadržaja dokumenta, mora biti sačuvan nakon parsiranja
 - **nevažan**: primjena u formatiranju dokumenta, ne mora biti sačuvan nakon parsiranja
- Svi bijeli znakovi se, u općem slučaju, čuvaju tijekom parsiranja dokumenta

Ekvivalentnost XML dokumenata



- **Logička ekvivalentnost:**

- jednak sadržaj dokumenata

```
<student><jmbag>003643443</jmbag><ime>Marko</ime>  
<prezime>Ferković - Enter</prezime></student>
```

```
<student>
```

```
<ime>Marko</ime>
```

```
<prezime>Ferković - Enter</prezime>
```

```
<jmbag>003643443</jmbag>
```

```
</student>
```

- **Fizička ekvivalentnost:**

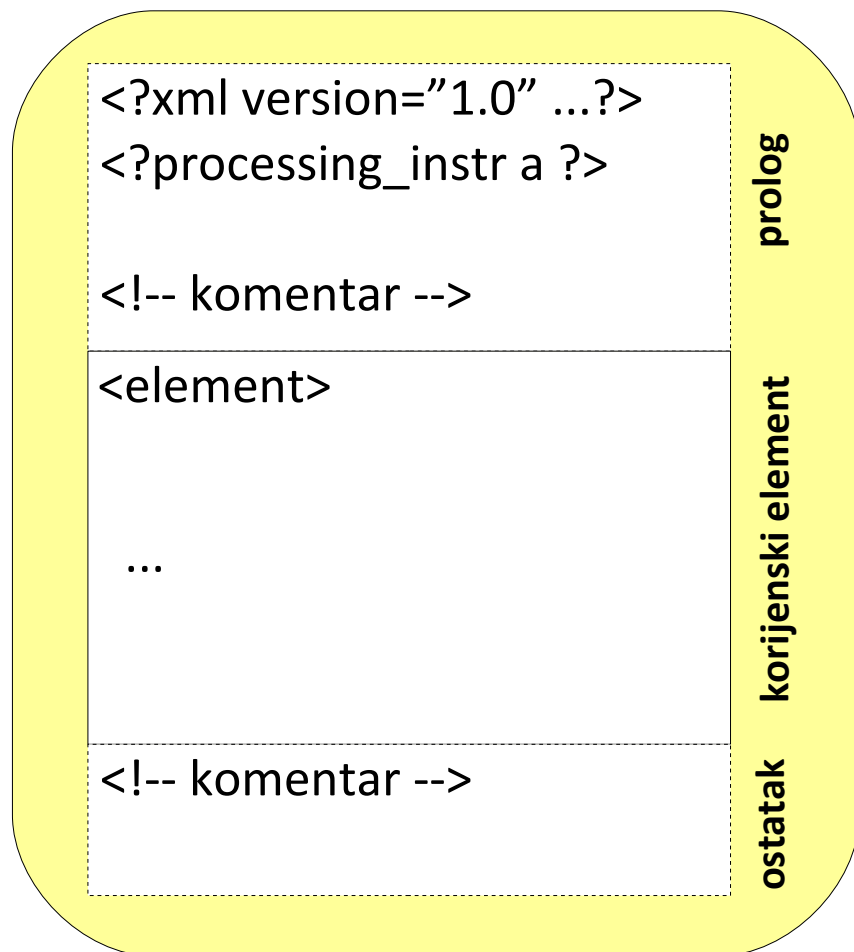
- datoteke jednake na razini okteta (npr. korištenje alata `diff` ne pokazuje razlike između datoteka)
- svođenje na **kanonički oblik** (struktura, poredak atributa, kodiranje ...) za usporedbu fizičke ekvivalentnosti

