

# XML i njegova primjena u MARCXML i MODS standardima

---

**Knežević, Daniel**

**Undergraduate thesis / Završni rad**

**2022**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zadar / Sveučilište u Zadru**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:162:028257>

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

*Download date / Datum preuzimanja:* **2025-01-18**



**Sveučilište u Zadru**  
Universitas Studiorum  
Jadertina | 1396 | 2002 |

*Repository / Repozitorij:*

[University of Zadar Institutional Repository](#)



zir.nsk.hr



DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

Sveučilište u Zadru

Odjel za informacijske znanosti

Sveučilišni preddiplomski studij Informacijske znanosti (jednopedmetni)

**Daniel Knežević**

**XML i njegova primjena u MARCXML i MODS  
standardima**

**Završni rad**

Zadar, 2022.

Sveučilište u Zadru

Odjel za informacijske znanosti

Sveučilišni preddiplomski studij Informacijske znanosti (jednopedmetni)

# **XML i njegova primjena u MARCXML i MODS standardima**

Završni rad

Student/ica:

Daniel Knežević

Mentor/ica:

doc. dr. sc. Krešimir Zauder

Zadar, 2022.



## Izjava o akademskoj čestitosti

Ja, **Daniel Knežević**, ovime izjavljujem da je moj **završni** rad pod naslovom **XML i njegova primjena u MARCXML i MODS standardima** rezultat mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na izvore i radove navedene u bilješkama i popisu literature. Ni jedan dio mojega rada nije napisan na nedopušten način, odnosno nije prepisan iz necitiranih radova i ne krši bilo čija autorska prava.

Izjavljujem da ni jedan dio ovoga rada nije iskorišten u kojem drugom radu pri bilo kojoj drugoj visokoškolskoj, znanstvenoj, obrazovnoj ili inoj ustanovi.

Sadržaj mojega rada u potpunosti odgovara sadržaju obranjenoga i nakon obrane uređenoga rada.

Zadar, 2022.

## Sadržaj

|  |           |
|--|-----------|
| <b>1. UVOD .....</b>   | <b>1</b>  |
| <b>2. POVIJEST NASTANKA XML-a.....</b>                       | <b>2</b>  |
| <b>3. ULOGA I VAŽNOST XML-a .....</b>                        | <b>4</b>  |
| 3.1. Namjena XML-a.....                                      | 4         |
| 3.2. Proširivost XML-a.....                                  | 5         |
| 3.3. Samoopisivanje, pasivnost i neovisnost o softveru ..... | 5         |
| 3.4. Jednostavnost XML-a.....                                | 6         |
| 3.5. Interoperabilnost XML-a.....                            | 6         |
| <b>4. SINTAKSA.....</b>                                      | <b>6</b>  |
| 4.1. XML stablo.....   | 6         |
| 4.2. Dijelovi XML dokumenta .....                            | 7         |
| 4.2.1. Elementi.....   | 7         |
| 4.2.2. Atributi .....  | 8         |
| 4.2.3. Deklaracija.....                                      | 8         |
| 4.2.4. Ostali dijelovi .....                                 | 9         |
| <b>5. PRIMJERI XML DOKUMENATA .....</b>                      | <b>10</b> |
| <b>6. POPRATNE TEHNOLOGIJE.....</b>                          | <b>13</b> |
| 6.1. XML Sheme .....   | 13        |
| 6.2. XSL i njegovi dijelovi .....                            | 14        |
| <b>7. XML I HTML – USPOREDBA.....</b>                        | <b>15</b> |
| <b>8. PRIMJENA XML-a NA BIBLIOGRAFSKE PODATKE.....</b>       | <b>17</b> |
| 8.1. MARCXML .....   | 17        |
| 8.2. MODS .....  | 19        |
| 8.3. XML u knjižnici .....                                   | 20        |
| <b>9. MARCXML I MODS – PRIMJERI .....</b>                    | <b>21</b> |
| <b>10. ZAKLJUČAK.....</b>                                    | <b>27</b> |
| <b>11. LITERATURA .....</b>                                  | <b>28</b> |

## SAŽETAK

Svrha ovog rada je opisati XML kao jedan od najvažnijih označiteljskih jezika, objasniti njegovu primjenu te istaknuti važnost primjene XML-a u knjižnicama u suvremenom informacijskom okruženju. U prvom dijelu rada opisan je označiteljski jezik XML, odnosno njegova povijest kao i kratki pregled jezika koji su mu prethodili, njegovi osnovni pojmovi i sintaksa, popratne tehnologije i primjena, a uz to su navedeni i odgovarajući primjeri. Također, napravljena je njegova usporedba s drugim označiteljskim jezicima, a to se posebno odnosi na usporedbu s HTML-om. Nakon opisa XML-a, slijedi opis dva odabrana standarda napisana u označiteljskom jeziku XML, a to su MARCXML i MODS koji se ponajviše koriste u knjižnicama prilikom digitalizacije građe. Prvo je opisan MARCXML, njegov dizajn, shema i primjena. Zatim je opisan standard MODS, odnosno njegova shema, primjena i dizajn. Osim toga, u radu je istaknuta uloga koju primjena XML-a danas ima u knjižnici, odnosno u procesu digitalizacije građe gdje XML zauzima važnu ulogu kao prikladan format za serijalizaciju, razmjenu i rukovođenje metapodacima u knjižnici. Nakon opisa standarda MARCXML i MODS, slijede njihovi praktični primjeri, kao i njihova usporedba u načinu primjene. Osim toga, navedene su i prednosti, odnosno pozitivne strane njihovog korištenja, a isto tako i nedostaci s kojima se korisnici susreću prilikom korištenja tih standarda.

Ključne riječi: označiteljski jezici, XML, MARCXML, MODS, metapodaci

## 1. UVOD

Brzi razvoj Interneta i tehnologije općenito te važnost mrežnih tehnologije u današnjem društvu dovela je do velike proizvodnje sadržaja u digitalnom obliku kao i do digitalizacije različitih sadržaja, ali i korištenja strukturiranih podataka u svrhe upravljanja, organizacije i istraživanja. Navedeno je u konačnici dovelo do potrebe za razmjenu velike količine podataka. Razmjena podataka je dovela do potrebe za označiteljskim jezicima koji će na strukturiran način omogućiti tu razmjenu između različitih aplikacija. Označiteljski jezici zauzimaju važnu ulogu u suvremenom informacijskom okruženju, a radi se o učinkovitim alatima koji su razvijeni u digitalnom obliku te služe za označavanje metapodataka iz različitih područja ljudskog djelovanja. Za potrebe ovog rada metapodatke možemo definirati kao podatke o podacima koji su temeljni dio svake (digitalne) zbirke, a prisutni su u knjižnicama pa tako i u arhivima i u muzejima. Kao jedan od glavnih označiteljskih jezika koji se danas koriste istaknuo se XML, odnosno *eXtensible Markup Language*, jednostavan i praktičan jezik za označavanje koji omogućuje razmjenu podataka između različitih računalnih aplikacija, istovremeno opisujući te podatke. Njegovo korištenje važno je i u radu knjižnica, gdje je postao jedan od glavnih formata u procesu digitalizacije knjižnične građe te se na njemu temelje različiti metapodatkovni standardi poput MARCXML-a i MODS-a.

Cilj ovog rada je istaknuti važnost označiteljskog jezika XML te ulogu koju zauzima u procesu razmjene podatka među različitim aplikacijama. U radu će se opisati povijest njegovog nastanka, jezici koji su mu prethodili, kao i njegove karakteristike i dijelovi od kojih se sastoji te funkcije koje ti dijelovi imaju. Osim toga, bit će riječi o popratnim tehnologijama koje su vezane za njega, odnosno XML Shemama i transformacijama. Uz to, usporedit će se s HTML-om, drugim jezikom s kojim dijeli određene karakteristike, ali se razlikuje po namjeni. Također, XML ima važnu ulogu i u radu knjižnica pa je cilj rada istaknuti važnost njegove primjene u suvremenom informacijskom okruženju. Kad je riječ o njegovoj ulozi u knjižnicama, posebna pažnja će se posvetiti dvama standardima koji se temelje na XML-u, a to su MARCXML i MODS. Kroz rad će se opisati njihova uloga, prednosti i nedostaci koji se pronalaze prilikom njihovog korištenja. Dakle, u radu će se odgovoriti na pitanja: Koje su prednosti XML-a u odnosu na druge označiteljske jezike?, Koja je važnost primjene XML-a u knjižnicama? te Koje su prednosti i mane MARCXML i MODS metapodatkovnih shema?.

## 2. POVIJEST NASTANKA XML-a

U današnje vrijeme, XML je jedan od označiteljskih jezika koji se najčešće koriste za zapis strukturiranih podataka ili pak dodavanje strukturiranih podataka u slobodan tekst. Međutim, njegovoj pojavi prethodila je pojava više jezika za označavanje, a prvi koji se pojavio naziva se GML (*Generalized Markup Language*). GML su u 60-im godinama 20. stoljeća počeli razvijati Ed Mosher, Ray Lorie i Charles Goldfarb koji su radili za američku firmu IBM, a njihov cilj je bio strukturirati dokumente na način koji će omogućiti da ti dokumenti budu čitljivi na računalu (Banerjee i Reese Jr. 2019, 73). Naziv je dobio zahvaljujući tome što se elementi koje koristi za označavanje nekog dokumenta mogu odnositi općenito na sve dokumente, a ne samo na neku određenu aplikaciju ili stil formatiranja. GML se drži dva temeljna načela, prvo podrazumijeva da se označavanje koristi kako bi se opisala struktura nekog dokumenta, a drugo da je označavanje određeno strogim sintaktičkim pravilima koja su vezana za obradu nekog dokumenta u različitim formatima (Bosančić 2011, 71).

