

XML Schema jezici za opis strukture XML dokumenta

Šoštarić, Marko

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Humanities and Social Sciences / Sveučilište Josipa Jurja Strossmayera u Osijeku, Filozofski fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:142:308230>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-02-28**



FILOZOFSKI FAKULTET
SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

Repository / Repozitorij:

[FFOS-repository - Repository of the Faculty of Humanities and Social Sciences Osijek](#)



Sveučilište J.J. Strossmayera u Osijeku
Filozofski fakultet
Preddiplomski studij informatologije

Marko Šoštarić

XML Schema jezici za opis strukture XML dokumenta

Završni rad

Mentor izv. prof. dr. sc. Boris Bosančić

Osijek, 2019.

Sveučilište J.J. Strossmayera u Osijeku
Filozofski fakultet Osijek
Odsjek za informacijske znanosti
Preddiplomski studij informatologije

Marko Šoštarić

XML Schema jezici za opis strukture XML dokumenta

Završni rad

Društvene znanosti, informacijske i komunikacijske znanosti, informacijski
sustavi i informatologija

Mentor izv. prof. dr. sc. Boris Bosančić

Osijek, 2019.

Prilog: Izjava o akademskoj čestitosti i o suglasnosti za javno objavljivanje

Obveza je studenta da donju Izjavu vlastoručno potpiše i umetne kao treću stranicu završnog odnosno diplomskog rada.

IZJAVA

Izjavljujem s punom materijalnom i moralnom odgovornošću da sam ovaj rad samostalno napravio te da u njemu nema kopiranih ili prepisanih dijelova teksta tuđih radova, a da nisu označeni kao citati s napisanim izvorom odakle su preneseni.

Svojim vlastoručnim potpisom potvrđujem da sam suglasan da Filozofski fakultet Osijek trajno pohrani i javno objavi ovaj moj rad u internetskoj bazi završnih i diplomskih radova knjižnice Filozofskog fakulteta Osijek, knjižnice Sveučilišta Josipa Jurja Strossmayera u Osijeku i Nacionalne i sveučilišne knjižnice u Zagrebu.

U Osijeku, datum

Marko Šestarić, 0122223744
ime i prezime studenta, JMBAG

Sadržaj

Sažetak	6
1. UVOD	7
2. XML OZNAČITELJSKI JEZIK	8
3. XML SCHEMA JEZICI.....	13
3.1. DTD	13
3.2. W3C XML Schema	17
3.3. RELAX NG	20
3.4. Schematron	21
4. Zaključak	22
Literatura	23

Sažetak

Svrha rada ogleda se u osvrtu na XML schema jezike koji se rabe za opis strukture XML dokumenta. U skladu sa svrhom ciljevi rada odnose se na prikaz DTD-ja i W3C XML Scheme, kao danas najupotrebljavanijih XML schema jezika, a zatim i ostalih, manje poznatih, ali također važnih XML schema jezika, kao što su RELAX NG i Schematron. U prvom dijelu rada govori se općenito o XML označiteljskom jeziku, povodu njegova nastanka, sintaksi i semantici. Zatim se pozornost posvećuje pojedinačnom osvrtu na svaku od navedenih XML schema, izlaže se način njihova funkcioniranja, opis njihove sintakse te navode prednosti i nedostaci. Ujedno, iznose se i primjeri kako bi se lakše razumjela svrha navedenih XML schema jezika. Pritom, poseban naglasak stavljen je na usporedbu dvije najkorištenije XML sheme danas u mrežnom okruženju – DTD-ja i W3C XML Scheme.

Ključne riječi: XML, XML schema jezici, DTD, W3C XML Schema.

1. UVOD

XML schema jezici opisuju strukturu XML dokumenta na način definiranja elemenata i atributa koji se koriste u XML dokumentu, kao i na određenim ograničenjima u pogledu njihove upotrebe. XML schema jezici uglavnom se fokusiraju na definiranje validacijskih pravila vezanih uz sintaksu i semantiku prije nego na modeliranje klasa XML dokumenta. Svi XML schema jezici definiraju *transformacije* koje se primjenjuju na instancama klasa odnosno tipova XML dokumenata. Transformacije zatim koriste te instance kao ulazne jedinice za generiranje validacijskih izvješća koja utvrđuju valjanost svake pojedine instance prema definiranim transformacijama. Jezici koji su razvijeni specifično za izražavanje takvih transformacija i koji propisuju strukturu XML dokumenta nazivaju se XML schema jezici, a njima pripadaju: DTD, W3C XML Schema, RELAX NG, Schematron i dr. Što se tiče validacije, ona predstavlja proces provjere usklađenosti XML dokumenta s pripadnom XML schemom čime se utvrđuje da je neki XML dokument dobro oformljen. Glavni cilj XML schema jezika je specificirati kakva bi struktura XML dokumenta trebala biti. To se odnosi na određivanje strukture XML dokumenta kao i ograničenja za njegove elemente, attribute i ostale značajke. XML schema jezici mogu se shvatiti i kao gramatika XML dokumenta, jer na neki način definiraju vokabular i pravila uporabe vokabulara koji se može koristiti u dokumentu.

