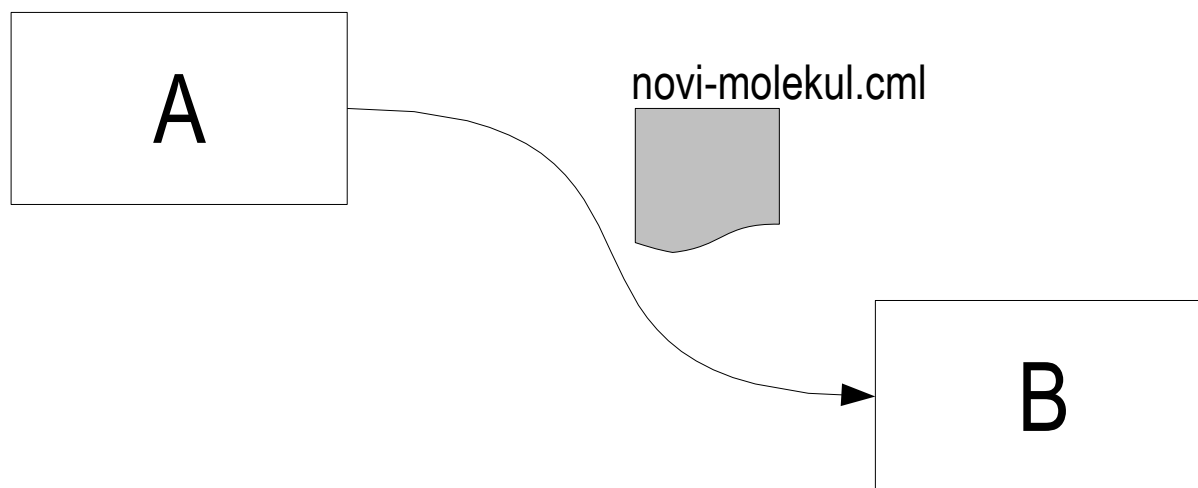


Document Type Definition (DTD)

XML za razmenu podataka

- scenario: dva ili više učesnika komuniciraju putem XML dokumenata
 - XML dokumenti su i human-readable i machine-readable
 - softver za rad sa XML dokumentima postoji na svim računarskim platformama (PC/Mac, Windows/Linux/Unix/MacOS)
 - primer: učesnik A šalje učesniku B opis novog molekula
 - opis je dat u XML obliku, po CML standardu
 - nije važno koji konkretan softver koriste A i B, sve dok mogu da čitaju CML



XML za razmenu podataka

- primer 2: svi učesnici koji razmenjuju vektorske slike koriste SVG (Scalable Vector Graphics) format
- učesnici mogu da koriste različit softver
 - Adobe Illustrator
 - Inkscape
 - CorelDRAW
- ukoliko softver potpuno podržava SVG format, prilikom razmene ne dolazi do gubitaka

XML za razmenu podataka

- za uspešnu razmenu podataka potrebno je da svi učesnici koriste isti format
- ako koriste XML, format može da se definiše u posebnom dokumentu
 - dokument koristi posebnu sintaksu
 - može se publikovati svim zainteresovanim učesnicima
 - predstavlja specifikaciju poruka koje se koriste u komunikaciji

Document Type Definition (DTD)

- deo osnovnog XML standarda
- DTD fajl (dokument) opisuje format klase/familije/tipa XML dokumenata
 - koji elementi i entiteti se mogu pojaviti na kom mestu u dokumentu
 - šta je sadržaj elemenata i atributa

Document Type Definition (DTD)

- primer 1: CML (Chemical Markup Language)

```
<!-- =====-->
<!-- ELEMENTS for chemical and crystallographic concepts -->
<!-- =====-->
<!-- NOTE
  for elements which have element-specific values for the
  builtin attribute, those values are already listed as
  entities
-->
<!-- =====-->
<![%fullCML;[
<!ENTITY %molecule.content 'ANY'>
]]>
<!ENTITY %molecule.content '(atomArray, bondArray)'>
<!ELEMENT molecule      %molecule.content;>
<!ATTLIST molecule
          %tit_id_conv_dict;
          %count;
>
<!ELEMENT formula      ANY>
<!ATTLIST formula
          %tit_id_conv_dict;
          %count;
>
```