S vremenom se GML počeo dodatno razvijati i nadograđivati te iz njega 1986. nastaje prva službena inačica SGML-a (*Standard Generalized Markup Language*). Razvoj SGML-a donio je napredak u računalnoj obradi podataka, a njime je omogućeno stvaranje vlastitih namjenskih označiteljskih jezika koji se definiraju uz pomoć standardne sintakse te se tako omogućava da sve aplikacije koje prepoznaju SGML mogu upotrebljavati dokumente napisane u tako razvijenim jezicima te im istovremeno omogućava rukovanje takvim dokumentima na prikladan način (Fawcett, Quin i Ayers 2012, 6). Drugim riječima, SGML se može definirati kao meta-jezik, odnosno jezik koji služi za opisivanje dozvoljene gramatike i pravopisa ostalih jezika (Kirasić 2005). Osim toga, meta-jezik određuje simbole i pravila koja služe za stvaranje drugih jezika (Maleš i Mladenović 2007, 11). No, iako se predviđalo da će SGML postići veliki uspjeh te postati najkorišteniji označiteljski jezik, to se ipak nije dogodilo. Iako je SGML bio dobro osmišljen i opsežan, problem je predstavljala njegova složenost, odnosno bio je prekomplikiran za korištenje. Problem je predstavljalo i to što mnoge aplikacije nisu bile u stanju čitati dokumente zapisane u SGML-u, a samim time nisu mogle ni pravilno upravljati takvim dokumentima (Fawcett, Quin i Ayers 2012, 6). Zbog toga su označiteljski jezik HTML koji se bazira na SGML-u te inačica SGML-a, XML kasnije postigli puno veći uspjeh od svog prethodnika.

Najčešće korišten jezik za označavanje koji se temelji na SGML-u zove se HTML (*Hyper Text Markup Language*), a razvijen je početkom 90-ih godina 20. stoljeća kao potporna tehnologija World Wide Weba. Za njegov razvoj zaslužan je fizičar Tim Berners-Lee koji je



bio djelatnik CERN-a u Švicarskoj. Do njegovog razvoja je došlo prilično brzo, a tome je najviše doprinio ubrzani razvoj Interneta i izuzetno brz rast popularnosti WWW-a (Maleš i Mladenović 2007, 14). Djelovanje i korištenje HTML-a prvenstveno se, dakle, veže za izradu web stranica (Kirasić 2005). Taj je jezik za označavanje postao glavno sredstvo za izradu web stranica te je na taj način postao prihvaćen u cijelom svijetu, a počeo se koristiti i izvan područja računalne industrije koja se bavi obradom teksta (Bosančić 2011, 75). Međutim, unatoč svojoj jednostavnosti i prednostima korištenja, HTML je pokazao i određene nedostatke. Jedan od glavnih nedostataka je problem nestandardnosti među softverima te nepostojanje strogih jedinstvenih pravila za procesuiranje HTML dokumenata pa je tako način na koji će se ti dokumenti prikazati na računalu prepušten na izbor različitim web preglednicima. Svaki web preglednik na različiti način može tumačiti HTML pa tako često dolazi do različitih prikazivanja istog koda što stvara probleme web dizajnerima jer ne mogu uskladiti HTML kod s određenim web preglednicima (Bosančić 2011, 76).

Upravo kako bi se prevladali ti nedostaci, organizacija W3C (*World Wide Web Consortium*) je 1996. godine započela razvijati podskup SGML-a, odnosno XML (*eXtensible Markup Language*) koji se razvio u puno uspješniju inačicu SGML-a (Banerjee i Reese Jr. 2019, 73). Ta je organizacija 1996. godine objavila prvi nacrt XML-a, a preporuka W3C-a za njegovo korištenje je objavljena dvije godine kasnije, odnosno 1998. (Fawcett, Quin i Ayers 2012, 6). Cilj s kojim su stvarali XML je bilo kreiranje označiteljskog jezika koji će biti kombinacija jednostavnosti koju posjeduje HTML i opsežnosti i snage koja je karakteristična za SGML (Huitfeldt 2004 prema Bosančić 2011, 76). Glavna karakteristika po kojoj se XML razlikuje od SGML-a je njegova jednostavnost. XML je puno jednostavniji označiteljski jezik od SGML-a, a jedan od najboljih pokazatelja te činjenice su njihove osnovne specifikacije. Pošto je SGML puno kompliciraniji i složeniji, njegova se osnovna specifikacija sastoji od 150 stranica, dok je osnovna specifikacija XML puno sažetija i sastoji se od samo 35 stranica (Maleš i Mladenović 2007, 97). Isto tako, XML se pokazao praktičnijim od svog prethodnika i zato što za obradu njegovih dokumenata nije potrebno imati odgovarajući DTD, odnosno *Document Type Definition*, za razliku od SGML-a kod kojega je DTD obavezan (Bosančić 2011, 76). Uz to, XML je prilagođen i poboljššan za korištenje na webu, a to se postiglo zahvaljujući tome što su iz SGML-a uklonjeni svi elementi koji su nepotrebni za WWW. Zbog toga se XML jako brzo rasprostranio i postao jezik za razmjenu podataka na webu te između računalnih sustava (Maleš i Mladenović 2007, 76). XML je dosad objavljen u dvije verzije. Prva verzija je prvi put

predstavljena 1996. pod nazivom XML 1.0, a druga verzija je po prvi puta objavljena 2004. godine pod nazivom XML 1.1 (Bosančić 2011, 77).

### **3. ULOGA I VAŽNOST XML-a**

Nedugo nakon objavljivanja XML-a, taj označiteljski jezik zauzeo je važno mjesto u procesu obrade podataka te se brzo pretvorio u jedan od najvažnijih jezika za označavanje.

#### **3.1. Namjena XML-a**

XML je vrlo rasprostranjen te se upotrebljava u mnogim kontekstima, koristi se u razne svrhe pa je njegova primjena vrlo široka i zastupljena u različitim područjima (Cole i Myung 2013, 3). Jedno od najvažnijih područja koje XML zauzima je ono vezano za informacijske tehnologije, često je zastupljen u različitim informatičkim sustavima, najviše za razmjenu i obradu podataka putem interneta i WWW-a (W3Schools. n.d.). Osim toga, XML je prisutan i u mnogim aplikacijama koje ga koriste za čuvanje internih podataka potrebnih za osnovno funkcioniranje aplikacije ili pak za komunikaciju s drugim aplikacijama. Zbog navedenog programeri moraju biti dobro upoznati s njegovim funkcijama i pravilima kako bi ga mogli koristiti na pravilan način i iskoristiti prednosti koje njegovo korištenje donosi (Fawcett, Quin i Ayers 2012, 3). Također, XML olakšava njihov posao jer podatke pohranjuje u obliku običnog teksta te tako osigurava neovisno softversko i hardversko dijeljenje, pohranu i prijenos podataka (W3Schools. n.d.). Fawcett, Quin i Ayers (2012, 4) ističu dvije osnovne namjene koje ima XML. Kao prvu namjenu navode opisivanje jednostavnijih, odnosno primitivnijih podataka u koje se ubrajaju, na primjer, konfiguracijske datoteke. Kao drugu važnu funkciju navode usmjerenost na dokumente (engl. *document-centric*) gdje se sadržaj koji je slabije strukturiran, poput dijelova pravnih dokumenata ili poglavlja neke knjige, označavaju metapodacima. XML koji je usmjeren na dokumente koristi se prvenstveno kako bi se olakšalo objavljivanje podataka putem više kanala istovremeno, kako bi se omogućilo ponovno korištenje postojećeg sadržaja, odnosno podataka ili kako bi se tekst obogatio dodatnim podacima kao što su, na primjer, oni potrebni za računalnu analizu prirodnog jezika. Uz to, XML ima važnu ulogu i u knjižnicama koje ga koriste u procesu digitalizacije svojih zbirki te za očuvanje bibliografskih i digitalnih podataka. S tim ciljem su se razvili različiti standardi temeljeni na XML-u poput MODS-a, MARCXML-a i ostalih (Banerjee i Reese Jr. 2019, 67). O ulozi XML-a u radu knjižnica u suvremenom informacijskom okruženju govorit će se u kasnijim poglavljima.

### 3.2. Proširivost XML-a

Kratica XML dolazi od riječi *eXtensible Markup Language* iz čega se može zaključiti da se radi o označiteljskom jeziku čija je karakteristika proširivost, odnosno da se radi o proširivom jeziku za označavanje. Razlog zbog kojega ga se naziva proširivim je u tome što omogućuje definiciju vlastitih oznaka i strukture oznaka. Za razliku od HTML-a, koji propisuje koje oznake postoje i kako se koriste, XML omogućuje da definiramo vlastite oznake i pravila njihova korištenja. Na primjer, može se definirati oznaka <film> koja prima obveznu oznaku <godina\_izdanja> koja se ne smije ponavljati; obveznu oznaku <režiser> koja se smije ponavljati i neobveznu oznaku <opis> koja se ne smije ponavljati.

### 3.3. Samoopisivanje, pasivnost i neovisnost o softveru

XML je dizajniran tako da bude samoopisujući jezik (W3Schools. n.d.). Dokumenti koji su zapisani u XML-u stoga se često nazivaju samoopisujućim dokumentima zbog toga što je njihov sadržaj velikim dijelom sveden na opis samih podataka koji se nalaze u njima (Maleš i Mladenović 2007, 98). Uz to, Kirasić (2005) kao jednu od karakteristika XML-a navodi njegovu pasivnost. XML predstavlja jezik koji je pasivan zato što dokument koji je napisan u njemu ne može raditi ništa, ne može se „izvesti“. Dakle, XML predstavlja samo tekstni format koji opisuje podatke nekog dokumenta. Istu stvar navodi i W3Schools (n.d.) koji XML definira kao skup informacija koje su upakirane u oznake, što predstavljaju kroz sljedeći primjer:

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

#### Ispis 1. Primjer zapisa u XML-u

Prva stvar koju navedeni ispis pokazuje jest to kako je dokument zapisan u XML-u samoopisujuć, odnosno kako objašnjava sam sebe. Radi se o poruci, odnosno zapisu razmijenjenom između dva prijatelja, a zapis u sebi sadrži informaciju o tome tko je pošiljalac (Jani) i primatelj (Tove), naslov i tijelo poruke. Ipak, XML nema drugu funkciju osim toga što donosi podatke u primjeru, predstavlja samo niz informacija te ne radi ništa drugo, kao što je to, na primjer, definicija prikaza ovih podataka (W3Schools. n.d.). Osim toga, glavna stvar kod

XML-a je to što nije vezan za niti jedan softver, niti jednu aplikaciju, programski jezik ili operativan sustav već se bazira na običnom tekstu i otvorenom standardu.