Svrha ovoga rada je prikaz najkorištenijih XML schema jezika za opis strukture XML dokumenta. U skladu sa svrhom rada ciljevi rada odnose se na prikaz DTD-ja, W3C XML Scheme te ostalih, manje poznatih, ali također bitnih XML schema jezika. Prvo se poglavlje odnosi na prikaz XML označiteljskog jezika te opis njegove strukture i svrhe. Drugo je poglavlje posvećeno XML schema označiteljskim jezicima i u njemu će se opisati DTD i, W3C XML Schema kao najčešće upotrebljavani XML schema jezici, te RELAX NG i Schematron. Potonji su XML schema jezici koji nisu u čestoj upotrebi, kao prethodna dva jezika, ali su vrijedni spomena zbog nekih specifičnosti.

2. XML OZNAČITELJSKI JEZIK

XML (*Extensible Markup Language*) je označiteljski jezik (engl. *markup language*) koji definira pravila strukturiranja XML dokumenata u formatu koji je ljudski i strojno čitljiv. XML je zapravo *metajezik*, odnosno sredstvo za formalno opisivanje označiteljskih jezika. Glavni ciljevi kod dizajniranja ovog označiteljskog jezika su naglasiti jednostavnost, općenitost te mogućnost upotrebe u mrežnom okruženju. Tijekom dizajniranja XML-a važno je bilo voditi računa o tome da se prijenos kodiranih XML dokumenata s jednog hardvera i softvera na drugi obavlja bez gubitka informacija. Gledajući sa stajališta sintakse, XML je vrlo sličan HTML-u, no koristi potpuno drugačijoj svrsi. XML opisuje podatke te im daje značenje, dok HTML te podatke prikazuje na mrežnoj stranici. Iako je XML sličan HTML-u, on nikad ne može predstavljati zamjenu za HTML. Postoje i karakteristična svojstva XML-a koji XML-u daju karakteristike po kojima se on razlikuje od drugih označiteljskih jezika. To su:

- XML provodi opisno označavanje teksta, a ne proceduralno.
- XML ne ovisi ni o hardveru ni o softveru.
- XML razlikuje koncepte sintaktičke ispravnosti i semantičke valjanosti s obzirom na pripadnu XML shemu.

U usporedbi s HTML-om, XML ima neke važne karakteristike koje ga razlikuju od HTML-a:

- XML je proširiv, odnosno ne sadrži fiksni skup oznaka.
- XML dokumenti moraju biti dobro oformljeni prema unaprijed definiranim sintaktičkim pravilima.
- XML se usredotočuje na značenje podataka, a ne na njihov prikaz.¹

Kod sustava opisnih označiteljskih jezika, označiteljski kôdovi imaju više funkcija. Označiteljski kôd, kao na primjer `<p>`, označava početak odlomka. Radi usporedbe, proceduralni označiteljski sustav definira koji će se procesi odvititi na kojem dijelu dokumenta, npr. „pomjeri desnu marginu dvije točke u lijevo, pomjeri lijevu marginu dvije točke desno, prijeđi u novi red“ i tako dalje. U XML-u, upute potrebne za procesuiranje i obrađivanje

¹A Gentle Introduction to XML. URL: <https://tei-c.org/release/doc/tei-p5-doc/en/html/SG.html> (2019-9-7)

dokumenta za potrebe njegova prikaza u pregledniku prilično se razlikuju od opisnog označivanja koji se pojavljuje u dokumentu. Te se upute kreiraju izvan dokumenta u odvojenim programima i obično su pohranjene u posebnom dokumentu zvanom *stylesheet*.

```
<?xml version="1.0" encoding="utf-8" ?>
<popis_korisnika>
  <korisnik>
    <ime>Martin</ime>
    <prezime>Žica</prezime>
    <odjel naziv="dječji" />
    <ak_godina>2016/17</ak_godina>
    <e-mail>mzica@ffos.hr</e-mail>
    <godina_studija>&godina;</godina_studija>
  </korisnik>
  <korisnik>
    <ime>Tomo</ime>
    <prezime>Perić</prezime>
    <odjel naziv="odrasli" />
    <ak_godina>2016/17</ak_godina>
    <e-mail>tperic@ffos.hr</e-mail>
    <godina_studija>&godina;</godina_studija>
  </korisnik>
  <korisnik>
    <ime>Marin</ime>
    <prezime>Zdravković</prezime>
    <odjel naziv="studijaska_čitaonica" />
    <ak_godina>2016/17</ak_godina>
    <e-mail>mzdravkovic@ffos.hr</e-mail>
    <godina_studija>&godina;</godina_studija>
  </korisnik>
</popis_korisnika>
```