XML deklaracija

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
```

- Navođenje deklaracije na početku XML dokumenta
 - preporučeno (ali nije nužno)
- Znak **<** mora biti prvi znak unutar teksta
- Atribut version mora biti naveden kao prvi atribut,
 - vrijednost mora biti "1.0"
- Atribut encoding je neobavezan, vrijednost mora sadržavati oznaku kodiranja dokumenta
 - ISO-*, Unicode, UTF, ...
- Atribut standalone je neobavezan, označava ovisnost dokumenta o vanjskim entitetima

Struktura XML dokumenta



- XML dokument sastoji se od tri osnovna dijela:
 - prologa (neobavezan)
 - korijenskog elementa (obavezan)
 - ostatka dokumenta (neobavezan)

Prolog i ostatak dokumenta

- Prolog dokumenta:
 - xml deklaracija
 - deklaracija tipa dokumenta
 - komentari
 - naredbe obrade
 - whitespace
- Ostatak dokumenta:
 - naredbe obrade
 - komentari
 - whitespace

Korijenski element

```
<?xml version="1.0" encoding="UTF-8"
<doc>
    <content readonly='true'>
        <chapter> ... </chapter>
    </content>
    <content> ... </content>
</doc>
```

- Mora postojati isključivo jedan osnovni, korijenski element dokumenta
- Sadržaj dokumenta je unutar korijenskog elementa, u obliku elemenata, atributa, teksta itd.
- Unutar korijenskog elementa moguće navođenje
 - komentara, procesnih naredbi, whitespace, CDATA blokova, entiteta itd.

Dobro oblikovani XML dokument



- Dobro oblikovani (eng. well-formed) XML dokument zadovoljava sva prethodno navedena pravila strukture dokumenta, imenovanja elemenata i atributa, gnježđenja elemenata ...
- Parseri odbijaju obradu loše oblikovanih XML dokumenata

Primjer dobre oblikovanosti

- Primjer dobro oblikovanog XML dokumenta:
`<ž/>`
- Kolika je dužina ovog dokumenta u oktetima?

Ovisi ;) `<ž/>`

ASCII: ne ide!

ISO-8859-1 ne ide!

ISO-8859-2: 4 okteta

`3C BE 2F 3E`

UTF-8: 8 okteta

`EF BB BF 3C C5 BE 2F 3E`

UTF-16: 10 okteta

`FF FE 3C 00 7E 01 2F 00 3E 00`

- Koji je kod za znak `ž` u Unicode? `0x017E`

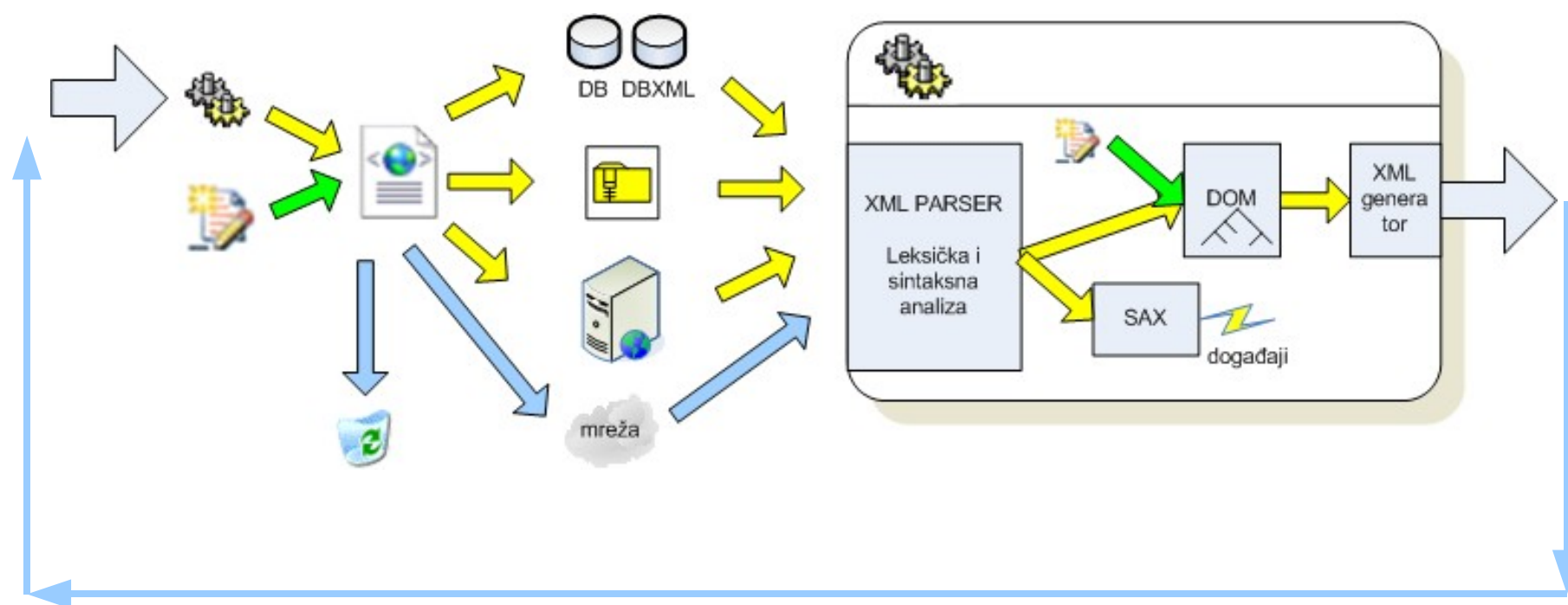


Primjene XML-a

Alati za pregled i uređivanje XML dokumenata

- **Prikaz XML dokumenata:**