### 3.4. Jednostavnost XML-a

Kao što je već rečeno u prethodnom poglavlju, razlog zbog kojega se XML istakao kao jedan od najvažnijih označiteljskih jezika leži u njegovoj jednostavnosti. Cole i Myung (2013, 4) tako navode da je XML standard i napravljen s ciljem da pojednostavi organizaciju, održavanje i dijeljenje podataka prilikom kodiranja i opisivanja podataka u smislu serijalizacije (engl. *serialization*) podataka, odnosno pretvaranja programskih struktura u tekst neovisnog o programskom jeziku, programskoj logici ili nekoj egzaktnoj aplikaciji. Osim toga, Fawcett, Quin i Ayers (2012, 14) ističu kako se XML često koristi zbog svoje fleksibilnosti te zbog toga što koristi prirodni jezik razumljiv ljudima. Stoga dokumente zapisane u XML-u na jednostavan način mogu uređivati kako ljudi tako i strojevi (iako je u standardnoj praksi XML usmjeren na strojnu obradu).

### 3.5. Interoperabilnost XML-a

Fawcett, Quin i Ayers (2012, 12) ističu važnost XML-a zbog njegove interoperabilnosti. XML format tako omogućava bržu razmjenu podataka između različitih aplikacija. Kao primjer navode sustav Microsoft Word kojem je korištenje XML-a, odnosno mogućnost spremanja dokumenata u XML formatu sa strukturom koja je dokumentirana omogućilo čitanje takvih dokumenata u različitim aplikacijama te stvaranje dokumenata čak i uz korištenje najosnovnijih alata. Uz to, XML omogućuje da se takvi dokumenti poprave ukoliko dođe do određenog oštećenja, što prije nije bilo moguće jer bi oštećeni dokumenti bili izgubljeni bez mogućnosti popravka.

## 4. SINTAKSA

### 4.1. XML stablo

Svaki XML dokument sastoji se od niza znakova od kojih su neki neizostavni te imaju važniju ulogu od ostalih znakova dok drugi nisu obavezni, već služe kao nadopuna obaveznim znakovima. U svakom slučaju, kako bi XML dokument bio valjan te ispravno napisan, struktura elemenata koji ga sačinjavaju mora biti oblikovana kao stablo. Svako XML stablo mora započinjati glavnim, odnosno korijenskim elementom koji se na engleskom naziva *root element*. Na korijenski element se vezuju ostali elementi koji se nazivaju *child elements* te

funkcioniraju kao potomci korijenskog elementa koji ima ulogu „roditelja“. Osim toga, na svaki element potomak se mogu vezati novi elementi potomci (W3Schools. n.d.). Nazivi roditelj, potomci i djeca koristi se kako bi se opisala veza među elementima XML dokumenta, pošto je prikaz XML stabla vrlo sličan prikazu obiteljskog stabla (Maleš i Mladenović 2007, 100).

```
<sastanak>
  <datum>
    <dan>5</dan>
    <mjesec>veljača</mjesec>
    <godina>2022</godina>
  </datum>
</sastanak>
```

## Ispis 2. Informacije o sastanku zapisane u XML-u

Gledajući navedeni ispis može se vidjeti kako ulogu korijenskog elementa u ovom zapisu ima <sastanak> na kojega se vezuje element potomak <datum>. Na element potomak se zatim vezuju drugi elementi potomci <dan>, <mjesec> i <godina>. U XML dokumentu je hijerarhija jasno određena te svaki dokument može imati samo jedan korijenski element. Upravo se mogućnost korištenja samo jednog korijenskog elementa često navodi kao zamjerka XML-u jer se smatra kako bi korištenje više korijenskih elemenata u jednom dokumentu bilo praktičnije i korisnije (Fawcett, Quin i Ayers 2012, 32).

## 4.2. Dijelovi XML dokumenta

### 4.2.1. Elementi

Kao što je već rečeno, svaki XML dokument je običan tekst. Kao i kod HTML-a, osnovna sintaktička jedinica XML dokumenata naziva se element. Pod elementom se podrazumijeva početna i završna oznaka te sadržaj koji se nalazi između tih oznaka (W3Schools. n.d.). Dakle, svaki element sastoji se od početne oznake koja se zapisuje znakovima „<“ i „>“, a između njih se zapisuje naziv elementa kao, na primjer, <mjesec> te završne oznake koja se također označava znakovima „<“ i „>“, a između njih se zapisuje naziv elementa ispred kojeg se dodaje kosa crta „/“ koja označava kraj elementa, npr. </mjesec>. Osim toga, element sačinjava i podatak, odnosno sadržaj koji se nalazi između početne i završne oznake kao, na primjer, <mjesec>veljača</mjesec> (Fawcett, Quin i Ayers 2012, 30). Podatak koji se nalazi između početne i završne oznake predstavlja sadržaj i vrijednost elementa (Maleš

i Mladenović 2007, 102). Tako u navedenom primjeru „veljača“ predstavlja sadržaj i vrijednost elementa „mjesec“. Element može imati različite sadržaje, a može sadržavati i druge elemente, attribute ili tekst (W3Schools. n.d.). Ukoliko sadržaj elementa sadrži druge elemente nazivamo ga složenim elementom. Ipak, element može biti i bez sadržaja, odnosno prazan. U tom slučaju ne mora završavati završnom oznakom već se na kraju imena u početnoj oznaci može dodati kosa crta „/“ (Kirasić 2005). To se može vidjeti u primjeru `<mjesec/>`. Međutim, prazan element može se zapisati i koristeći početnu i završnu oznaku između kojih se ne nalazi nikakav podatak (Cole i Myung 2013, 52). Jedan od takvih elemenata je `<mjesec></mjesec>`. Također, postoje pravila vezana za nazive elemenata pa tako njihovi nazivi ne mogu započinjati brojevima, interpunkcijskim znakovima poput zareza, točke ili spojnice, ne smiju započinjati razmakom te sami nazivi ne smiju sadržavati razmake (Fawcett, Quin i Ayers 2012, 31).

#### 4.2.2. Atributi

Druga vrsta znakova koja je važna u XML dokumentu su atributi. Atributi sadrže podatke, odnosno dodatne informacije za određeni element (W3Schools. n.d.). Oni se najčešće koriste kako bi se obogatilo i nadopunilo značenje elemenata, zbog čega ih se može usporediti s pridjevima koji nadopunjavaju imenice (Cole i Myung 2013, 52). Svaki element može posjedovati jedan ili više atributa. Atributi se mogu pojaviti samo kao dio početne oznake, a sastoje se od imena atributa i vrijednosti atributa koja uvijek mora biti zapisana u navodnicima (" ") (Maleš i Mladenović 2007, 103). Npr., u zapisu `<mjesec godina="2022">veljača</mjesec>` dio zapisa „godina="2022"“ predstavlja atribut elementa *mjesec*, odnosno *godina* predstavlja ime atributa dok 2022 predstavlja vrijednost tog atributa. Između naziva i vrijednosti atributa obavezno je koristiti znak jednakosti „=“, a vrijednost atributa se mora zapisati u navodnicima. Također, atributi ne mogu imati više vrijednosti niti strukturu stabla te se ne mogu lako proširiti (W3Schools. n.d.). Osim toga, unutar jednog elementa ne može se nalaziti više atributa s istim imenom. (Kirasić 2005).

#### 4.2.3. Deklaracija

Jedan od dijelova XML dokumenta može biti deklaracija. XML deklaracija nije obavezan dio dokumenta, a, ukoliko je prisutna, mora biti napisana na početku dokumenta te tako sačinjava dio koji se naziva *prolog* (W3Schools. n.d.). Svrha deklaracije je označavanje verzije XML specifikacije po kojoj je dokument izrađen, a u svakoj deklaraciji je zapisana

verzija XML-a. Postoje samo dvije verzije koje se mogu koristiti, a to su verzija 1.0 ili 1.1 (Kirasić 2005). Kao primjer može se uzeti deklaracija koja označava verziju 1.0:

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

Kao što je vidljivo u primjeru, XML deklaracija uvijek mora biti zapisana malim slovima (Maleš i Mladenović 2007, 102). Osim informacije o verziji XML-a, deklaracija može sadržavati i podatak o kodnoj stranici koja je korištena prilikom izrade dokumenta. XML podržava i najčešće koristi *Unicode*, odnosno njegovu varijantu UTF-8, što se može vidjeti u sljedećem primjeru:

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

XML može koristiti i druge varijante *Unicode*-a poput UTF-16, no ipak se preporučuje i najčešće koristi varijanta UTF-8 (Fawcett, Quin i Ayers 2012, 28).

#### 4.2.4. Ostali dijelovi

U XML dokument mogu se ubrojiti i ostali dijelovi koji se mogu dodati po izboru kreatora dokumenta, a to su: komentari, procesne naredbe, DTD, reference na entitete te CDATA sekcije. Komentari služe kako bi se u dokument mogle dodavati određene napomene namijenjene korisniku ili autoru dokumenta (Maleš i Mladenović 2007, 104). Komentari tako omogućuju uključivanje zabilješki u XML dokument (Cole i Myung 2013, 57). Mogu se nalaziti u bilo kojem dijelu dokumenta, a moraju započinjati znakovima „<!--“ te završavati znakovima „-->“ dok se između njih nalazi željeni tekst (Kirasić 2005). To se može vidjeti u sljedećem primjeru: <!-- ovako treba izgledati xml komentar -->.

U XML dokumentu mogu se nalaziti i procesne naredbe. One se koriste kako bi se XML aplikacijama omogućile dodatne upute i informacije te samim time olakšalo korištenje XML dokumenta (Cole i Myung 2013, 54). Procesne naredbe moraju započinjati znakovima „<?“ te završavati znakovima „?>“, a između njih se nalazi sadržaj, odnosno ime programa kojem je naredba namijenjena i, ukoliko postoje, parametri naredbe (Maleš i Mladenović 2007, 103). Najčešće se susreće procesna naredba *stylesheet* namijenjena programima za prikazivanje XML dokumenta (Kirasić 2005). Takva se procesna naredba može vidjeti u sljedećem primjeru:

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

Program kojem je namijenjena naredba u ovom slučaju je *xml-stylesheet*, a ovom naredbom će se postići to da program XSL pretvori, odnosno transformira ovaj XML dokument.

