Elektronske znanstvenokritične izdaje slovenskega slovstva: standardi in izzivi

Tomaž Erjavec, Ljubljana

Projekt Elektronske znanstvenokritične izdaje slovenskega slovstva je po tehnični plati zaznamovan z uporabo odprtih računalniških standardov, ki naj bi rezultatom projekta prinesli jasnost, prenosljivost in trajnost. V prispevku najprej predstavimo standarde za digitalni zapis besedil, ki jih uporabljamo v projektu, predvsem XML in TEI. Nato opišemo metodologijo pretvorbe izdaj iz izvorne oblike v kanonični standardizirani zapis in iz njega v obliko, primerno za branje in pregledovanje. Obravnavamo še izzive, ki so pred nami, predvsem uvajanje jezikovnih tehnologij v izdelavo digitalnih znanstvenokritičnih izdaj.

1 Uvod

Praktična težava pri objavljanju znanstvenokritičnih izdaj v tradicionalni, tiskani obliki je, da so proizvodni stroški zelo visoki, bralska publika takšnih izdaj pa majhna, še zlasti v Sloveniji. Elektronska znanstvenokritična izdaja ponuja odlično rešitev, ne le za ekonomski problem, ampak razreši edicijo še mnogih drugih omejitev: brez težav je mogoče vključiti digitalizirani faksimile, brez prostorskih omejitev je mogoče vključiti v izdajo diplomatični in kritični prepis ter morebitna variantna besedila, nobenih omejitev ni glede obsega kritičnega aparata. In kar je največja prednost: z analitičnimi orodji je mogoče iskati po besedilu na različne načine, kar pri tiskanih izdajah pomeni precejšnjo izgubo časa.

Praksa objavljanja besedil na medmrežju pa je pokazala tudi šibko stran elektronskih edicij: brez standardiziranega kodiranja se trajnost ter izmenljivost elektronskih besedil bistveno zmanjšata. Razvoj elektronskega objavljanja besedil je zato naravnan k izdelavi in uporabi standardiziranega kodiranja in označevanja besedil (*mark-up*), ki naj nevtralizira vpliv spreminjajoče se strojne in programske opreme na elektronske tekste (Erjavec, 2003).

V projektu Elektronske znanstvenokritične izdaje slovenskega slovstva dosledno upoštevamo mednarodne standarde kodiranja, na sam proces produkcije

digitalnih izdaj pa poleg tega vpliva tudi sestava projektne skupine, ki vsebuje raziskovalce tako z Inštituta za slovensko literaturo ZRC SAZU kot z Odseka za tehnologije znanja IJS. Prvi imajo znanje in izkušnje v tekstni kritiki, ki pa je bila izvajana na klasičen (tj. nedigitalen) način, kjer služi računalnik predvsem kot urejevalnik besedil. IJS pa ima za projekt koristne izkušnje v razvoju in uporabi jezikovnih tehnologij, predvsem izdelavi označenih besedilnih korpusov (sodobnega) slovenskega jezika. V dosedanjem delu na projektu (Erjavec, Ogrin, 2004; Slomšek et al., 2004) smo zato posvečali pozornost predvsem vzpostavitvi medsebojnega sodelovanja in prenosu izkušenj in metod, kjer je bil cilj izdelati sicer standardizirane digitalne izdaje, ki pa v mnogočem še vedno samo preslikujejo svoj papirni izvor. V naslednjih letih se bo poudarek prenesel k uporabi in nadaljnji obdelavi materialov – ena od bolj zanimivih smeri raziskav je jezikoslovna obravnava besedil, ki je usmerjena v študij leksike materialov.

V nadaljevanju prispevka predstavimo v 2. razdelku standarde digitalnega zapisa besedil, s poudarkom na znanstvenokritičnih izdajah, v 3. metodologijo izdelave standardiziranih digitalnih izdaj v projektu, v 4. pa metodologijo jezikoslovnega označevanja besed in možne uporabe. V 5. razdelku zaključimo s širšo vizijo nadaljnjega dela.

2 Standardi digitalnega zapisa

Precej zgodaj so se že začele pojavljati pobude za standardizacijo zapisa besedil, kjer se trudijo predpisati javno dostopne in trajne načine označevanja. Tako zapisana besedila potem uporabljajo orodja, ki implementirajo te standarde, bodisi neposredno ali pa tako, da lahko vanje podatke uvažajo oz. izvažajo. Najbolj vplivna pobuda s tega področja je bil leta 1986 izdani ISO standard SGML (Standard Generalised Markup Language), ki določa jezik za predstavitev dokumentov, nad katerimi bodo delovali programi za procesiranje besedil. Ta standard naj bi zagotovil način zapisa, ki je prenosljiv med računalniškimi platformami, odporen na tehnološke spremembe in omogoča uporabo dokumentov v različne namene. Velika prednost SGML pred drugimi zapisi je tudi, da je mogoče avtomatsko preveriti, ali je neki dokument zapisan v skladu s standardom.