Document Type Definition (DTD)

- primer 2: SVG (Scalable Vector Graphics)

```
<!-- Shape
      path, rect, circle, line, ellipse, polyline, polygon
      This module declares markup to provide support for graphical
      shapes.
-->
<!-- a list of points -->
<!ENTITY % Points.datatype "CDATA" >

<!-- Qualified Names (Default) ..... -->
<!ENTITY % SVG.path.qname "path" >
<!ENTITY % SVG.rect.qname "rect" >
<!ENTITY % SVG.circle.qname "circle" >
<!ENTITY % SVG.line.qname "line" >
<!ENTITY % SVG.ellipse.qname "ellipse" >
<!ENTITY % SVG.polyline.qname "polyline" >
<!ENTITY % SVG.polygon.qname "polygon" >
...
```

Primer DTD-a

- primer XML dokumenta

```
<person>
  <name>
    <first_name>Alan</first_name>
    <last_name>Turing</last_name>
  </name>
  <profession>computer scientist</profession>
  <profession>mathematician</profession>
  <profession>cryptographer</profession>
</person>
```

- korenski element je person, i sadrži četiri podelementa
 - element name, koji sadrži dva podelementa
 - first_name
 - last_name
 - tri elementa profession

Primer DTD-a

- DTD koji odgovara prethodnom dokumentu


```
<!ELEMENT person      (name, profession*)>
<!ELEMENT name        (first_name, last_name)>
<!ELEMENT first_name  (#PCDATA)>
<!ELEMENT last_name   (#PCDATA)>
<!ELEMENT profession  (#PCDATA)>
```

 - element `person` sadrži podelemente `name` (tačno jednom) i `profession` (nula ili više puta)
 - element `name` sadrži podelemente `first_name` i `last_name`
 - element `first_name` sadrži tekst
 - element `last_name` sadrži tekst
 - element `profession` sadrži tekst
- dtd se obično čuva kao poseban fajl, npr. `person.dtd`

Primer DTD-a

- prethodni DTD je definisao klasu *validnih* dokumenata
- dokument je validan ako odgovara svom DTD-u
- primer validnog dokumenta

```
<person>
  <name>
    <first_name>Alan</first_name>
    <last_name>Turing</last_name>
  </name>
</person>
```



element profession
nije obavezan

Primer DTD-a

- primer invalidnog dokumenta

```
<person>  
  <profession>computer scientist</profession>  
  <profession>mathematician</profession>  
  <profession>cryptographer</profession>  
</person>
```

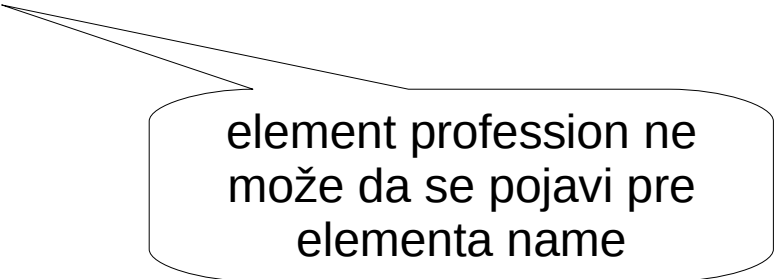


element name je
obavezan

Primer DTD-a

- primer invalidnog dokumenta

```
<person>
  <profession>computer scientist</profession>
  <name>
    <first_name>Alan</first_name>
    <last_name>Turing</last_name>
  </name>
  <profession>mathematician</profession>
  <profession>cryptographer</profession>
</person>
```