Također, u XML dokumentu mogu se naći i DTD te reference na entitete. DTD je skraćenica engleskog naziva *Document Type Definition*. Također, ni DTD nije obavezan u

XML dokumentu, a ukoliko se koristi njegova uloga određivanje strukture i dopuštenih elemenata i atributa u određenom XML dokumentu (W3Schools. n.d.). Osim toga, DTD određuje odnos između elemenata, atributa i entiteta te se njegovim korištenjem može provjeriti valjanost dokumenta (Cole i Myung 2013, 165).

```
<!DOCTYPE sastanak
[
<!ELEMENT sastanak (datum, dan, mjesec, godina)>
<!ELEMENT datum (#PCDATA)>
<!ELEMENT dan (#PCDATA)>
<!ELEMENT mjesec (#PCDATA)>
<!ELEMENT godina (#PCDATA)>
]>
```

### Ispis 3. Primjer DTD-a u XML dokumentu

Ispis 3 prikazuje primjer jednog DTD-a. Iz njega se može vidjeti kako je za korijenski element određen pojam *sastanak* te kako element *sastanak* mora sadržavati elemente potomke *datum*, *dan*, *mjesec* i *godina*. Također, određeno je kako će elementi *datum*, *dan*, *mjesec* i *godina* sadržavati „PCDATA“, odnosno tekst po izboru kreatora dokumenta. Također, u DTD-u mogu biti prisutne i reference na entitete koje se odnose na vrijednost nekog entiteta, a zapisuju se uz pomoć znaka & nakon kojeg dolazi ime entiteta i točka-zarez (;) (Kirasić 2005).

Osim toga, kao jedan od dijelova XML dokumenta mogu se pojaviti i CDATA sekcije. One se koriste kada se u dokument žele uvrstiti dijelovi teksta koji neće biti obrađeni na nijedan način (Kirasić 2005). Uz to, CDATA sekcije omogućavaju kreatoru dokumenta da u njega uvrste znakove koji nisu normalizirani i raščlanjeni (Cole i Myung 2013, 57). Sekcije započinju znakovima „<![CDATA[“, a završavaju znakovima „]]>“. Između tih znakova nalazi se tekst koji može sadržavati bilo kakve znakove, poput znaka „>“ ili „&“ (Fawcett, Quin i Ayers 2012, 39).

## 5. PRIMJERI XML DOKUMENATA

Kao što se može vidjeti u prethodnom poglavlju, XML ima jednostavna i jasna sintaktička pravila kojih se treba pridržavati kako bi dokumenti koji se izrađuju bili valjani. Ukoliko se slijede zadana pravila, izrada jednostavnih XML dokumenata ne bi trebala predstavljati problem. Kao primjer kreiranja jednostavnih dokumenata mogu se uzeti sljedeći XML dokumenti:



```

<?xml version="1.0" encoding="UTF-8"?>
<videogames>
  <videogame>
    <developers>
      <developer>EA Vancouver</developer>
      <developer>EA Romania</developer>
    </developers>
    <publisher>EA Sports</publisher>
    <title>FIFA 22</title>
    <series>FIFA</series>
    <genre>Sports</genre>
    <release>01/10/2021</release>
    <platforms>
      <platform>Microsoft Windows</platform>
      <platform>Nintendo Switch</platform>
      <platform>PlayStation 5</platform>
      <platform>Xbox Series X/S</platform>
    </platforms>
    <price currency="EUR">59,99</price>
  </videogame>
</videogames>

```

#### Ispis 4. Opis videoigre zapisan u XML-u

Ispis 4 prikazuje primjer jednostavnog XML dokument u kojem su zapisane informacije vezane za videoigru *FIFA 22*. Prvi korak u kreiranju dokumenta bio je stvaranje prologa, odnosno početnog dijela u kojem se nalazi XML deklaracija. Iz nje se može vidjeti da je verzija XML dokumenta u ovom slučaju 1.0 te da je za izradu dokumenta korištena kodna stranica *Unicode*, odnosno njena inačica UTF-8. Zatim je određen korijenski element, a to je u ovom slučaju „videogames“. Na njega se zatim vezao njegov element potomak, a to je „videogame“. Kao sljedeći korak elementu „videogame“ su se dodali njegovi elementi potomci, a to su „developers“, „publisher“, „title“, „series“, „genre“, „release“, „platforms“ i „price“. Zatim su elementima „developers“ i „platforms“ dodani njihovi elementi potomci, odnosno „developer“ i „platform“. Svi elementi sadržavaju početnu i završnu oznaku, a između njih se nalazi sadržaj elementa. Također, u ovom dokumentu je izrađen samo jedan atribut i to za element „price“, naziv atributa je „currency“, a vrijednost je EUR, a atribut je korišten kako bi se dodatno pojasnio element „currency“, odnosno kako bi se označilo da je cijena izražena u eurima.

```

<?xml version="1.0" encoding="UTF-8"?>
<racuni trgovina="Eurospin Zadar">
  <racun broj_blagajne="5">
    <artikli>
      <artikl kol="2">
        <naziv>Svježi kvasac 42g</naziv>
        <iznos valuta="HRK">5.78</iznos>
      </artikl>
      <artikl kol="3">
        <naziv>Sok 100% Jabuka</naziv>
        <iznos valuta="HRK">19.96</iznos>
      </artikl>
      <artikl kol="1">
        <naziv>Sir Emmental 400g</naziv>
        <iznos valuta="HRK">16.49</iznos>
      </artikl>
      <artikl kol="1">
        <naziv>Šampinjoni 250g</naziv>
        <iznos valuta="HRK">5.99</iznos>
      </artikl>
      <artikl kol="1">
        <naziv>Krumpir 5kg</naziv>
        <iznos valuta="HRK">14.79</iznos>
      </artikl>
      <artikl kol="1">
        <naziv>Pecivo Hamburger 300g</naziv>
        <iznos valuta="HRK">7.49</iznos>
      </artikl>
    </artikli>
    <ukupno valuta="HRK">73.29</ukupno>
    <placanje>Kreditna kartica</placanje>
    <za_vratiti valuta="HRK">0.00</za_vratiti>
    <datum>28/08/2021</datum>
    <broj_racuna>32593/310003/5</broj_racuna>
  </racun>
</racuni>

```

### Ispis 5. Račun iz trgovine zapisan u XML-u

I XML dokument iz ispisa 5 je napravljen na vrlo sličan način, razliku predstavlja jezik, odnosno ovaj dokument je zapisan na hrvatskom, dok je dokument iz ispisa 4 zapisan na engleskom. Dokument iz ispisa 5 predstavlja račun iz trgovine zapisan u XML-u. Ovaj dokument također započinje prologom, odnosno XML deklaracijom. Važno je napomenuti kako taj dio nije obavezan, već je napravljen po izboru kreatora dokumenta, a dokument bi funkcionirao i bez tog dijela. Zatim je određen korijenski element, u ovom slučaju radi se o elementu „racuni“ koji ima atribut čije je ime „trgovina“, a vrijednost „Eurospin Zadar“, a korišten je kako bi se dodatno opisalo u kojoj su trgovini ispostavljeni računi. Zatim je kreiran element potomak, odnosno „racun“, koji ima atribut koji pojašnjava na kojoj je blagajni uspostavljen račun. Zatim su kreirani njegovi elementi potomci, odnosno „artikli“, „ukupno“, „placanje“, „za\_vratiti“, „datum“ i „broj\_racuna“. Zatim su kreirani elementi potomci za element „artikli“. Elementi potomci nazvani su „artikl“, a svaki od njih ima atribut čiji je naziv

„kol“, odnosno količina, a koristi se kako bi se definiralo kolika je količina kupljenih proizvoda. Pošto je dokument napisan na hrvatskom, može se primijetiti da znakovi poput č ili ć nisu dopušteni u nazivu elementa te se moraju zamijeniti drugim znakovima kao što je c. Također, i u ovom dokumentu svi elementi sadržavaju početnu i završnu oznaku, a između njih se nalazi sadržaj elementa. Pošto su u izradi dokumenta ispoštovana sva sintaktička pravila XML, ovakvi dokumenti se definiraju kao valjani.