 - bilo koji uređivač teksta opće namjene - samo sadržaj
 - Netscape/Firefox, Internet Explorer, Opera - sadržaj i struktura dokumenta, verifikacija, formatiranje, transformacije
- **Alati za uređivanje:**
 - uređivači teksta opće namjene (npr. notepad, vi, joe)
 - prilagođeni uređivači (npr. XMLNotepad)
 - složeni alati za razvoj (npr. XMLSpy)

"Vrtlog života" XML dokumenta



Pohrana podataka

- Skalabilnost:
 - jednostavne datoteke (npr. konfiguracijske datoteke aplikacije, *ant build* datoteke)
 - pohrana složenih dokumenata, korištenih isključivo od strane aplikacija (npr. ova prezentacija)
- XML datoteke stvarane od
 - čovjeka (u pravilu jednostavne)
 - aplikacija (od jednostavnih do vrlo složenih)
- XML datoteke korištene gotovo isključivo od aplikacija

Životni vijek XML dokumenata



- XML dokumenti po duljini životnog vijeka:
 - privremeni dokumenti (eng. transient)
 - trajni dokumenti (eng. persistent)
- Privremeni dokumenti se ne arhiviraju
 - ostaju u radnoj memoriji ili putuju mrežom u obliku poruka) ili se privremeno zapisuju u datotečnom sustavu tek radi razmjene s drugim aplikacijama
- Trajni dokumenti arhiviraju se u datotečnom sustavu ili u bazi podataka
 - mogu biti dostupni i putem mreže (web, ...)

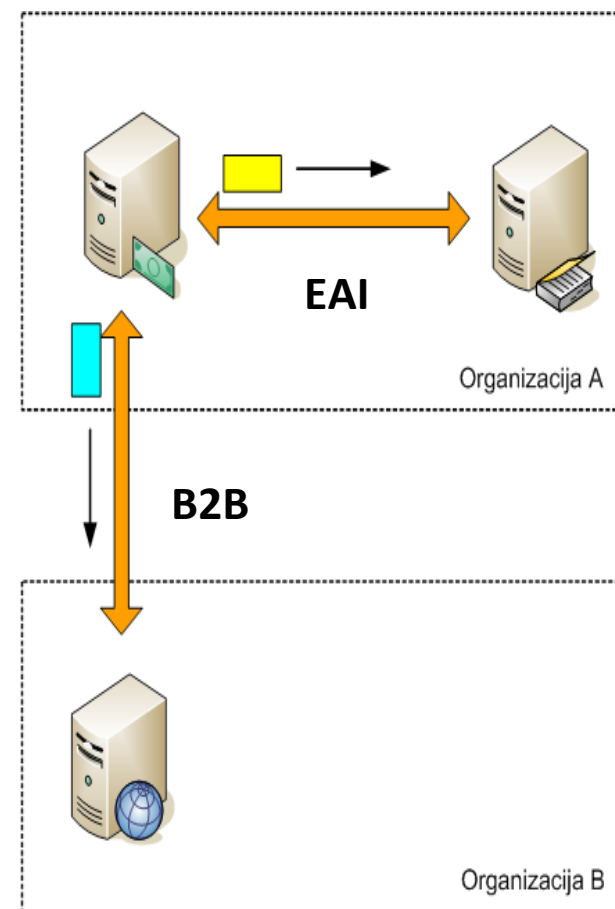
Arhiviranje XML dokumenata

- Izvorne XML datoteke su tekstualne datoteke
 - mogu se koristiti alati za održavanje verzija (npr. subversion)
- XML datoteke mogu biti sažete
 - više srodnih datoteka može biti sažeto unutar iste arhive (npr. ova prezentacija: XML.odp)
 - Vrlo visok stupanj sažimanja XML datoteka
- Baze podataka:
 - nativeXML – hijerarhijske baze podataka
 - relacijske baze podataka s mapiranjem
XML \leftrightarrow relacijski model
- Pretraživanje dokumenata i baza dokumenata

Integracija heterogenih aplikacija



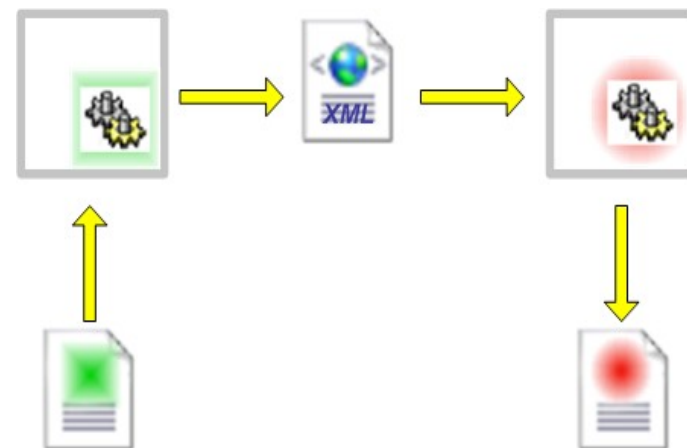
- Različite aplikacije unutar organizacije moraju razmjenjivati podatke (npr. skladište i financije)
 - EAI (Enterprise Application Integration)
- Različiti sustavi različitih organizacija moraju razmjenjivati podatke, npr.:
 - e-narudžbe, e-plaćanje, e-PDV
 - B2B, B2G
- Asinkrona komunikacija mrežom slanjem poruka u obliku privremenih XML dokumenata



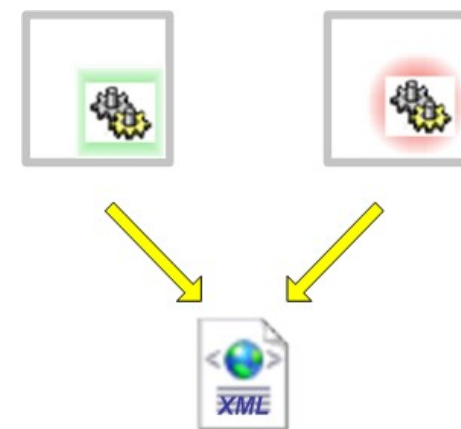
Prenosiva reprezentacija podataka



- Standardizirani format dokumenta neovisan o aplikaciji, jeziku, platformi, načinu prijenosa i/ili arhiviranja
- Omogućuje razmjenu dokumenata između aplikacija kao međufORMAT
- Omogućuje suradnju aplikacija na istom dokumentu ili parcijalno uređivanje dijelova dokumenta



razmjena dokumenata

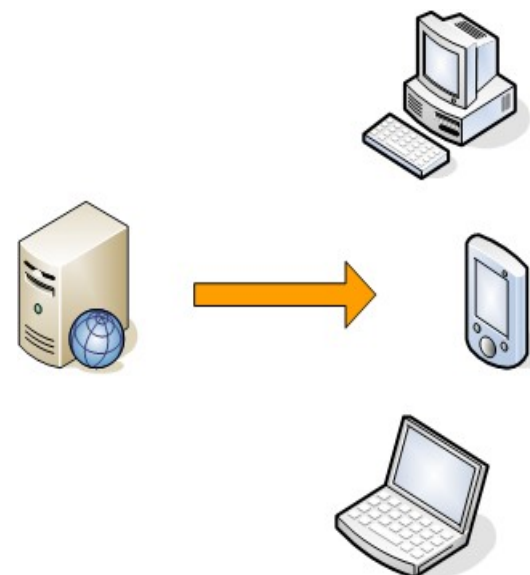


suradnja na istom dokumentu

Objavljivanje i prikaz dokumenta



- XML odvaja podatke i njihov prikaz
- Različiti prikazi XML dokumenta:
 - po korisniku
 - po uređaju
 - po mediju
- Objavljivanje:
 - XML dokumenta
 - transformacija nakon prijenosa na uređaj koji služi za prikaz sadržaja
 - dokumenta spremnog za prikaz
 - transformacija prije objave dokumenta ili prilikom dohvata (npr. transformacija XML → HTML+CSS)



Učitavanje XML dokumenta u aplikaciju



- Parsiranje dokumenta
 - provjera dobre oblikovanosti dokumenta (obvezno)
 - provjera valjanosti dokumenta
 - validirajući parseri
 - dokument mora zadovoljiti prvu ili obje provjere
 - dvije vrste parsiranja:
 - linijsko
 - potpuno
- Stvaranje unutarnje reprezentacije dokumenta
 - linijsko parsiranje (SAX) – ne stvara reprezentaciju
 - potpuno parsiranje (DOM) – objektni model dokumenta

Uređivanje dokumenta XML iz aplikacije



- Objektni model:
 - uređivanje sadržaja i strukture
 - stvaranje modela novog dokumenta
 - izvoz modela u formatu XML (datoteka, tok, ...)
- Bez modela
 - stvaranje XML datoteke ispisivanjem teksta u datoteku korak po korak, npr.:
 - korištenjem funkcije `fprintf()`



Prostori imena

Problem konflikta imena elemenata



- Imena nastaju neovisno, npr.
 - elNdeks i potvrda o liječničkom pregledu
- Problem kod spajanja
 - sadržaji iz neovisnih izvora
 - za opis različitih koncepata mogu se koristiti elementi jednakih imena, a različite semantike, sadržaja, strukture ...
- Potrebno dodatno označiti
 - o kojem se točno konceptu (elementu) radi