Slika 1. - Popis korisnika knjižnice u XML-u u datoteci *korisnici.xml*

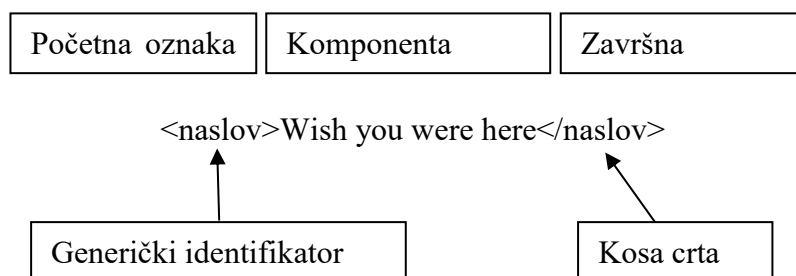
Svaki XML dokument sastoji se od dva dijela, a to su: zaglavlje i sadržaj dokumenta. U zaglavlju XML dokumenta navodi se XML deklaracija, dok sadržaj dokumenta obuhvaća kako tekst, tako i oznake kojima je označen kôd. Prva zbirna oznaka u XML dokumentu, kojom započinje XML deklaracija, odnosi se na prepoznatljiv niz znakova (<?xml) koji upućuje na to da se radi o XML dokumentu. Nadalje, slijedi obavezni atribut *@version* u kojem se navodi verzija XML dokumenta (najčešće se koristi verzija 1.0), a nakon toga slijedi neobavezan atribut *@encoding* u kojem se navodi kodna stranica (najčešće je to utf-8 koja označava da se radi o 8-bitnoj UNICODE kodnoj stranici).

Tehnički termin u XML specifikaciji koji se koristi za označavanje komponenti teksta odnosno objekata sadržaja naziva se „element“. Različiti tipovi elemenata imaju drugačije nazive, no XML ne pruža značenje pojedinih tipova elemenata, nego samo ukazuje na to koji

su tipovi elemenata međusobno povezani. Svatko ima mogućnost kreiranja svog vlastitog skupa elemenata. Pri kreiranju i odabiru elemenata za vlastiti skup treba obratiti pozornost na to da se opisni karakter elementa, odnosno njegovo značenje, može iščitati iz njegova naziva i da njegov sadržaj mora odgovarati njegovu značenju. Na primjer, za komponentu teksta „Drvo“ naziv elementa može biti „Biljka“, ali i „Materijal“. Elementi u XML-u, osim što opisuju sadržaj, daju mu, dakle, i značenje te pomažu pri razumijevanju cjelokupne strukture dokumenta. U našem primjeru, komponenta teksta „Drvo“ zadobit će jedno značenje ako se označi elementom „Biljka“, a sasvim drugo značenje ako se označi elementom „Materijal“. Svi elementi u XML-u sastoje se od početne oznake, odgovarajuće komponente teksta i završne oznake. Kako bi lakše razumjeli sintaksu XML dokumenta navest će se sljedeći primjer:

```
<p>Sutra ću kupiti album  
<naslov>Wish you were here</naslov>  
grupe Pink Floyd</p>
```

Dakle, znak „<“ (manje) predstavlja početak početne oznake elementa nakon koje slijedi slovo („p“) ili riječ („naslov“) koji predstavljaju generički identifikator određenog elementa, a početna oznaka završava znakom „>“ (veće). Generički identifikator (engl. *generic identifier* - GI) predstavlja tehnički termin pomoću kojeg se definira naziv tipa elementa. Oznake „<“ i „>“ u XML-u se koriste kao zagrade u koje se smješta generički identifikator elementa koji ujedno predstavlja sam naziv elementa. Iza početne oznake elementa slijedi odgovarajući tekst (npr. elementu <p> pripada tekst „Sutra ću kupiti album ... grupe Pink Floyd“, a elementu <naslov> pripada tekst „Wish you were here“). Nakon teksta slijedi završna oznaka koja također počinje i završava znakovima „<“ i „>“, no za razliku od početne oznake sadrži još jedan znak, a to je kosa crta „/“ koja se stavlja ispred generičkog identifikatora koja ukazuje na to da se radi o završnoj oznaci XML elementa.



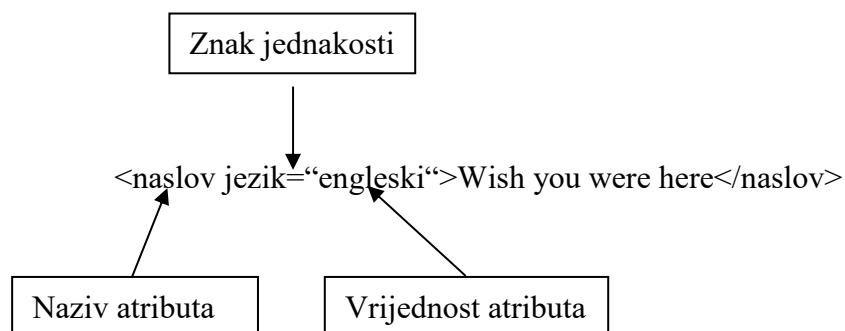
Slika 2. - XML element bez atributa.

Element može sadržavati tekst, attribute, ostale elemente ili kombinaciju svega navedenog. Pri davanju naziva XML elementima potrebno je pridržavati se sljedećih pravila:

- mogu sadržavati slova, brojeve i druge znakove;
- osjetljivi su na velika i mala slova;
- ne mogu započeti znakom zarez ili brojem;
- ne smiju započeti slovima xml (ili Xml ili XML itd.);
- ne smiju sadržavati prazne prostore².