Standard SGML je bil dobro premišljen, vendar pa zelo kompleksen, zato je svojo nišo našel predvsem v podjetjih, ki posedujejo velike količine dragocenih besedil in so bila pripravljena v standardizacijo njihovega zapisa investirati veliko

denarja. Zaradi kompleksnosti je bilo na voljo razmeroma malo programov, ki so implementirali standard; ti so bodisi delovali na platformi Unix, ki se je uporabljala predvsem v akademskih krogih, ali pa so bili izredno dragi. Zaradi teh problemov SGML nikoli ni dosegel zares široke popularnosti.

S prodorom svetovnega spleta pa se je pojavila potreba, da se mrežnim aplikacijam zagotovi standardizirani in prenosljivi način zapisa podatkov; to ni mogel biti HTML, saj ima ta zelo omejene izrazne možnosti, zaradi svoje kompleksnosti pa tudi ne SGML. Rešitev, ki jo je izdelal konzorcij za svetovni splet W3C (Word Wide Web Consortium), je bila podmnožica SGML, ki ohranja dobre lastnosti standarda, vendar pa izpusti elemente, ki so vnašali pretirano kompleksnost. Ta izvedenka SGML se imenuje XML (eXtended Markup Language); prva različica specifikacije je bila objavljena leta 1998, druga, trenutno zadnja, ki je popravila nekatere napake iz prve, pa leta 2000.

Za razliko od SGML je XML postal izjemno popularen in dejansko postaja univerzalni medij zapisa ne samo besedil, temveč tudi drugih podatkov, in to ne samo kot način hranjenja teh podatkov, temveč tudi za neposredno izmenjavo med različnimi programi. K njegovi odmevnosti je pripomoglo tudi veliko število prosto dostopnih programov, ki standard implementirajo, kot tudi obilica pridruženih standardov, ki ga dodatno osmislijo.

V izdelavo digitalnih edicij besedil za namene znanstvenih raziskav je potrebno vložiti veliko dela, potencialno pa so nato uporabna v raznovrstne namene, zato so se tudi tu že zgodaj pojavile pobude za standardizacijo zapisa. Največ je na tem področju naredila iniciativa za zapis besedil TEI (Text Encoding Initiative) s svojimi Priporočili, ki definirajo konkretne oznake za opis besedil, namenjenih znanstveni obravnavi; TEI je najprej temeljil na SGML, trenutno zadnja različica (Sperberg-McQueen, Burnard, 2002), pa podpira tudi XML.

V nadaljevanju tega poglavja predstavimo osnove XML in razložimo strukturo priporočil TEI.

2.1 Strukture XML

Standard XML formalno definira računalniški zapis besedila in uvede načine, kako lahko to besedilo označimo in strukturiramo. Na sliki 1 vidimo primer dokumenta XML; kot prvo lastnost zapisa je pomembno izpostaviti, da je dokument berljiv

tudi neposredno v obliki XML. Čeprav ni mišljeno, da bi dokumente XML brali »surove«, pa je vseeno koristno, da jih lahko tudi brez posebnih orodij beremo in popravljamo.

Dokument XML vsebuje elemente, od katerih se vsak začne z začetno oznako, npr. `<l>`, in zaključi s končno, ki vsebuje poleg puščičastih oklepajev in imena elementa še poševnico. Element XML je tako sestavljen iz treh delov: obeh oznak in vsebine. Izjema so prazni elementi (primer je `<pb/>` spodaj, ki naj bi označeval prelom strani), ki imajo za razliko od tistih z vsebino samo eno oznako, ta pa se končuje s poševnico.

Elementi lahko vsebujejo besedilo ali pa spet druge elemente oz. mešanico obojega, celoten dokument pa mora vsebovati natanko en vrhnji element; na sliki 1 je to `<div>`. Dokumenti XML so tako strukture z dobro poznanimi formalnimi lastnostmi, t. i. drevesa.

```
<div n="14" id="jenko.14">
 <head>Uvod.</head>
 <lg>
  <l>Dvigni se! ukaz mi reče.</l>
  <l>Srce pade mi v oblasti</l>
  <l>Silne, prej neznane strasti,</l>
  <l>Ki ko živi ogenj peče.</l>
 </lg>
 <pb/>
 <lg>
  <l>Čut se zlije mi v besede. -</l>
  <l>Preč so črne bolečine,</l>
  <l>Strast občutkov divjih mine,</l>
  <l>Jasen mir se v prsi vsede.</l>
 </lg>
</div>
```

Slika 1: Primer dokumenta XML

Drevesni model označevanja je intuitiven in enostaven za računalniško obdelavo, ni pa zadosten za vse vrste struktur. Predvsem je problematično, kadar bi dokument radi kategorizirali v več nepovezanih hierarhij, npr. če hočemo zajeti tipografsko in retorično strukturo besedila, saj so navzkrižna gnezdenja, npr. `<div>` ... `<page>` ... `</div>` ... `</page>`, prepovedana.