```
<eindex version='1.0'>
  <student>
    <id>0036435453</id>
    <firstName>Marko</firstName>
    <status>A</status>
  </student>
  <course>
    <title>OpenComputing</title>
    <semester>6</semester>
    <exam>
      <date>2008-07-04</date>
      <grade>C</grade> ...
    </exam>
  </eindex>
```

```
<UMed>
  <exam>
    <id>A65ZGB00456345F</id>
    <date>
      <year>2008</year>
      <month>02</month>
      <day>17</day>
    </date>
    <status>45</status>..
  </exam>
</UMed>
```

Prostori imena

- Rješenje na razini sintakse:
 - korištenje prefiksa za razlikovanje različitih elemenata jednakog imena
 $\text{<p:id/>} \neq \text{<q:id/>}$
 - **p:id** i **q:id** su valjana imena elemenata, tretirana od strane parsera kao i ostala valjana imena elemenata
- Rješenje na logičkoj razini:
 - **p** i **q** su dva različita prostora imena (*namespaces*);
 - svi elementi s istim prefiksom pripadaju zasebnom prostoru imena (**p:id** pripada prostoru imena **p**)
- Prefiksi se definiraju za svaki dokument zasebno
 - globalna definicija prefiksa ne bi riješila problem !

Jedinstveni identifikator prostora imena



- URI kao globalno jedinstveni identifikator resursa, u ovom slučaju prostora imena

<http://www.w3.org/TR/html4/>

<http://www.w3.org/1999/XSL/Transform>

<http://www.fer.hr/eindex/ver1.0>

- temelji se na hijerarhijskoj podjeli odgovornosti imenovanja domena i upravljanju resursima npr. ICANN*-CARNet-FER)
- stvaran resurs reprezentiran URI-jem ne mora postojati

*Internet Corporation for Assigned Names and Numbers

Deklariranje prostora imena

- Povezivanje prefiksa (lokalna informacija) i URI-ja (globalna informacija) atributom **xmlns:**