## 6. POPRATNE TEHNOLOGIJE

### 6.1. XML Sheme

XML Sheme, koje se još nazivaju XSD (*XML Schema Definition*) definiraju se kao popratna tehnologija u obliku teksta koja služi za opisivanje strukture XML dokumenta. Imaju jednaku namjenu kao i DTD, ali razliku predstavlja to što su XML Sheme opsežnije i novije od DTD-a (Kirasić 2005). Kao i DTD, XML Sheme se koriste za provjeravanje valjanosti dokumenta (Fawcett, Quin i Ayers 2012, 17). Osim toga, XML Sheme određuju nazive elemenata za određenu aplikaciju, nazive atributa i sadržaj koji su potrebni za određivanje gramatike određene aplikacije (Cole i Myung 2013, 60). Također, utvrđuju oznake i redoslijed dijelova u XML dokumentu, pravila koja se moraju poštovati kako bi se dokument smatrao valjanim te uvjetuju i ograničavaju strukturu XML dokumenta i njegov sadržaj (Kirasić 2005).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:element name="sastanak">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="datum" type="xs:string"/>
      <xs:element name="dan" type="xs:string"/>
      <xs:element name="mjesecc" type="xs:string"/>
      <xs:element name="godina" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
```

#### Ispis 6. Primjer XML Sheme

Ispis 6 prikazuje na koji način može izgledati jedna XML Shema. Kao što se vidi iz ispisa, Shema započinje XML deklaracijom. Zatim slijedi oznaka kojom se opisuje element „sastanak“ te oznaka „<xs:complexType>“ kojom je određeno da će element „sastanak“ biti složen. Oznaka „<xs:sequence>“ definira da se taj složeni element sastoji od niza znakova, a zatim

slijede oznake koje određuju vrstu svakog pojedinog elementa. U ovom slučaju, svi elementi su tipa *string*, odnosno svi elementi kao vrstu teksta imaju niz znakova.

Iako DTD i XML Sheme imaju istu funkciju, preporuča se korištenje XML Shema te se ističu njihove prednosti u odnosu na DTD. Kao glavna prednost XML Shema navodi se to što su zapisane u XML-u, zbog čega nije potrebno učenje novog jezika, za razliku od DTD-a koji koristi zasebnu sintaksu (Fawcett, Quin i Ayers 2012, 117). Osim toga, prednost je i u tome što se mogu proširiti različitim dodacima, podržavaju vrste podataka i XML prostor imenovanja (engl. *namespace*) koji predstavlja metodu kojom se izbjegavaju nesporazumi prilikom imenovanja elemenata u XML dokumentu (W3Schools. n.d.). Podržavanje vrsti podataka predstavlja veliku prednost jer XML Sheme tako omogućavaju lakše opisivanje dopuštenog sadržaja u dokumentu, jednostavnije provjeravanje valjanosti podataka, lakše određivanje ograničenja i formata podataka te lakše pretvaranje podataka (W3Schools. n.d.).

## 6.2. XSL i njegovi dijelovi

Važnu ulogu za XML imaju i alati koji omogućavaju transformacije XML dokumenata iz postojećeg u neki drugi oblik ili format. Kao najvažniji alat koji omogućuje transformacije ističe se jezik XSL (*eXtensible Stylesheet Language*). Radi se o označiteljskom jeziku čija je glavna namjena transformiranje XML dokumenta iz jednog oblika u drugi (Maleš i Mladenović 2007, 119). Također, opisuje na koji se način trebaju prikazati XML elementi. XSL se sastoji od 4 dijela, a to su: XSLT, XPath, XSL-FO te XQuery (W3Schools. n.d.).

Najvažniji dio XSL jezika je XSLT, odnosno *XSL Transformations* koji se koristi za pretvaranje XML dokumenta u drugi XML dokument ili u neki drugi format poput HTML-a (W3Schools. n.d.). Osim toga, XSLT ima sposobnost obrađivanja podataka koji ne spadaju u XML te transformiranje običnog teksta u XML format (Fawcett, Quin i Ayers 2012, 240). Također se definira kao funkcionalan i deklarativan programski jezik koji se razlikuje od ostalih programskih jezika jer se temelji na predlošcima (engl. *templates*), a oni mu pokazuju na koji način treba upravljati elementima i atributima koji se nalaze u XML dokumentu koji se transformira (Cole i Myung 2013, 297). Pošto je XSLT deklarativan jezik, predlošci mogu biti navedeni bilo kojim redoslijedom, a označavaju se oznakom „xsl:template“ te svaki od njih ima atribut „match“ (Maleš i Mladenović 2007, 119, 124). Najčešće se koristi za pretvaranje XML dokumenata u formate koji su namijenjeni prezentaciji, npr. HTML ili za pretvaranje dokumenata prilagođenih određenoj aplikaciji u format koji zahtjeva druga aplikacija (Fawcett, Quin i Ayers 2012, 240). Osim toga, može se koristiti za dodavanje ili uklanjanje elemenata i

atributa u ili iz izlazne datoteke kao i za preuređivanje i sortiranje elemenata te odlučivanje na koji način će elementi biti prikazani (W3Schools. n.d.).

Drugi dio XSL-a je XPath, odnosno *XPath Language*, jezik koji se koristi za razlikovanje pojedinih dijelova određenog XML dokumenta i njihov odabir za kasnije korištenje (Fawcett, Quin i Ayers 2012, 216). XPath je važan i za XSLT koji ga za vrijeme transformacija koristi za traženje informacija u XML dokumentu (W3Schools. n.d.). Povezanost između ta dva jezika je važna te je korištenje XSLT-a bez XPatha besmisleno pošto u tom slučaju nije moguće obrađivanje samo jedne skupine podataka, nego je potrebna obrada cijelog dokumenta (Maleš i Mladenović 2007, 119). XPath u sebi sadrži više od 200 ugrađenih funkcija, a u njemu se može nalaziti 7 čvorova: element, atribut, tekst, prostor imenovanja, procesne instrukcije, komentar i čvorovi dokumenta. Među čvorovima koji se nalaze u njemu postoji hijerarhija te neki od njih čine pretke, a neki potomke (W3Schools. n.d.). Uz to, koristi predikate koji filtriraju rezultate te tako omogućuju pronalaženje određenog čvora (Fawcett, Quin i Ayers 2012, 19).

Svojevremeno je u uporabi bio i dio XSL-a koji se zove XSL-FO (*XSL Formatting objects*), a radi se o jeziku koji se koristio za definiranje izgleda i oblikovanje XML dokumenata (Fawcett, Quin i Ayers 2012, 240). Međutim, ukinut je 2013. godine te se danas više ne koristi kao dio XSL-a.

Kao dio XSL-a ubraja se i XQuery, a radi se o jeziku koji služi za postavljanje upita XML dokumentima te za traženje i izdvajanje elemenata i atributa iz tih dokumenata. Stvoren je na elementima XPatha te podržava iste operatore i funkcije kao XPath 2.0 (W3Schools. n.d.). Također, sličan je XSLT-u te može raditi na zasebnim dokumentima, iako se često koristi i na većim zbirkama. Razlika između XQueryja i XSLT-a je u tome što XQuery nije napisan u XML-u, za razliku od XSLT-a, zbog čega zahtjeva posebni i prilagođeni softver za obradu, dok se XSLT obrađuje u klasičnim XML obrađivačima (Fawcett, Quin i Ayers 2012, 21).

## **7. XML I HTML – USPOREDBA**

XML, kao jedan od najvažnijih označiteljskih jezika, često se uspoređuje s ostalim jezicima za označavanje, a najčešća usporedba je ona s HTML-om, označiteljskim jezikom koji je namijenjen izradi web stranica. Radi se o jeziku koji sadrži niz elemenata čija je uloga opisivanje web stranica i davanje uputa internetskom pregledniku o tome kako prikazati sadržaj na webu (W3Schools. n.d.).

XML i HTML imaju određene sličnosti, a XML se često smatra zamjenom za HTML, iako se ta tvrdnja ne može smatrati točnom (Maleš i Mladenović 2007, 98). Ono što je ovim jezicima zajedničko je to što za označavanje dijelova određenog dokumenta koriste oznake. S druge strane, namjena njihovih oznaka je različita, zbog čega se XML ne može smatrati zamjenom za HTML (Kirasić 2005). Oznake koje se koriste u XML-u služe kako bi se opisalo značenje i struktura podataka dok se oznake u HTML-u koriste za određivanje izgleda i stila u kojem će ti podaci biti zapisani (Microsoft. n.d.). Jednostavnije rečeno, HTML služi opisivanju dijelova dokumenata za potrebe grafičkog prikaza tog dokumenta. XML nema nikakve veze s prikazom, već služi strukturiranju i opisu bilo kakvih podataka. Tako će u HTML dokumentima biti prikazane informacije o izgledu sadržaja poput veličine i vrste fonta, položaju naslova i odlomaka i slično, za razliku od XML dokumenata koji takve informacije ne sadrže (Maleš i Mladenović 2007, 98). Osim toga, razlika je i u tome što su oznake u HTML-u uvijek unaprijed određene, dok u XML-u to nije slučaj (W3Schools. n.d.). Uz to, XML ima stroža sintaktička pravila kojih se treba obavezno pridržavati kako bi dokumenti bili valjani, za razliku od HTML-a. Tako se u XML dokumentu ne smiju nikako izostaviti znakovi poput završne oznake elementa jer će se u tom slučaju pojaviti greška, a dokument neće biti valjan. Također, u XML zapisu ne smije doći do preklapanja elemenata, vrijednost atributa se mora obavezno zapisati u navodnicima te postoji jasna razlika između velikih i malih slova (Bosančić 2011, 77). S druge strane, HTML također ima jasna pravila, ali ona se ne moraju uvijek poštovati. Kao primjer može se uzeti korištenje završne oznake elementa koja se koristi i u HTML-u kao i u XML-u. Razlika je u tome što se u HTML-u završna oznaka može izostaviti jer sustav neće javiti pogrešku, dok je u XML-u to nemoguće (Maleš i Mladenović 2007, 104). Razliku predstavlja i to što su XML dokumenti obrađeni uz pomoć XML parsera, za razliku od HTML dokumenata koji nemaju takve aplikacije (Bosančić 2011, 77).

Kao jedna od najvažnijih prednosti XML-a navodi se upravo to što nema unaprijed određene oznake. To korisnicima omogućuje kreiranje bilo koje oznake kojom će moći opisati podatke i njihovu strukturu, što kod HTML-a nije moguće pošto sadrži unaprijed određene oznake (Microsoft. n.d.). Također, velika prednost je i u tome što XML odvaja sadržaj dokumenta od njegove prezentacije što prilikom premještanja podataka iz jedne aplikacije u drugu ne ometa propusnost zbog toga što se ne prenose suvišne informacije vezane za dizajn i izgled dokumenta (Fawcett, Quin i Ayers 2012, 10).

## 8. PRIMJENA XML-a NA BIBLIOGRAFSKE PODATKE

Zbog prednosti koje donosi njegovo korištenje, poput lakše i brže razmjene podataka, XML je zauzeo važno mjesto i u radu knjižnica, posebno u njihovom radu s bibliografskim podacima prilikom digitalizacije, odnosno izrađivanja digitalnih zbirki. Kako bi sačuvale bibliografske podatke i uspješno digitalizirale svoje analogne zbirke, mnoge knjižnice prihvatile su i počele se služiti metapodatkovnim shemama koje su zapisane u XML-u. Upravo su zbog toga metapodatkovne sheme u današnje vrijeme postale neizostavni dio u digitalnim publikacijama, a samim time i u knjižnicama u suvremenom informacijskom okruženju (Banerjee i Reese Jr. 2019, 67).

Među metapodatkovnim shemama zapisanim u XML jeziku koje se koriste u procesu digitalizacije nalaze se MARCXML te MODS (*Metadata Object Description Schema*).

### 8.1. MARCXML

MARCXML je inačica MARC-a koju je 1994. godine razvila Kongresna knjižnica te se koristi u knjižnicama za razmjenu bibliografskih podataka i opis građe. Radi se o standardu koji se temelji na XML-u, odnosno koji predstavlja MARC 21 format zapisan u XML-u bez bilo kakvog gubitka MARC podataka (Banerjee i Reese Jr. 2019, 129). Osnovu MARCXML-a predstavlja jednostavna XML shema u kojoj se nalaze MARC podaci te koja zadržava semantiku MARC formata. U njoj se nalaze svi važni podaci za MARC pretvoreni te izraženi u XML formatu (The Library of Congress 2022a).