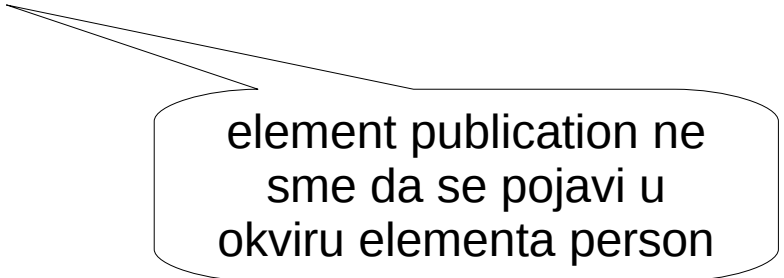


element profession ne
može da se pojavi pre
elementa name

Primer DTD-a

- primer invalidnog dokumenta

```
<person>
  <name>
    <first_name>Alan</first_name>
    <last_name>Turing</last_name>
  </name>
  <profession>mathematician</profession>
  <profession>cryptographer</profession>
  <publication>On Computable
    Numbers...</publication>
</person>
```



element publication ne
sme da se pojavi u
okviru elementa person

Primer DTD-a

- primer invalidnog dokumenta

```
<person>
  <name>
    <first_name>Alan</first_name>
    <last_name>Turing</last_name>
  </name>
  was a <profession>computer scientist</profession>,
  a <profession>mathematician</profession>, and a
  <profession>cryptographer</profession>
</person>
```



u okviru elementa name
mešaju se podelementi i tekst

Vrste deklaracija u DTD-u

- deklaracija elementa
- deklaracija atributa
- deklaracija entiteta
- deklaracija notacija
- ...

Deklaracija elementa

- opšti oblik
`<!ELEMENT naziv (specifikacija_sadržaja)>`
- naziv elementa mora da odgovara pravilima
 - nema razmaka; počinje slovom ili donjom crtom; uključuje slova, cifre, donju crtu, crticu i tačku

Specifikacija sadržaja elementa

- #PCDATA (parsed character data)
 - sadržaj elementa je tekst, bez podelemenata
 - tekst je parsiran, tj. reference na entitete su razrešene
 - primer
`<!ELEMENT phone_number (#PCDATA)>`

Specifikacija sadržaja elementa

- sekvenca podelemenata
 - podelementi se razdvajaju zarezmom
 - uz naziv podelementa navodi se oznaka broja ponavljanja
 - redosled podelemenata je bitan
 - broj ponavljanja podelemenata je bitan

Specifikacija sadržaja elementa

- sekvenca podelemenata
 - primer
`<!ELEMENT name (first_name, last_name)>`
 - validan XML:
`<name>
 <first_name>...</first_name>
 <last_name>...</last_name>
</name>`
 - invalidan XML:
`<name>
 <last_name>...</last_name>
 <first_name>...</first_name>
</name>`

`<name><first_name>...</first_name></name>`

Specifikacija sadržaja elementa

- sekvenca podelemenata
 - broj ponavljanja podelemenata izražava se posebnim znacima:
 - ? : element se pojavljuje jednom ili nijednom ($0..1$)
 - * : element se pojavljuje nijednom, jednom, ili više puta ($0..\infty$)
 - + : element se pojavljuje jednom ili više puta ($1..\infty$)
 - ako se oznaka ne navede, podrazumeva se tačno jednom ($1..1$)

Specifikacija sadržaja elementa

- sekvenca podelemenata
 - primer

```
<!ELEMENT name (first_name, middle_name?, last_name?)>
```
 - validni elementi:

```
<name>  
  <first_name>Madonna</first_name>  
  <last_name>Ciccone</last_name>  
</name>
```

```
<name>  
  <first_name>Madonna</first_name>  
  <middle_name>Louise</middle_name>  
  <last_name>Ciccone</last_name>  
</name>
```

```
<name>  
  <first_name>Madonna</first_name>  
</name>
```

Specifikacija sadržaja elementa

- sekvenca podelemenata

- primer

- ```
<!ELEMENT name (first_name, middle_name?, last_name?)>
```

- invalidni elementi:

- ```
<name>
```

- ```
 <last_name>Ciccone</last_name>
```

- ```
  <first_name>Madonna</first_name>
```

- ```
</name>
```

redosled nije  
ispravan

- ```
<name>
```

- ```
 <first_name>Madonna</first_name>
```

- ```
  <middle_name>Louise</middle_name>
```

- ```
 <middle_name>Veronica</middle_name>
```

- ```
  <last_name>Ciccone</last_name>
```

- ```
</name>
```

middle\_name ne  
može da se  
ponavlja više  
puta

# Specifikacija sadržaja elementa

---

- izbor (jednog od navedenih podelemenata)
  - podelementi se razdvajaju uspravnom crtom
  - izbor može obuhvatati dva ili više podelemenata