```
xmlns:namespace-prefix="namespaceURI"
<ex:eindex xmlns:ex="http://www.fer.hr/eindex/ver1.0">
...
</ex:eindex>
```

- Doseg prostora imena je element njegove deklaracije uključujući sadržaj

- u primjeru, dosega **ex** je element **eindex**

- Podrazumijevani (eng. default) prostor imena:

```
xmlns="namespaceURI"
<eindex xmlns="http://www.fer.hr/eindex/ver1.0">
...
</eindex>
```

Primjer korištenja prostora imena



```
<ex:eindex xmlns:ex="http://www.fer.hr/eindex/ver1.0" ex:version='1.0'>
  <ex:student>
    <ex:studentID>0036435453</ex:id>
    <ex:firstName>Marko</ex:firstName> ...
  </ex:student>
  <ex:courses>
    <ex:course>
      <ex:title>OpenComputing</ex:title> ...
    <ex:exam>
      <ex:date>2008-07-04</ex:date>
      <ex:grade>C</ex:grade> ...
    </ex:exam>
  </ex:course>
  <ex:attachments>
    <med:UMed xmlns:med="http://www.hzzo-net.hr/hrmedSchema/2008-01">
      <med:exam>
        <med:id>A65ZGB00456345F</med:id>
        <med:date>
          <med:year>2008</med:year>
          <med:month>02</med:month>
          <med:day>17</med:day>
        </med:date>
        <med:status>45F</med:status> ...
      </med:exam>
    </med:UMed> ...
  </ex:attachments>
</ex:eindex>
```


Literatura

- <http://www.w3.org/XML/>
- <http://www.w3schools.com/>
- <http://www.itwriting.com/xmlintro.php>



Pitanja?

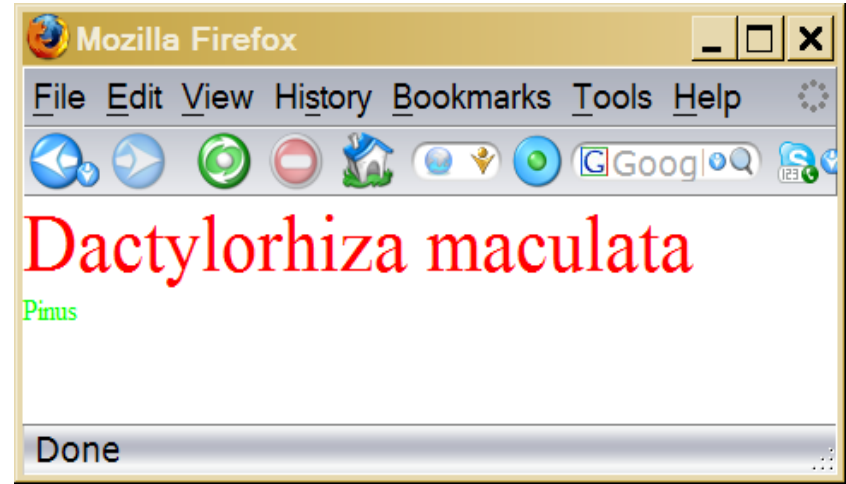
Prilozi uz XML:

Primjer XML - CSS:



```
<?xml version='1.0' encoding='UTF-8'?>
<?xml-stylesheet type="text/css"
  href="test1.css"?>
<BILJKA>
  <ORHIDEJA>
    Dactylorhiza maculata
  </ORHIDEJA>
  <BOR>
    Pinus
  </BOR>
</BILJKA>
```

test1.xml



```
ORHIDEJA{display:block; font-size:
20pt; color:#FF0000;}
BOR{display:block; color:#00FF00;}
```

test1.css