Iako MARCXML u potpunosti zadržava karakteristike MARC-a 21, iznimku čini kodna stranica. Naime, MARCXML kao obaveznu kodnu stranicu mora koristiti UTF-8, *Unicode* varijantu koju podržava XML, za razliku od MARC-a 21 koji koristi MARC-8 (Cole i Myung 2013, 81).

Razlog za pretvaranje MARC-a u XML leži u tome što se na taj način omogućava kompatibilnost i razmjena podataka s drugim shemama te različite mogućnosti prikaza zahvaljujući korištenju XSLT-a (Golub i Knok 2013, 26). MARCXML se tako najčešće koristi u knjižnicama, a služi kao alat za prijenos MARC podataka u druge metapodatkovne sheme (Banerjee i Reese Jr. 2019, 122). Kao što je već navedeno, MARCXML također služi za prikazivanje cijelog MARC zapisa u XML-u, a može se koristiti i kao nadopuna METS (*Metadata Encoding and Transmission Standard*) shemi (The Library of Congress 2022a). Radi se o standardu za kodiranje strukturnih, administrativnih i opisnih metapodataka koji su dio digitalne knjižnice, a izražavaju se uz pomoć XML sheme (The Library of Congress 2022c).

Također, MARCXML se može koristiti za prikazivanje metapodataka za inicijativu pod nazivom *Open Archives Initiative* (OAI), kao i za opisivanje metapodataka u XML-u koji dolaze kao dio elektroničkog izvora (The Library of Congress 2022a).

Pošto se MARCXML temelji na XML-u, njegovim korištenjem se izbjegavaju određena tehnička ograničenja koja su prisutna kod MARC-a 21. To se ponajprije odnosi na veličinu zapisa koja u MARCXML-u nije ograničena, što korisniku daje veću slobodu. Osim toga, može sadržavati slike i dokumente koji su direktno ugrađeni u metapodatke (Banerjee i Reese Jr. 2019, 122). Prednost je i u tome što se radi o shemi koja podržava sve MARC podatke bez obzira na to o kojem se formatu radi te što, zbog svoje proširive arhitekture, omogućava korisniku uključivanje i reprodukciju različitih softverskih dijelova kako bi prilagodili strukturu (The Library of Congress 2022a).

MARCXML također omogućuje novi način dijeljenja bibliografskih metapodataka, što prije nije bilo moguće korištenjem MARC-a. Naime, iako knjižnica posjeduje velike zbirke, ograničen broj aplikacija koje mogu koristiti metapodatke u MARC-u sprječava njihovu dostupnost široj zajednici. No, ako su ti metapodaci u MARCXML formatu, njihovo preuzimanje i transformacija je znatno lakša i brža te se samim time povećava pristup knjižničnim zbirkama. Osim toga, korištenje MARCXML u digitalnim zapisima olakšava transformaciju tih zapisa na webu, što dodatno povećava pristup svim korisnicima (Cole i Myung 2013, 81). Drugim riječima, XML je lako čitati, obrađivati i općenito strojno koristiti s mnogim suvremenim i iznimno važnim tehnologijama poput popularnih programskih jezika, dok je za podatke u MARC formatima to znatno otežano.

Uz to, MARCXML omogućava jednostavan način provjeravanja valjanosti bibliografskih zapisa korištenjem softverskih alata. Na taj način se omogućava osnovna provjera valjanosti XML-a prema MARCXML shemi, valjanosti MARC 21 oznaka te valjanosti sadržaja MARC zapisa kao što su vrijednosti, datumi i slično (The Library of Congress 2022a).

Ipak, iako je MARCXML donio napredak u odnosu na MARC 21 format, taj standard posjeduje i određene nedostatke zbog kojega se njegovo korištenje nije prihvatilo u potpunosti, već ga je prihvatio samo dio knjižničnih sustava. Kao vjerojatni razlog za to smatra se njegova odanost MARC-u kojeg mnogi smatraju arhaičnim standardom za opis građe. Osim toga, zahtjeva za time da MARCXML prikazuje MARC 21 podatke bez bilo kakvog gubitka doveo je do toga da i u MARCXML-u postoje određena tehnička ograničenja koja su prisutna i u MARC 21. Iako je prisutnost tih ograničenja u MARCXML-u manja nego u MARC-u 21, ta



ograničenja ipak postoje što se može navesti kao njegov nedostatak ((Banerjee i Reese Jr. 2019, 129). Uz to, nedostatkom se može smatrati i to što MARCXML provjere valjanosti ne provodi sama shema, nego vanjski softver (The Library of Congress 2022a).

## 8.2. MODS

Nakon objavljivanja MARCXML-a, Kongresna knjižnica je nastavila s razvijanjem novih metapodatkovnih shema te je 2002. godine objavila MODS, odnosno *Metadata Object Description Shema*-u. Riječ je o shemi koja je namijenjena opisu metapodataka za skup bibliografskih elemenata koji mogu biti korišteni u različite svrhe, a ponajviše za primjenu u knjižničnoj zajednici (The Library of Congress 2022b). Može ga se definirati kao jednostavniju verziju MARCXML-a koja je prilagođenija webu, kompatibilna je s MARC 21 formatom, ali je manje detaljna od MARCXML-a (Cole i Myung 2013, 110).

Razvoj MODS-a predstavlja sljedeći korak u evoluciji MARC formata u XML te se radi o puno jednostavnijoj alternativni koja i dalje ostaje kompatibilna s MARC-om 21, a osim toga predstavlja bogatiju alternativu ostalim metapodatkovnim shemama poput *Dublin Core*-a. Ipak, iako zadržava kompatibilnost s MARC 21 formatom, MODS-ova struktura omogućava reorganizaciju i ponovno grupiranje metapodatkovnih elemenata unutar samih zapisa metapodataka (Banerjee i Reese Jr. 2019, 129). Također, kao njegova važna karakteristika ističe se to što koristi jezične oznake, za razliku od MARC 21 koji koristi broježane oznake, što MODS čini jednostavnijom shemom za razmjenu i opis podataka. Kao i kod MARCXML-a, radi se o shemi zapisanoj u XML-u, a MODS verzija koja se trenutno koristi je verzija 3.7, izdana u veljači 2022. godine (The Library of Congress 2022b).

Također, iako su MARC i MODS povezani, MODS se ne može smatrati njegovom zamjenom, već se radi o dva zasebna standarda koja imaju preklapanja u semantici, ali nemaju potpuno iste funkcije. Bolje rečeno, MODS ne može napraviti sve ono što može MARC i obrnuto. Kao dobar primjer za to može se navesti već spomenuto korištenje jezičnih oznaka u MODS-u zbog čega su njegove oznake čitljive ljudima te su prikladnije za opis digitalnih izvora (Cole i Myung 2013, 111).

MODS može biti korišten u različite svrhe te se može primjenjivati u različitim područjima. Na primjer, može se koristiti kao SRU, odnosno *Search/Retrieve via URL*, a radi se o standardu koji koristi XML prilikom pretraživanja različitih upita. MODS služi i kao nadopuna metapodatkovne sheme METS. Također, može se koristiti za pojednostavljeno prikazivanje MARC zapisa u XML-u kao i za predstavljanje metapodataka u XML-u koji se

pojavljaju kao dio elektroničkog izvora (The Library of Congress 2022b). Osim toga, može biti korišten za stvaranje izvornih zapisa te olakšava prikupljanje različite knjižnične građe. Isto tako, njegova upotreba predstavlja namjeru Kongresne knjižnice za stvaranjem jednostavnijih i raznovrsnijih XML formata za rad s knjižničnim nasljeđem (Banerjee i Reese Jr. 2019, 131).

Postoje različite prednosti koje se vezuju za MODS i njegovo korištenje. Jednom od prednosti može se smatrati njegova fleksibilnost. To se prije svega odnosi na duljinu zapisa. Osim što je duljina zapisa fleksibilna, takvi zapisi se mogu lako izmijeniti. Osim toga, takvi zapisi se mogu lakše kodirati i zapisati u web aplikacijama i preglednicima, pošto MODS podržava obje verzije *Unicode*-a, odnosno verziju UTF-8 kao i UTF-16 (Cole i Myung 2013, 110).

Osim toga, korištenje MODS-a ne zahtjeva korištenje niti jednog specijalnog kataloškog pravilnika te se veći broj elemenata u MARC-u može spojiti u jedan element u MODS-u (The Library of Congress 2022b). To je omogućeno zahvaljujući povećanoj granuliranosti, odnosno fragmentarnosti elemenata koju MODS ima, a to omogućava njegovim zapisima da donose detaljnije i bogatije opise podataka od ostalih shema, poput *Dublin Core*-a i ostalih. Pošto MODS koristi širu skupinu elemenata od ostalih metapodatkovnih shema, njegovo korištenje korisnicima omogućava da opišu određenu ideju i elemente koji je sačinjavaju, što se također može smatrati kao prednost u odnosu na ostale sheme (Banerjee i Reese Jr. 2019, 131).

Međutim, iako njegovo korištenje donosi određene prednosti, ipak postoje nedostaci i ograničenja vezanih za ovaj standard. Tako prilikom pretvaranja bibliografskih podataka iz MARC 21 u MODS može doći do gubljenja dijela specifičnih podataka. Isto tako, prilikom ponovne pretvorbe podataka iz MODS-a u MARC može doći do ispuštanja dijela podataka te određeni dio podataka možda neće biti smješten u isto polje u kojem je bio na početku, prije pretvorbe (The Library of Congress 2022b).

Uz to, iako se MODS u određenoj mjeri udaljio od MARC 21 formata te se razlikuje od njega, i dalje je blisko povezan s tim formatom što ga vezuje samo za rad knjižničnih zajednica te se njegovo korištenje ne uspijeva proširiti i na ostala područja, unatoč tome što ga je Kongresna knjižnica željela promovirati kao opći metapodatkovni format (Banerjee i Reese Jr. 2019, 132).

### 8.3. XML u knjižnici

Kao što je već navedeno, XML ima različite uloge i funkcije, a kao jedna od važnijih može se navesti ona koju ima u radu knjižnica. Digitalizacija je postala sveprisutna pa tako nije zaobišla

ni knjižnice koje stalno povećavaju svoje digitalne zbirke u koje, osim knjiga, novina i ostalih publikacija danas ubrajaju formate poput slika, videa, videoigara i slično. XML u procesu digitalizacije zauzima važnu ulogu te je prikladan format za serijalizaciju, razmjenu i rukovođenje metapodacima u knjižnici te olakšava njihovu obradu i ponovnu upotrebu (Cole i Myung 2013, 349).

Također, korištenje metapodatkovnih shema koje se temelje na XML-u, poput MARCXML-a i MODS-a olakšavaju digitalizaciju. Njihovo korištenje smanjuje probleme s kojima se korisnici susreću prilikom stvaranja bibliografskih metapodataka. Razlog tome je što takve sheme razdvajaju administrativne (tehnički podaci o predmetu koji se uvrštava u digitalnu zbirku), strukturalne (podaci koji pohranjuju zapis svih predmeta) i bibliografske metapodatke (Banerjee i Reese Jr. 2019, 86).

Osim toga, kako bi knjižnice mogle kreirati metapodatke koji se mogu dijeliti potrebni su im formati prilagođeni webu, što je jedna od glavnih karakteristika XML-a. Njegovim korištenjem, metapodaci će biti prilagođeni webu, a korištenje XSLT-a će povećati i poboljšati njihovu interoperabilnost. Uz XSLT, važne su i ostale popratne tehnologije poput XML Shema i XML prostora imenovanja koje olakšavaju zapisivanje metapodataka koji uz izvorne vrijednosti mogu imati i vrijednosti koje je dodao korisnik, a da pri tome zadržavaju porijeklo svih vrijednosti metapodataka (Cole i Myung 2013, 353).

## **9. MARCXML I MODS – PRIMJERI**

Kao što je već rečeno, MARCXML i MODS su sheme koje se temelje na XML-u, a koriste se u knjižnicama za izgradnju digitalnih zbirki. To se može vidjeti u sljedećim primjerima.