# Specifikacija sadržaja elementa

---

- izbor

- primer

- <!ELEMENT methodResponse (params | fault)>

- validan XML

- <methodResponse><params>...</params></methodResponse>

- <methodResponse><fault>...</fault></methodResponse>

- invalidan XML

- <methodResponse>

- <params>...</params>

- <fault>...</fault>

- </methodResponse>

- <methodResponse></methodResponse>



# Specifikacija sadržaja elementa

---

- upotreba zagrada
  - sekvence, izbori i sufiksi se mogu kombinovati
  - sekvenca ili izbor može se staviti unutar zagrada ( . . . )
  - na zagradu se može dodati sufiks \*, +, ?
  - zagrada se može koristiti kao podelement u sekvencama i izborima

# Specifikacija sadržaja elementa

- upotreba zagrada

- primer 1

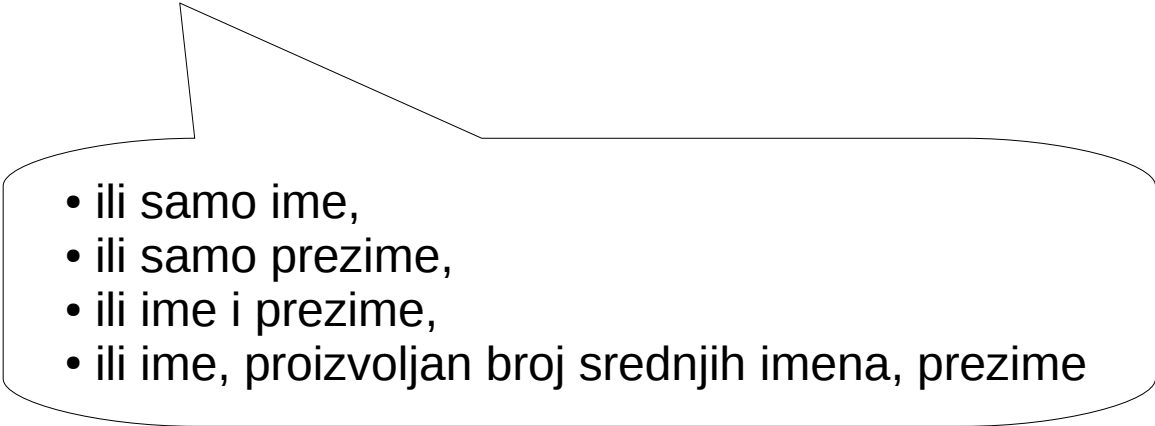
```
<!ELEMENT circle (center, (radius | diameter))>
<!ELEMENT center ((x, y) | (r, t))>
```

- primer 2

```
<!ELEMENT center ((x,y)|(y,x)|(r,t)|(t,r))>
```

- primer 3

```
<!ELEMENT name (last_name |
(first_name, ((middle_name+, last_name) |
(last_name?)))>
```

- 
- ili samo ime,
  - ili samo prezime,
  - ili ime i prezime,
  - ili ime, proizvoljan broj srednjih imena, prezime

# Specifikacija sadržaja elementa

---

- mešani sadržaj
  - element sadrži tekst i podelemente
  - definiše se kao izbor čiji prvi element je #PCDATA, ostali elementi su mogući podelementi, a cela grupa ima broj ponavljanja \*
  - primer  
`<!ELEMENT definition (#PCDATA | term)*>`
  - validan XML  
`<definition>The <term>Turing Machine</term> is an abstract finite state automaton with infinite memory that can be proven equivalent to any other finite state automaton with arbitrarily large memory. Thus what is true for a Turing machine is true for all equivalent machines no matter how implemented.</definition>`

# Specifikacija sadržaja elementa

---

- mešani sadržaj
  - nije moguće navesti broj ponavljanja podelemenata
  - nije moguće navesti da tekst mora biti samo ispred ili iza podelementa
  - deklaracija mešanog sadržaja se ne može koristiti za dalje grupisanje
  - nije moguće fiksirati redosled podelemenata