Također se izbjegava upotreba znaka crtice („-“), točke („.“) i dvotočke („:“). U slučaju da se naziv elementa želi kreirati od više riječi, koristi se znak donje crte („_“) koja se stavlja umjesto razmaka među riječima u elementu (npr. žanr_glazbe).

Elementi u XML-u, kao i u HTML-u, mogu imati attribute. Atributi sadrže podatke koji su vezani uz specifični element. Atribut se uvijek navodi samo unutar početne oznake elementa, a vrijednost atributa nalazi se u navodnicima; ispred vrijednosti atributa navodi se ime atributa i znak jednakosti.



Slika 3. - XML element s atributom.

Atributi u odnosu na elemente imaju neke nedostatke:

- atributi ne mogu sadržavati više vrijednosti (elementi mogu);
- atributi ne mogu imati strukturu stabla (elementi mogu);
- atributi su teško proširivi i nisu lako prilagodljivi za buduće izmjene informacija.³

² W3School.com: XML tutorial URL: <https://www.w3schools.com/xml/>

³ W3School.com: XML tutorial URL: <https://www.w3schools.com/xml/>

Zbog navedenih nedostataka atributa savjetuje se korištenje elemenata. Atributi su pogodni za dodavanje dodatnih informacija koje služe programima za parsiranje XML dokumenata kako bi prikazali sadržaj dokumenta na željeni način.

Sintaktička pravila XML-a koja se navode u nastavku vrlo su jednostavna i lako zapamtljiva:

- XML dokumenti mogu imati samo jedan korijenski element (jedan glavni element u koji se ugnježđuju svi drugi elementi).
- Svi elementi u XML dokumentu moraju imati početnu i završnu oznaku.
- XML elementi osjetljivi su na velika i mala slova.
- XML elementi moraju biti pravilno ugnježđeni bez međusobnog preklapanja elemenata.
- Vrijednost atributa mora biti navedena unutar navodnika.

XML dokumenti koji odgovaraju navedenim pravilima smatraju se dobro oformljenim XML dokumentima.

3. XML SCHEMA JEZICI

3.1. DTD

DTD (engl. *Document Type Definition*) je prva i široko upotrebljavana schema označiteljskih jezika koja se može ugraditi u XML dokument. DTD sadrži pravila kojima se definira pojedini označiteljski jezik za konkretnu namjenu, a ta su pravila vrlo sažeta te ih je lako primijeniti.

Tip XML dokumenta također je jedan od ključnih aspekata XML-a. Dokumenti imaju tipove isto kao i ostali objekti koji se procesuiraju putem računala. Tip XML dokumenta definira se svojim sastavnim dijelovima te svojom strukturom. Ako su dokumenti poznatih tipova, posebni program (*XML Parser*) može provjeriti ispunjava li dokument sve potrebne uvjete da bude svrstan u određeni tip dokumenata. Naravno, prije upotrebe takvog programa trebaju se definirati tipovi XML dokumenata čija će se specifikacija ispitivati.⁴ Provjerom XML dokumenata može se utvrditi je li formalno valjan ili nevaljan u odnosu na propisana pravila i strukturu tipa dokumenta kojem pripada.

Svojim razvojem, XML je nadmašio mnoge značajke SGML-a (engl. *Standard Generalized Markup Language*)⁵. Jedno od prvih poboljšanja u razvoju XML-a odnosi se na uvođenje imenskih prostora (engl. *namespace*). Još jedno važno poboljšanje je razvoj W3C XML Scheme koja je dizajnirana kako bi ispravila i nadopunila mnoga ograničenja DTD-ja kao jezika za definiranje strukture XML dokumenta. Ta ograničenja su sljedeća:

- DTD ne podržava imenske prostore;
- sintaksa DTD-ja je prilično drugačija od sintakse XML dokumenta; stoga se ne mogu koristiti standardni XML alati u svrhu upravljanja DTD shemama;
- DTD podržava samo nekoliko ugrađenih tipova blokova sadržaja kao što su PCDATA, CDATA te ne podržava korisnički definirane tipove i ne može ograničiti znakovne podatke;
- mehanizmi za specificiranje referentnog integriteta su slabi;
- definicije elemenata odnose se na cijeli dokument.

⁴ Hedgebeth, D., "Managing structured data with the extensible markup language (XML)", *VINE*// 38, 3(2008), str. 348- 356. URL: <https://doi.org/10.1108/03055720810904853> (2019-9-7)

⁵ SGML je opisni označiteljski jezik objavljen 1986.