```

<?xml version="1.0" encoding="UTF-8"?>
<record>
  <leader>00000cam0|2200000..|450|</leader>
  <controlfield tag="001">901228941</controlfield>
  <datafield tag="010" ind1=" " ind2=" " >
    <subfield code="a">9789532462654</subfield>
  </datafield>
  <datafield tag="100" ind1=" " ind2=" " >
    <subfield code="a">20200317d2016 mmy0hrva01010101ba</subfield>
  </datafield>
  <datafield tag="101" ind1=" " ind2=" " >
    <subfield code="a">hrv</subfield>
    <subfield code="c">eng</subfield>
  </datafield>
  <datafield tag="105" ind1=" " ind2=" " >
    <subfield code="a">b 001yy</subfield>
  </datafield>
  <datafield tag="135" ind1=" " ind2=" " >
    <subfield code="a">drcn nnnunnun</subfield>
  </datafield>
  <datafield tag="200" ind1=" " ind2=" " >
    <subfield code="a">Europa</subfield>
    <subfield code="b">Elektronička građa</subfield>
    <subfield code="e">borba za nadmoć od 1453. do danas</subfield>
    <subfield code="f">Brendan Simms</subfield>
    <subfield code="g">[prevoditelj Emil Heršak]</subfield>
  </datafield>
  <datafield tag="210" ind1=" " ind2=" " >
    <subfield code="a">Zagreb</subfield>
    <subfield code="c">Mate</subfield>
    <subfield code="d">[2016?]</subfield>
  </datafield>
  <datafield tag="215" ind1=" " ind2=" " >
    <subfield code="a">1 online jedinica građe</subfield>
    <subfield code="c">zemljop. crteži</subfield>
  </datafield>
  <datafield tag="225" ind1=" " ind2=" " >
    <subfield code="a">Biblioteka Međunarodni odnosi</subfield>
  </datafield>
  <datafield tag="304" ind1=" " ind2=" " >
    <subfield code="a">Stv. nasl. s naslovnice</subfield>
    <subfield code="a">Prijevod djela: Europe : the struggle for supremacy, 1453 to the present</subfield>
  </datafield>
  <datafield tag="307" ind1=" " ind2=" " >
    <subfield code="a">Tiskano izd. sadrži XXIV, 653 str.</subfield>
  </datafield>
  <datafield tag="320" ind1=" " ind2=" " >
    <subfield code="a">Bibliografija</subfield>
    <subfield code="a">Kazalo</subfield>
  </datafield>
  <datafield tag="327" ind1=" " ind2=" " >
    <subfield code="a">Borba za nadmoć ili za što? / Emil Heršak, Borna Jalšenjak</subfield>
  </datafield>
  <datafield tag="336" ind1=" " ind2=" " >
    <subfield code="a">Publikacija u formatu EPUB</subfield>
  </datafield>
  <datafield tag="337" ind1=" " ind2=" " >
    <subfield code="a">ZaKi Book</subfield>
  </datafield>
  <datafield tag="607" ind1=" " ind2=" " >
    <subfield code="3">Europa - povijest - 15/21. st.</subfield>
    <subfield code="3">Europa - politička povijest - 15/21. st.</subfield>
    <subfield code="3">Europa - međunarodni odnosi - 15/21. st.</subfield>
  </datafield>
  <datafield tag="675" ind1=" " ind2=" " >
    <subfield code="a">94(4)(0.034EPUB)</subfield>
    <subfield code="a">32(4)(0.034EPUB)</subfield>
    <subfield code="c">Povijest Europe. EPUB digitalni dokumenti</subfield>
    <subfield code="c">Politika u Europi. EPUB digitalni dokumenti</subfield>
  </datafield>
</record>

```

## Ispis 7. Primjer MARCXML zapisa

Iz prikazanog ispisa može se vidjeti kako su svi elementi koji se nalaze u MARC zapisu prisutni i u ovom zapisu, što potvrđuje tezu da MARCXML vjerno prikazuje MARC podatke bez ikakvog gubitka. Kao korijenski element u MARCXML zapisu nalazi se element <record>

koji ima elemente potomke <leader>, <controlfield> i <datafield>. Važno je naglasiti kako korijenski element <record> može imati samo jedan element potomak, <leader> te mora imati barem jedan element potomak <controlfield> i barem jedan <datafield>. Svaki od tih elemenata sadrži i odgovarajuće atribute. Iako korijenski element u ispisu ne sadržava atribute, on ih može imati, a radi se o atributima „id“ ili „type“. S druge strane, element <controlfield> sadržava samo jedan atribut „tag“, dok <datafield> sadržava tri atributa, odnosno „tag“, „ind1“ te „ind2“. Uz to, <datafield> mora sadržavati barem jedan element potomak <subfield>. Također, kao korijenski element, što se ne događa toliko često, može se pojaviti element <collection> koji može imati samo jedan element potomak, odnosno <record> (Cole i Myung 2013, 75,76).