# Specifikacija sadržaja elementa

---

- prazan sadržaj
  - prazan sadržaj se označava ključnom reči EMPTY
  - primer  
`<!ELEMENT img EMPTY>`
  - validan XML  
``

# Specifikacija sadržaja elementa

---

- bilo kakav sadržaj
  - označava se ključnom reči ANY
  - uključuje tekst, podelemente, isti element ponovljen rekurzivno
  - nalik mešanom sadržaju bez fiksirane liste podelemenata
  - podelementi koji se pojavljuju u sadržaju moraju biti deklarirani
  - u praksi se koristi samo u toku razvoja DTD-a
  - primer  
`<!ELEMENT page ANY>`

# Deklaracija atributa

---

- opšti oblik  
`<!ATTLIST imeElementa imeAtr tipAtr default>`
- naziv atributa mora da odgovara pravilima za formiranje imena
- tip atributa se bira iz konačnog skupa
- obaveznost atributa se bira iz konačnog skupa
- jedna deklaracija može da obuhvati više atributa jednog elementa
  - u praksi se svi atributi jednog elementa stavljaju u jednu deklaraciju
  - u praksi se deklaracija atributa navodi odmah ispod deklaracije elementa

# Specifikacija sadržaja atributa

---

- primer

```
<!ELEMENT img EMPTY>
```

```
<!ATTLIST img
```

```
 src CDATA #REQUIRED
```

```
 width CDATA #REQUIRED
```

```
 height CDATA #REQUIRED
```

```
 alt CDATA #IMPLIED
```

```
>
```

- validan XML

```

```



# Specifikacija sadržaja atributa

---

- mogući tipovi atributa
  - CDATA : bilo koji dobro formirani tekst
  - NMTOKEN : XML *name token* je (slično XML imenima) tekst koji nema razmake, sastavljen je od slova, cifara, znakova `_`, `-`, `:`, `.` i može početi bilo kojim od dozvoljenih znakova
  - NMTOKENS : jedan ili više NMTOKEN-a razdvojenih razmacima
  - *nabrajanje mogućih vrednosti*: navodi se lista mogućih vrednosti atributa međusobno razdvojenih uspravnim crtom, slično kao kod definicije izbora. Ukupna lista se stavlja u zagrade.
  - ID : string koji je XML ime i jedinstven je u okviru celog dokumenta, odnosno nijedan drugi atribut tipa ID ne može imati tu vrednost. Element može imati najviše jedan atribut tipa ID.

# Specifikacija sadržaja atributa

---

- mogući tipovi atributa
  - IDREF : postojeća vrednost ID atributa nekog elementa u istom dokumentu
  - IDREFS : niz IDREF vrednosti razdvojenih razmakom
  - ENTITY : ime neparsiranog entiteta koji je deklarisan u DTD-u (*kasnije*)
  - ENTITIES : više ENTITY vrednosti razdvojenih razmakom
  - NOTATION : sadrži ime notacije koja je deklarisana u DTD-u (*kasnije*)

# Specifikacija sadržaja atributa

---

- primer upotrebe različitih tipova atributa
  - tim inženjera radi na više projekata; jedan projekat uključuje više inženjera, a jedan inženjer može da radi na više projekata
  - za projekte i inženjere uvodimo identifikatore - po jedan atribut tipa ID
  - za povezivanje inženjera sa projektima uvodimo atribut tipa IDREFS
  - za povezivanje projekta sa inženjerima uvodimo atribut tipa IDREFS

```
<!ELEMENT project (name)>
```

```
<!ELEMENT engineer (firstName, lastName)>
```

```
<!ATTLIST engineer
```

```
 engineerID ID #REQUIRED
```

```
 assignments IDREFS #REQUIRED>
```

```
<!ATTLIST project
```

```
 projectID ID #REQUIRED
```

```
 team IDREFS #REQUIRED>
```

# Specifikacija sadržaja atributa

---

- primer upotrebe različitih tipova atributa
  - definisanje dana u nedelji  
`<!ATTLIST date  
dayOfWeek (mon|tue|wed|thu|fri|sat|sun)>`
  - definisanje godišnjih doba  
`<!ATTLIST date season (spring|summer|autumn|winter)>`