Unatoč ograničenjima, DTD je još uvijek u širokoj upotrebi u slučajevima u kojima nisu potrebne napredne značajke koje nudi W3C XML Schema. Unatoč svojoj popularnosti, W3C XML Schema ne može u potpunosti zamijeniti DTD.⁶

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE popis_korisnika[
  <!ELEMENT popis_korisnika (korisnik+)>
  <!ELEMENT korisnik (ime, prezime, odjel, ak_godina?, e-mail*, godina_studija)>
  <!ELEMENT ime (#PCDATA)>
  <!ELEMENT prezime (#PCDATA)>
  <!ELEMENT odjel EMPTY>
  <!ELEMENT ak_godina (#PCDATA)>
  <!ELEMENT e-mail (#PCDATA)>
  <!ELEMENT godina_studija (#PCDATA)>
  <!ATTLIST odjel naziv (dječji|odrasli|studijska_čitaonica) #IMPLIED>
  <!ENTITY godina "1. godina">
]
```

Slika 4. - Primjer DTD deklaracije

Na Slici 4. prikazan je primjer deklaracije DTD-ja na početku XML dokumenta. Svaka deklaracija DTD-ja započinje identifikatorom DOCTYPE sa znakom „<“ i „!“ ispred identifikatora, nakon kojeg slijedi korijenski element (u ovom slučaju „popis_korisnika“) a nakon toga uglate zagrade („[]“) unutar kojih se pojedinačno deklariraju XML elementi. XML elementi deklariraju se pomoću identifikatora odnosno tzv. „rezervirane riječi“ „ELEMENT“ ispred koje se, kao i u slučaju DOCTYPE-a, navode znakovi „<“ i „!“ (<!ELEMENT). Postoje četiri vrste DTD deklaracija za četiri vrste XML elemenata:

- DTD deklaracija elementa koji sadrži druge elemente:

```
<!ELEMENT naziv_elementa (element1, element2, ...)
```

Primjer:

```
<!ELEMENT korisnik (ime, prezime, odjel, ak_godina?, e-mail*, godina_studija)>
```

- DTD deklaracija elementa u kojima se navodi krajnji sadržaj:

```
<!ELEMENT naziv-elementa (#PCDATA)>
```

Primjer:

⁶ Liu, M., "DTD schema: a simple but powerful XML schema language", *International Journal of Web Information Systems.* // 4, 4(2008), str. 465-483. URL: <https://doi.org/10.1108/17440080810919495>(2019-9-7)

```
<!ELEMENT ime (#PCDATA)>
```

- DTD deklaracija elementa koji sadrže druge elemente i u kojima se navodi krajnji sadržaj:

```
<!ELEMENT naziv-elementa (#PCDATA | element1 | element2  
| str)*>
```

Primjer:

```
<!ELEMENT kolegij (#PCDATA | profesor | prostorija)*>
```

- DTD deklaracija praznog elementa:

```
<!ELEMENT naziv-elementa EMPTY>
```

Primjer:

```
<!ELEMENT odjel EMPTY>
```

Pri deklariranju elemenata može se odrediti broj njihovog pojavljivanja u dokumentu (tzv. kardinalnost elemenata). Oznake koje to određuju navode se s desne strane elementa u modelu sadržaja. U slučaju da nije naveden indikator kardinalnosti elementa to znači da se taj element u dokumentu može pojaviti jednom i samo jednom u nadređenom elementu. Ostali indikatori kardinalnosti su znak upitnika („?“), znak plus („+“) i znak zvjezdice („*“). Značenje indikatora kardinalnosti opisano je u Tablici 1.

Tablica 1. - Kardinalnost elemenata.

Znak	Broj pojavljivanja u nadređenom elementu	Primjer
	jednom i samo jednom	<!ELEMENT korisnik (ime, prezime)>
?	nijedno ili jednom	<!ELEMENT kolegij (naziv, prostorija?)>
+	jednom ili više puta	<!ELEMENT profesor (ime, prezime+)>
*	nijednom ili više puta	<!ELEMENT profesor (ime, prezime*)>

DTD atributi deklariraju se prema sljedećoj sintaksi:


```
<!ATTLIST naziv-elementa naziv-atributa tip-atributa deklaracija-  
vrijednosti-atributa>
```

Primjer:

```
<!ATTLIST odjel naziv (dječji|odrasli|studijska_čitaonica) #IMPLIED>
```

DTD deklaracija atributa sastoji se od pet dijelova:

- „rezervirane riječi“ ATTLIST (svi atributi deklariraju se unutar ATTLIST deklaracije);
- naziva elementa kojem atribut pripada;
- naziva atributa;
- oznake za tip atributa (ID, IDREF, IDREFS, NMTOKEN, NMTOKENS, ENTITY, ENTITIES, NOTATION, CDATA) koja određuje na koji način će se procesuirati vrijednost atributa;
- deklaracije vrijednosti atributa (#REQUIRED, #IMPLIED, #FIXED) koja određuje (ne)obveznost i način pojavljivanja vrijednosti atributa u dokumentu.⁷

Entiteti u DTD-ju koriste se za definiranje prečaca do određenih vrijednosti. Entiteti mogu biti definirani unutrašnjom ili vanjskom deklaracijom. U XML dokumentu mogu se pojaviti četiri vrste entiteta:

- referentni entiteti;
- znakovni entiteti;
- opći entiteti;
- parametarski entiteti.

Entiteti deklarirani unutar DTD-ja nazivaju se unutrašnji entiteti (engl. *internal entities*) te se deklariraju po definiranoj sintaksi:

```
<!ENTITY naziv_entiteta "vrijednost entiteta">
```

Primjer:<!ENTITY godina "1. godina">

⁷ w3webtutorial: DTD Attributes. URL: <http://www.w3webtutorial.com/dtd/dtd-attribute.php>

3.2. W3C XML Schema

W3C XML Schema je vrlo moćan, ali i kompleksan mehanizam za opis strukture XML dokumenta. Funkcija W3C XML Scheme ogleda se u sljedećim aspektima:

- **deklaraciji elemenata**, odnosno definiranju svojstava elemenata koja uključuju naziv elementa i njihov imenski prostor, dok je važno svojstvo elementa njegov tip koji ograničava i određuje koje atribute i podređene elemente može imati;
- **deklaraciji atributa**, odnosno definiranju svojstava atributa. Kao i kod deklaracije elemenata ova deklaracija također uključuje naziv i ciljni imenski prostor. Tip atributa ograničava vrijednost koju atribut može poprimiti;
- **jednostavnim i složenim tipovima podataka**;
- **definiciji grupe modela i grupe atributa**, odnosno imenovane skupine elemenata i atributa koji se mogu ponovno upotrijebiti u raznim definicijama tipova.

Za razliku od DTD-ja, W3C XML Schema ne posjeduje ograničenja koja su navedena u prethodnom poglavlju. U W3C XML Schemi osigurana je potpuna podrška za XML imenske prostore i tipove podataka te je omogućen opis različitih modela sadržaja. Sintaksa XML dokumenta i W3C XML Scheme je ista. Za razliku od DTD-ja, W3C XML Schema ne podržava deklaraciju entiteta te se ne može ugraditi u XML dokument kao što je slučaj s DTD-jem⁸. W3C XML Schema podržava mnogo različitih tipova podataka. Osnovni i najčešći tipovi podataka koji se koriste u W3C XML Schemi su:

- nizovi znakova (*string*);
- decimalni broj (*decimal*);
- cijeli broj (*integer*);
- Booleov operator (*boolean*);
- datum (*date*);
- vrijeme (*time*) i dr.

Osim navedenih tipova podataka, u W3C XML Schemi je osigurana podrška i za mnoge druge izvedene tipove podataka (*token*, *int*, *float*, *duration*, *positiveInteger* itd.), a ona također

⁸ Ioannides, D., "XML schema languages: beyond DTD", *Library Hi Tech.* // 18, 1(2000), str. 9-15. URL: <https://doi.org/10.1108/07378830010314366> (2019-9-7)

omogućuje i definiranje vlastitog tipa podataka. Najvećom prednošću smatramo podršku za XML imenske prostore koja je navedena i u preporuci specifikacije W3C XML Scheme. Imenski prostori koriste se za grupiranje elemenata i atributa koji su povezani na specifičan način. Uobičajeno, imenski prostori identificiraju se jedinstvenim URI-jem (engl. *Uniform Resource Identifier*). Imenski prostor se navodi u korijenskom elementu XML dokumenta kao vrijednost atributa `@xmlns`. Pritom, atributu `@xmlns` pridodan je prefiks koji zatim rabe i svi elementi deklariranog imenskog prostora u XML dokumentu. Ovdje je važno naglasiti da se imenski prostor ne mora odnositi na neku stvarnu URL adresu, već da zadovolji funkciju jedinstvenog identifikatora.



Slika 5. - Primjer XML Schema dokumenta.

Na Slici 5. može se vidjeti da se deklaracija XML imenskih prostora provodi u okviru korijenskog elementa W3C XML Schema dokumenta ili XSD dokumenta `<xs:schema>` putem atributa `@xmlns` koji ima prefiks `xs`. Imenski prostor za sve elemente XML Scheme glasi: <http://www.w3.org/2001/XMLSchema>. Drugim riječima, to znači da svi elementi s prefiksom `xs` koji se pojavljuju u XSD dokumentu pripadaju imenskom prostoru ili skupu elemenata W3C XML Scheme.⁹

⁹ w3: Naming and Addressing: URIs, URLs, ... URL: <https://www.w3.org/Addressing/> (2019-9-7)

Nakon atributa `@xmlns` navodi se imenski prostor za deklarirane elemente u XSD dokumentu (`targetNamespace="http://www.w3schools.com"`), a nakon toga navodi se zadani imenski prostor koji ne mora uključivati prefikse (`xmlns="http://www.w3schools.com"`). Na kraju deklaracije navodi se „`elementFormDefault='qualified'`“ koji označava da bilo koji element deklariran ovom W3C XML Schemom mora imati pripadni XML imenski prostor. Nakon deklaracije XML imenskih prostora, i nakon zaglavlja W3C XML Scheme, deklariraju se elementi i atributi koji definiraju strukturu XML dokumenta. Način deklaracije elemenata i atributa bitno se razlikuje od one kod DTD-ja. Pri deklaraciji elemenata i atributa u W3C XML Schemi specificira se naziv novog elementa odnosno atributa te sadržaj koji će im biti pridodan (tzv. dopušteni sadržaj). Kod deklaracije elemenata u W3C XML Schemi razlikujemo dva tipa elemenata: jednostavne i složene tipove elemenata. Jednostavni tipovi elemenata unutar sebe sadrže samo tekst te ne smiju sadržavati druge elemente i atribute. Primjeri deklaracija jednostavnih tipova elemenata dani su u nastavku:

```
<xs:element name="to" type="xs:string"/>
<xs:element name="from" type="xs:string"/>
```

Jednostavni tipovi elemenata mogu imati zadanu (engl. *default*) ili fiksnu (engl. *fixed*) vrijednost. Ako u XML dokumentu nije zadana neka druga vrijednost atributa, onda se primjenjuje zadana vrijednost, a u slučaju fiksne vrijednosti, sve ostale vrijednosti u XML dokumentu ne smiju biti različite od nje. To znači da fiksna vrijednost također ima kontrolni karakter. Za razliku od jednostavnih tipova elemenata, složeni tipovi elemenata definirani su elementom *complexType*. Razlikujemo četiri vrste složenih elemenata:

- prazni elementi - sadržavaju samo atribute i imaju samo početnu oznaku;
- elementi koji sadrže druge elemente koji sadrže tekst - osim drugih elemenata ne smiju imati tekstualni sadržaj;
- elementi koji sadrže samo tekst i atribute - unutar sebe ne smiju sadržavati druge elemente;
- elementi koji sadrže i tekst i druge elemente.

Atributi W3C XML Scheme detaljno opisuju elemente te se njihova vrijednost navodi unutar znakova navodnika. Atribut `@name` opisuje naziv atributa, a dozvoljeni tip definira se atributom `@type`. Što se tiče kardinalnosti podataka, ona je definirana atributima `@minOccurs`

(minimalan broj pojavljivanja elementa u nadređenom elementu) i *@maxOccurs* (maksimalan broj pojavljivanja elementa u nadređenom elementu). Početna vrijednost atributa *@minOccurs* i *@maxOccurs* iznosi '1' te se uzimaju u obzir ako jedan ili oba atributa nisu navedena. Osim brojčane vrijednosti, za vrijednost atributa *@maxOccurs* moguće je i upisati tekst *unbounded* što znači da broj pojavljivanja određenog elementa nije definiran te omogućuje neograničen broj pojavljivanja tog elementa u pripadnom modelu sadržaja.¹⁰ Atributi *@minOccurs* i *@maxOccurs* poznati su i pod nazivom indikatori događaja (engl. *occurrence indicators*).

Već opisani XML schema jezici (DTD i W3C XML Schema) najpopularniji su jezici za opis strukture XML dokumenta. No, postoje i drugi schema jezici koji su također u širokoj upotrebi kao što su: RELAX NG i Schematron.

3.3. RELAX NG

RELAX NG koristi se kao alternativa DTD-ju i W3C XML Schemi za provjeru sintakse i definiranje valjanosti XML dokumenata. RELAX NG rabi istu sintaksu kao i XML, a od W3C XML Scheme se razlikuje po tome što koristi drugačiji skup elemenata za deklariranje elemenata i atributa, a i razlikuje se i po nekim načinima deklariranja. RELAX NG pruža većinu prednosti koje W3C XML Schema ima u odnosu na DTD, k tomu je jednostavniji od W3C XML Scheme. RELAX NG pruža snažnu podršku neuređenom i neorganiziranom sadržaju,¹¹ te dozvoljava atributima da se tretiraju kao elementi u modelima sadržaja. U svrhu objašnjenja sintakse deklariranja elemenata i atributa u RELAX NG-u, u nastavku je dan prikaz kôda RELAX NG schema dokumenta kompaktne sintakse u kojem su deklarirani elementi iz datoteke *korisnici.xml* prikazane na Slici 1.

```
popis_korisnika_o = element popis_korisnika {korisnik_o+}
korisnik_o = element korisnik {ime_o+, prezime_o+, odjel_o+,
ak_godina_o+, e-mail_o+, godina_studija_o+}
ime_o = element ime {text }
prezime_o = element prezime { text }
odjel_o = element odjel { text }
ak_godina_o = element ak_godina { text }
e-mail_o = element e-mail { text }
```

¹⁰ Hong Quang, N. and Rahayu, W., "XML schema design approach", *International Journal of Web Information Systems*. 1, 3(2005), str. 161-178. URL: <https://doi.org/10.1108/17440080580000091>(2019-9-7)

¹¹ relaxng: RELAX NG Compact Syntax Tutorial. URL: <https://relaxng.org/compact-tutorial-20030326.html> (2019-9-7)

```
godina_studija_o = element godina_studija { text }
start = popis_korisnika_o
```

Prikazana kompaktna sintaksa RELAX NG-a počiva na svojevrsnim obrascima koji sadržavaju deklaracije elemenata i atributa iz dokumenta *korisnici.xml*. U primjeru obrazac *popis_korisnika_o* definiran je tako da se sastoji od deklaracije elementa `<popis_korisnika>` uz koji se navodi i pripadni model sadržaja (u vitičastim zagradama) u kojem je navedeno da se obrazac *korisnik_o* u njemu može pojaviti jednom ili više puta. Iz ovog primjera može se vidjeti da RELAX NG podržava oznake kardinalnosti elemenata iz DTD-ja. Atributi u RELAX NG dokumentu kompaktne sintakse deklariraju se pak na sljedeći način:

```
att.naziv = attribute naziv {"dječji" | "odrasli" | "studijska_čitaonica"}
```

Sličnosti u deklaraciji atributa RELAX NG-ja s deklaracijama atributa u DTD-ju odmah se uočavaju. Deklarira se obrazac naziva *att.naziv* koji se odnosi na atribut `@naziv` koji pak može poprimiti jednu od tri navedene vrijednosti (*dječji*, *odrasli*, *studijska_čitaonica*) i koje se navode u vitičastim zagradama.

3.4. Schematron

Schematron je validacijski XML schema jezik koji se temelji na pravilima o iznošenju izjava o prisutnosti ili odsutnosti obrazaca u XML stablima podataka. To je strukturalni jezik koji je izražen u XML-u koji koristi mali skup elemenata i XPath. Schematron ima sposobnost izražavanja ograničenja na načine na koji to ostali XML schema jezici ne mogu učiniti. Na primjer, Schematron može zahtijevati da se sadržaj nekog elementa kontrolira srodnim elementom ili može čak zahtijevati da korijenski element mora imati specifične attribute. Schematron također može specificirati potrebne veze između više XML datoteka. Utemeljen na XPath-u, Schematron specificira i izražava ograničenja koja mogu biti razvijena u XSLT kôdu te ih učiniti praktičnima za upotrebu u aplikacijama za validaciju podataka.¹² Umjesto stvaranja gramatike za XML dokument, Schematron iznosi tvrdnje koje se primjenjuju za specifični kontekst u dokumentu.

¹² Piez, Wendell; Lapeyre, Debbie. Introduction to Schematron, 2008. URL: <http://www.mulberrytech.com/papers/schematron-Philly.pdf> (2019-9-7)

4. Zaključak

XML je jedan od najpoznatijih označiteljskih jezika koji je razvio W3C (*World Wide Web Consortium*), te je i dan danas u širokoj upotrebi. XML je *metajezik* koji služi za formalno opisivanje drugih označiteljskih jezika. XML schema jezicima opisuje se struktura XML dokumenta u koje spadaju: DTD, W3C XML Schema, RELAX NG, Schematron i dr. Dva XML schema jezika koja su u najčešćoj upotrebi su DTD i W3C XML Schema o kojima se najviše posvetilo pozornosti u ovom radu. Oba XML schema jezika imaju svoje prednosti, ali i nedostatke po kojima se može procijeniti koji jezik je u kojem slučaju bolji izbor za konkretnu namjenu. DTD je stariji XML schema jezik te pomoću svoje specifične deklaracije elemenata i atributa definira važeće blokove XML dokumenta. No, s vremenom formati i oblik tih dokumenata su se razvili, te zadobili nove značajke koje više nisu mogle biti obuhvaćene DTD-jem. Ovaj nedostatak DTD-ja W3C XML Schema uspješno je razriješila. Velika prednost W3C XML Scheme nad DTD-jem odnosi se na davanje podrške imenskim prostorima (engl. *namespaces*). Osim toga, W3C XML Schema noviji je schema jezik koji je kompatibilniji s novijim oblicima XML dokumenata. Osim navedenih, u radu je dan osvrt na još dva shema jezika - RELAX NG i Schematron, koji se manje koriste, ali također imaju svoje prednosti u specifičnim područjima obrade i korištenja XML dokumenata. S daljnjim razvitkom XML dokumenata dolaze i nove potrebe za učinkovitijim schema jezicima koji bi te dokumente mogli adekvatno strukturirati i opisati.

Literatura

1. A Gentle Introduction to XML. URL: <https://tei-c.org/release/doc/tei-p5-doc/en/html/SG.html>(2019-9-7)
2. Hedgebeth, D., "Managing structured data with the extensible markup language (XML)", *VINE*. // 38, 3(2008), str. 348- 356. URL: <https://doi.org/10.1108/03055720810904853>(2019-9-7)
3. Hong Quang, N. and Rahayu, W., "XML schema design approach", *International Journal of Web Information Systems*. 1, 3(2005), str. 161-178. URL: <https://doi.org/10.1108/17440080580000091>(2019-9-7)
4. Ioannides, D., "XML schema languages: beyond DTD", *Library Hi Tech*. // 18, 1(2000), str. 9-15. URL: <https://doi.org/10.1108/07378830010314366> (2019-9-7)
5. Liu, M., "DTD schema: a simple but powerful XML schema language", *International Journal of Web Information Systems*. // 4, 4(2008), str. 465-483. URL: <https://doi.org/10.1108/17440080810919495> (2019-9-7)
6. Piez, Wendell; Lapeyre, Debbie. Introduction to Schematron, 2008. URL: <http://www.mulberrytech.com/papers/schematron-Philly.pdf> (2019-9-7)
7. relaxng: RELAX NG Compact Syntax Tutorial. URL: <https://relaxng.org/compact-tutorial-20030326.html> (2019-9-7)
8. van der Vlist, Eric. Comparing XML Schema Languages. URL: <https://www.xml.com/pub/a/2001/12/12/schemacompare.html> (2019-9-7)
9. w3webtutorial: DTD Attributes. URL: <http://www.w3webtutorial.com/dtd/dtd-attribute.php> (2019-9-7)
10. w3: Naming and Addressing: URIs, URLs, ... URL: <https://www.w3.org/Addressing/> (2019-9-7)