```
<?xml version="1.0" encoding="UTF-8"?>
<mods xmlns="http://www.loc.gov/mods/v3" xmlns:zs="http://docs.oasis-open.org/ns/search-ws/sruResponse"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="3.7"
xsi:schemaLocation="http://www.loc.gov/mods/v3 http://www.loc.gov/standards/mods/v3/mods-3-7.xsd">
  <titleInfo>
    <title>Tin - biografija</title>
    <subTitle>trideset godina putovanja</subTitle>
  </titleInfo>
  <name type="personal" usage="primary">
    <namePart>Boko, Jasen</namePart>
  </name>
  <typeOfResource>text</typeOfResource>
  <originInfo>
    <place>
      <placeTerm type="text">Zagreb</placeTerm>
    </place>
    <publisher>Profil knjiga</publisher>
    <dateIssued>2017</dateIssued>
  </originInfo>
  <language>
    <languageTerm authority="iso639-2b" type="code">hrv</languageTerm>
  </language>
  <physicalDescription>
    <form authority="marcform">print</form>
    <extent>256 str., [8] str. s tablama ; 23 cm</extent>
  </physicalDescription>
  <note type="statement of responsibility">0 autoru: str. [255]-256</note>
  <subject authority="lcsht">
    <name>Ujević, Tin</name>
    <temporal>1891-1955</temporal>
    <genre>biografije</genre>
  </subject>
  <classification authority="udc">821.163.42-94 UJEVIĆ</classification>
  <identifier type="isbn">978-953-313-597-7</identifier>
  <recordInfo>
    <descriptionStandard>aacr</descriptionStandard>
    <recordCreationDate encoding="marc">220817</recordCreationDate>
    <recordChangeDate encoding="iso8601">20060801143536.0</recordChangeDate>
    <recordIdentifier>11761548</recordIdentifier>
    <recordOrigin>Converted from MARCXML to MODS version 3.7 using MARC21slim2MODS3-7.xsl
      (Revision 1.140 20200717)</recordOrigin>
  </recordInfo>
</mods>
```

### Ispis 8. Primjer MODS zapisa

Kako je vidljivo u navedenom ispisu, korijenski element u MODS zapisu je element <mods> ispod kojega se može nalaziti niz elemenata poput „titleInfo“, „name“, „origin“,

„place“ i slično. Također, MODS zapis može sadržavati niz atributa i pod elemenata. Osim toga, ako se radi o jednom MODS dokumentu koji sadrži više MODS zapisa, kao korijenski element može se pojaviti <modsCollection>, koji za element potomak ima <mods>. MODS ima hijerarhijsku strukturu, što je jedan od razloga zbog kojega takvi zapisi nude specifičnije informacije (Cole i Myung 2013, 110).

Jedna od najvećih razlika između ova dva standarda je u tome što MODS koristi prirodni jezik za svoje oznake, dok MARCXML koristi brojeve oznake, kao što ih koristi i sam MARC. Zbog korištenje jezičnih oznaka, MODS zapisi su čitljiviji od MARCXML-a, što predstavlja veliku prednost i olakšava njegovo korištenje (Banerjee i Reese Jr. 2019, 130). No, važno je napomenuti kako su ti zapisi vezani za engleski jezik. Osim toga, MODS-ova prednost leži u njegovoj jednostavnosti te u tome što se radi o standardu koji se može primjenjivati i na građu koja nije vezana za knjižnicu, dok je MARCXML vjeran prikaz MARC podataka zbog čega ostaje samo u domeni knjižnice (Golub i Knok 2013, 26). Također, MODS je prilagođeniji webu od MARCXML-a, što se smatra velikom prednošću (The Library of Congress 2022b).

Ipak, iako se može primjenjivati i u drugim područjima, MODS se nije uspio rasprostraniti dovoljno te se i njegovo korištenje uglavnom svodi na knjižnice kao i korištenje MARCXML-a. Osim toga, kao što je već rečeno, MARCXML vjerno prikazuje sve MARC podatke, bez ikakvog gubitka, dok su kod MODS-a mogući gubitci određenih specifičnih podataka što se može smatrati njegovom negativnom stranom u odnosu na MARCXML (The Library of Congress 2022b).

Iako i jedan i drugi standard imaju određene mane, ipak se radi o standardima koji su donijeli promjenu i omogućili novi način stvaranja, razmjene i korištenja bibliografskih metapodataka, a samim time olakšali proces digitalizacije građe.

Kao zaključak proučavanja i istraživanja standarda MARCXML i MODS, u nastavku je prikazana tablica u kojoj su navedene njihove osnovne karakteristike, sličnosti i razlike te prednosti i mane.

|   | <b>MARCXML</b>   | <b>MODS</b>   |
|---|--|---|
| <b>Ljudski čitljive oznake</b>                                      | po uzoru na MARC koristi brojevne oznake što otežava njegovo korištenje i čitljivost ljudima   | koristi prirodni jezik, jezične oznake koje su čitljive ljudima i prikladnije za opis digitalnih izvora   |
| <b>Bazirano na XML tehnologiji</b>                                  | temelji se na XML-u, predstavlja MARC 21 format zapisan u XML-u, njegovu osnovu predstavlja XML shema koja sadrži podatke i semantiku MARC formata, koristi XML-ovu kodnu stranicu (UTF-8) | shema zapisana u XML-u, predstavlja jednostavniju verziju MARCXML-a i sljedeći korak u evoluciji MARC formata u XML, koristi XML-ove kodne stranice (UTF-8 i UTF-16)  |
| <b>Bazirano na postojećim knjižničnim standardima za opis građe</b> | inačica MARC 21 koja zadržava sve njegove karakteristike i sve važne podatke za MARC, iznimku predstavlja kodna stranica (MARC koristi MARC-8, a MARCXML UTF-8)                            | jednostavnija alternativa koja ostaje kompatibilna s MARC 21, ali ne može se smatrati njegovom zamjenom (nemaju potpuno iste funkcije, koriste različite oznake)  |
| <b>Područja primjene</b>  | predstavlja vjeran prikaz MARC podataka zbog čega je u potpunosti vezan za knjižničnu domenu   | predstavljen kao opći metapodatkovni format, može se primjenjivati i u ostalim područjima i na građu koja se ne nalazi u knjižnici, ipak nije dovoljno rasprostranjen u ostalim područjima, najčešće ostaje samo u domeni knjižnice |

|                                  |   |   |
|----------------------------------|---|---|
| <p><b>Prilagodенost webu</b></p> | <p>koristi se na webu, olakšava transformaciju digitalnih zapisa na webu i tako povećava pristup korisnicima, ipak, pošto vjerno slijedi MARC 21 posjeduje tehnička ograničenja zbog kojih je njegova prilagodенost webu smanjena</p> | <p>udaljava se od MARC 21, zbog toga je fleksibilnija i jednostavnija alternativa prilagođenija korištenju na webu, podržava i UTF-8 i UTF-16 zbog čega omogućava lakše zapisivanje i kodiranje zapisa u web aplikacijama i preglednicima</p> |
|----------------------------------|---|---|

Tablica 1. Usporedba MARCXML-a i MODS-a



## 10. ZAKLJUČAK

XML (*eXtensible Markup Language*), kojeg je 90-ih godina prošlog stoljeća razvila organizacija *World Wide Web Consortium*, u današnje vrijeme se istaknuo kao jedan od najvažnijih označiteljskih jezika koji omogućava bržu razmjenu podataka između različitih aplikacija putem Interneta i WWW-a. Njegov razvoj je od velike važnosti jer taj jezik opisuje podatke koji se razmjenjuju te time olakšava razmjenu podataka i ubrzava rad programera. Njegova je važnost i u tome što olakšava prijenos, dijeljenje i dostupnost podataka kao i promjene platformi. Najviše se koristi za programsku obradu, ali koristi riječi iz prirodnoga jezika koje su razumljive ljudima pa je tako čitljiv svima, i strojevima i ljudima, što predstavlja još jednu prednost ovog jezika.

Njegova primjena je široka, koristi se u različitim područjima, a ne može se zanemariti uloga koju u današnje vrijeme ima u radu knjižnica. Njegova uloga je važna u procesu digitalizacije građe koja je u knjižnicama postala sve češća. Knjižnice konstantno povećavaju svoje digitalne zbirke kako bi se građa sačuvala, ali kako bi postala dostupnija svim korisnicima. Uloga XML u tom procesu je značajna pošto se koristi za serijalizaciju, razmjenu i upravljanje metapodacima, ali i olakšava obradu metapodataka i njihovu ponovnu upotrebu. Osim toga, pospješuje razmjenu i čitljivost bibliografskih metapodataka, a istovremeno zadržava mogućnost korištenja stručnih knjižničarskih standarda i načela.

Važnost za rad knjižnice predstavljaju i metapodatkovni standardi koji se temelje na XML-u, a jedni od glavnih su MARCXML i MODS. Radi se o standardima koji se koriste prvenstveno u radu knjižnica, a razvila ih je Kongresna knjižnica. MARCXML predstavlja standard koji se temelji na XML-u te predstavlja MARC 21 format zapisan u XML-u. U njemu se nalaze MARC podaci koji su pretvoreni i izraženi u XML formatu, a bez ikakvog gubitka MARC podataka. S druge strane, MODS predstavlja jednostavniju verziju MARCXML-a koja je prikladnija webu te čitljiva ljudima. Također, predstavlja bogatiju alternativu ostalim metapodatkovnim shemama, a istovremeno ostaje kompatibilna s MARC-om 21.

## 11. LITERATURA

Banerjee, Kyle, Terry Reese jr. 2019. *Building digital libraries; A how to do it manual for librarians*. Chicago: American Library Association.

Bosančić, Boris. 2011. "Uloga opisnih označiteljskih jezika u razvoju digitalne humanistike." *Libellarium* 4, br. 1, 65-82. <https://hrcak.srce.hr/92394>

Cole, W. Timothy, Myung-Ja K. Han. 2013. *XML for catalogers and metadata librarians*. Santa Barbara: Libraries Unlimited.

Fawcett, Joe, Liam R.E. Quin, Danny Ayers. 2012. *Beginning XML*. Indianapolis: John Wiley & Sons, Inc.

Golub, Koraljka i Željko Knok. 2013. "Mogućnosti primjene XML-a na sustave za predmetno označavanje." *Zbornik radova Međimurskog veleučilišta u Čakovcu* 4, br. 1, 21-27.

<https://hrcak.srce.hr/104440>

Kirasić, D. 2005. "XML tehnologija i primjena u sustavima procesne informatike." U *Proceedings of the 28th International Convention MIPRO 2005*. Uredili J. Kljaić i N. Baranović. 30. svibanj - 03.lipanj 2005, Opatija, Hrvatska.

[https://www.fer.unizg.hr/download/repository/mipro\\_xml\\_tekst.pdf](https://www.fer.unizg.hr/download/repository/mipro_xml_tekst.pdf)

Maleš, Lada, Mladenović, Saša. 2007. *Osnove programiranja za web (HTML, JavaScript, XML i XLS)*. Split: Filozofski fakultet u Splitu.

Microsoft 365. n.d. "XML za početnike." Pristupljeno: 17. srpnja 2022.

<https://support.microsoft.com/hr-hr/office/xml-za-po%C4%8Detnike-a87d234d-4c2e-4409-9cbc-45e4eb857d44>

The Library of Congress. 2022a. "MARC 21 XML Schema." Pristupljeno: 11. kolovoz 2022.

<https://www.loc.gov/standards/marcxml/>

The Library of Congress. 2022b. "Metadata Object Description Schema." Pristupljeno: 12.

kolovoz 2022. <https://www.loc.gov/standards/mods/>

The Library of Congress. 2022c. "Metadata Encoding and Transmission Standard."

Pristupljeno: 12. kolovoz 2022. <http://www.loc.gov/standards/mets/>

W3Schools. n.d. "Introduction to XML." Pristupljeno: 17.srpnja 2022.

[https://www.w3schools.com/xml/xml\\_what\\_is.asp](https://www.w3schools.com/xml/xml_what_is.asp)

## **XML and its application in MARCXML and MODS standards**

### **Abstract**

The aim of this paper is to describe XML as one of the most important markup languages, to explain its application and to emphasize the importance of using XML in libraries in the modern information environment. The first part of the paper describes the markup language XML, its history and a brief overview of the languages that preceded it, the basic concepts and syntax of XML, its technologies, and applications with corresponding examples. Also, comparison was made with other markup languages, with emphasis on HTML. After the description of XML, there is a description of two selected standards written in the markup language XML, i.e., MARCXML and MODS, which are mostly used in libraries when digitizing library materials. First is described MARCXML, its design, schema, and implementation. Then is described MODS and its scheme, application, and design. In addition, the paper emphasizes the role that the application of XML has today in the library, that is, in the process of digitization of library materials where XML plays an important role as a suitable format for serialization, exchange and management of metadata in the library. The description of the MARCXML and MODS standards is later followed by their practical examples and their comparison as well. There are also listed the advantages and positive sides of their use as well as the disadvantages that users encounter when using these standards.

Keywords: markup languages, XML, MARCXML, MODS, metadata

## **PRILOZI**

Ispis 1. Primjer zapisa u XML-u

Ispis 2. Informacije o sastanku zapisane u XML-u

Ispis 3. Primjer DTD-a u XML dokumentu

Ispis 4. Opis videoigre zapisan u XML-u

Ispis 5. Račun iz trgovine zapisan u XML-u

Ispis 6. Primjer XML Sheme

Ispis 7. Primjer MARCXML zapisa

Ispis 8. Primjer MODS zapisa

Tablica 1. Usporedba MARCXML-a i MODS-a