# Specifikacija sadržaja atributa

---

- obaveznost pojavljivanja atributa
  - #IMPLIED : atribut nije obavezan; može se navesti ali i ne mora
  - #REQUIRED : atribut je obavezan; mora se navesti
  - #FIXED "value" : atribut nije obavezan, ali se smatra da uvek postoji u elementu i da ima datu fiksnu vrednost *value*; ako se eksplicitno navede u elementu, mora imati baš tu vrednost
  - "default" : atribut, ako se ne navede, ima podrazumevanu vrednost *default* datu pod navodnicima

# Specifikacija sadržaja atributa

---

- obaveznost pojavljivanja atributa - primeri
  - datum rođenja osobe je obavezan, a datum smrti nije  

```
<!ATTLIST person
 born CDATA #REQUIRED
 died CDATA #IMPLIED>
```
  - element biography ima atribut xmlns:xlink i njegova vrednost je uvek <http://www.w3.org/1999/xlink>, bez obzira da li je atribut naveden  

```
<!ATTLIST biography xmlns:xlink CDATA #FIXED
 "http://www.w3.org/1999/xlink">
```

# Deklaracija entiteta

---

- osim predefinisanih entiteta (&lt; &gt; &amp; &apos; &quot;) moguće je deklarirati nove
- opšti oblik deklaracije je  
`<!ENTITY naziv "sadržaj">`
- primer  
`<!ENTITY i18n "internationalization">`
  - referenca &i18n; koja se javlja u dokumentu po ovom DTD-u biće zamenjena sadržajem `internationalization`
  - zamena se vrši prilikom parsiranja (učitavanja) XML dokumenta
- sadržaj entiteta može da sadrži druge entitete
  - samo-referenciranje i cirkularno referenciranje nisu dozvoljeni

# Deklaracija entiteta

---

- eksterni entitet = smešten u posebnom fajlu, izvan DTD-a
  - zgodno kod velikih sadržaja
- opšti oblik deklaracije eksternog entiteta  
`<!ENTITY naziv SYSTEM "adresa_fajla">`
- primeri  
`<!ENTITY footer SYSTEM  
                  "http://www.ns.ac.yu/parts/footer.xml">`  
`<!ENTITY footer SYSTEM "parts/footer.xml">`



# Deklaracija entiteta

---

- eksterni entitet može imati i tzv. javni identifikator:  
`<!ENTITY name PUBLIC "pubid" SYSTEM "sysid">`
- javni identifikator može da identifikuje dobro poznati resurs (iz liste takvih) koji je javno dostupan
- liste javno dostupnih resursa su slabo razvijene
- PUBLIC identifikatori se retko koriste

# Deklaracija entiteta

---

- parametarski entiteti = entiteti koji se definišu za potrebe DTD-a, a ne XML dokumenta
- opšti oblik deklaracije

```
<!ENTITY % ime "vrednost">
<!ENTITY % ime SYSTEM "adresa">
```
- primer (iz SVG DTD-a)

```
<!ENTITY % descTitle
 "((desc,title?)|(title,desc?))?">
<!ENTITY % geExt "">
<!ENTITY % circleExt "">
<!ELEMENT circle (%descTitle;,
 (animate|set|animateMotion|animateColor|
 animateTransform %geExt;%circleExt;)*>
```

# Deklaracija entiteta

---

- neparsirani entiteti = fajlovi koji sadrže ne-XML podatke
- opšti oblik deklaracije

```
<!ENTITY naziv PUBLIC "pubid" notacija>
<!ENTITY naziv SYSTEM "sysid" notacija>
```
- referenca na neparsirani entitet može se naći samo kao vrednost atributa tipa ENTITY
- notacija predstavlja dodatnu informaciju za XML aplikaciju kako da rukuje podacima iz neparsiranog entiteta
- primer

```
<!ENTITY turing_getting_off_bus SYSTEM
 "http://www.turing.org.uk/turing/pi1/bus.jpg"
 jpeg>
```

# Deklaracija notacije

---

- notacija = pomoćni podaci za XML aplikaciju prilikom rukovanja sa neparsiranim entitetima
- opšti oblik deklaracije

```
<!NOTATION naziv PUBLIC "pubid">
<!NOTATION naziv SYSTEM "sysid">
```
- primer

```
<!NOTATION jpeg PUBLIC "image/jpeg">
<!NOTATION jpeg SYSTEM
 "C:/Program Files/imageViewer/viewer.exe">
```
- nema standarda niti preporuke kako treba da izgledaju notacije, sve je stvar aplikacije

# Primer složenijeg DTD-a

- primer XML dokumenta

```
<CHAPTER id="c12345">
<TITLE>Markup: XML and Related Technologies</TITLE>
<AU><NAME>William E. Kasdorf</NAME>
<AFF><ROLE>President</ROLE><ORG>Impressions Book and
Journal Services, Inc.</ORG></AFF></AU>
<INTRO>Markup enables the various parts and features of a
given set of content to be distinguished and named...</INTRO>
<SECT level="1"><HEAD>Overview</HEAD>
<SECT level="2"><HEAD>What is Markup?</HEAD>
<INDXENT>Markup<SUBENT>editorial</SUBENT></INDXENT>
<KW>markup</KW><KW>tags</KW><KW>codes</KW><KW>coding</KW>
<PARA>At the most basic level, markup can be thought of as the
tags and codes embedded in a given set of content that
delineate and describe the component parts of that content.
An editor marks up a <LINKXMPL>paper manuscript</LINKXMPL>...
</PARA></SECT></SECT></CHAPTER>
```

# Primer složenijeg DTD-a

- DTD za prethodni XML dokument

```
<!ELEMENT CHAPTER (TITLE, AU+, INTRO, SECT)>
<!ATTLIST CHAPTER id ID #REQUIRED>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT AU (NAME, AFF)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT AFF (ROLE, ORG)>
<!ELEMENT ROLE (#PCDATA)>
<!ELEMENT ORG (#PCDATA)>
<!ENTITY % emph.grp "IT | B | BI | CSC">
<!ELEMENT INTRO (#PCDATA | %emph.grp;)*>
<!ELEMENT SECT (HEAD, (PARA | SECT)+)>
<!ATTLIST SECT level (1|2|3|4) #REQUIRED>
<!ELEMENT HEAD (#PCDATA|IT)*>
<!ELEMENT PARA (#PCDATA|LINKXMPL|%emph.grp;)*>
<!ELEMENT LINKXMPL (#PCDATA)>
<!ELEMENT IT (#PCDATA)>
<!ELEMENT B (#PCDATA)>
<!ELEMENT BI (#PCDATA)>
<!ELEMENT CSC (#PCDATA)>
```

# Povezivanje XML dokumenta sa DTD-om

---

- tzv. prolog XML dokumenta sadrži
  - XML deklaraciju
  - deklaraciju tipa dokumenta koja povezuje dokument sa DTD-om
- opšti oblik deklaracije tipa dokumenta

```
<!DOCTYPE koren (PUBLIC "pubid" | SYSTEM) "sysid" [
 <!-- interne deklaracije -->]>
```
- primer (za HTML DTD)

```
<!DOCTYPE html PUBLIC
 "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

# Povezivanje XML dokumenta sa DTD-om

---

- primer sa dodatnom internom deklaracijom

```
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"
[
 <!ENTITY slika "images/mypic.gif" GIF>
]>
```



# Povezivanje DTD-ova

---

- uslovne sekcije INCLUDE i IGNORE
  - sekcija INCLUDE uključuje deo DTD-a
  - sekcija IGNORE isključuje deo DTD-a
  - opšti oblik

```
<![IGNORE [DTD section]]>
```

```
<![INCLUDE [DTD section]]>
```
- primer

```
<!ENTITY % draft 'INCLUDE' >
```

```
<!ENTITY % final 'IGNORE' >
```

```
<![%draft;[
```

```
<!ELEMENT book (comments*, title, body, supplements?)>]]>
```

```
<![%final;[
```

```
<!ELEMENT book (title, body, supplements?)>]]>
```