Obstaja več načinov, kako obidemo takšne probleme; enega smo že videli na sliki 1, kjer je `<pb/>` zapisan kot prazen element, ki tako označuje prelom, namesto da bi stran vseboval. Splošni način, kako se izogniti omejitvam neposrednega XML označevanja, pa je uvedba referenčnih mehanizmov. V modelu posrednega označevanja (*stand-off annotation*) je originalno besedilo nedotaknjeno, oznake pa se nahajajo v ločenih dokumentih in na besedilo samo kažejo (Thompson, McKelvie, 1997, Durusau, O'Donnell, 2001). V praksi se izkaže, da daje najboljše rezultate mešanica obeh pristopov, saj je bolj splošen model posrednega označevanja ustrezno kompleksnejši, težje preverljiv in zato okoren predvsem za gradnjo virov.

Poleg samih elementov definira XML tudi sredstvo za izražanje njihovih lastnosti, skozi t. i. atribute, ki jih lahko vsebujejo začetne značke elementov; kot vidimo na sliki 1, sledita imenu atributa (`n` in `id`) enačaj in vrednost atributa v navednicah.

Kot zadnjo lastnost dokumentov XML moramo omeniti nabore znakov in entitete za njihov zapis. Če ni drugače določeno, se privzame, da je dokument XML zapisan v skladu s standardom Unikod (Unicode), ki vsebuje večino svetovnih pismenk. Vendar pa je ta zapis več kot 8-biten in ga je dostikrat iz tehničnih in zgodovinskih razlogov še vedno potrebno prevesti v manjše in manj splošne nabore.

XML definira splošno uporaben način za zamenjavo nizov, tj. enostavno in strojno neodvisno metodo, ki omogoča določiti, da se mora pri obdelavi določen niz znakov v dokumentu XML zamenjati z nekim drugim nizom. Nize, definirane s to metodo za nadomeščanje, imenujemo entitete. K temu pojmu se še povrnemo kasneje, tu pa le omenimo, da entitete omogočijo prenos poljubnih znakov v naboru ASCII: med znaka za začetek (`&`) in konec (`;`) entitete pri tem napišemo kodo znaka iz kodnega nabora Unikod, npr. `kr&#x161;&#x10D;ansko`, kjer `#` pomeni, da sledi numerična koda, `x` pa, da je zapisana v šestnajstiškem zapisu; `161` in `10D` sta v naboru znakov Unikod kodi za znaka š in č.

Nekaj bolj uporabnih znakov pa je v XML tudi vnaprej definiranih z mnemoniki; od teh sta najpomembnejša `&` za `&` in `<` za `<`, saj ju je, kadar sta del besedila, potrebno zapisati z entitetami.

2.2 Definicija tipa dokumentov XML

Če bi dokumenti XML lahko vsebovali kakršne koli elemente v poljubnih medsebojnih odnosih in za elemente ne bi vedeli, kaj naj sploh pomenijo, ti dokumenti ne bi bili preveč uporabni. Seveda je možno, da je znanje o nekem tipu dokumentov vgrajeno v

samo aplikacijo; primer tega so spletni brkljalniki, saj na zaslon izpišejo zapis HTML, četudi jim ne podamo formalne specifikacije oznak, ki jih HTML uporablja.

Vendar je izredno koristno imeti možnost, da lahko formalno definiramo nabor elementov za določen tip dokumentov; pri SGML je celo veljalo, da mora vsak dokument vsebovati ali se vsaj sklicevati na tako formalno specifikacijo, t. i. definicijo tipa dokumentov (Document Type Definition, DTD). Ravno zaradi lažje komunikacije med procesi to pri XML ni potrebno, je pa možno. XML je od SGML podedoval (poenostavljen) mehanizem DTD-jev, ki uporablja poseben jezik, da definira skladnjo za elemente določenega tipa dokumentov.

Dokumenti XML, ki vsebujejo ali se sklicujejo na DTD, so pravilni (*valid*), tisti, ki pa se ne, vendar so vseeno zapisani po pravilih XML, pa so dobro oblikovani (*well-formed*); primer dobro oblikovanega dokumenta XML je bil podan na sliki 1.

DTD-ji omogočajo definicijo še raznih drugih gnezdenj elementov, vrednosti atributov, vsebujejo pa tudi mehanizme za modularizacijo, ki pa jih tu ne bomo obravnavali. Vseeno pa je treba omeniti entitete, ki smo jih že uvedli v prejšnjem razdelku; DTD namreč omogoča definicijo entitet kot poljubnih nizov ali kar celotnih datotek, ki jih nato lahko uporabimo v pravilnem dokumentu XML.

Pravilen dokument XML vsebuje DTD ali pa se nanj sklicuje s posebnim ukazom DOCTYPE, ki mora biti prvi v dokumentu. Če bi se denimo dokument na sliki 1 začel z vrstico `<!DOCTYPE div SYSTEM 'pesem.dtd'>` in bi datoteka pesem.dtd vsebovala DTD s slike 2, bi bil ta dokument pravilen, in ne samo dobro zapisan. Možnost preverjanja pravilnosti dokumentov XML je obenem tudi velika prednost tega formata pred večino ostalih, saj lahko s programi, t.i. razčlenjevalniki, ki preverjajo pravilnost (*validating parser*) ugotovimo, ali neki dokument res formalno ustreza svojemu tipu dokumentov.

```
<!ELEMENT  div      (head?, (lg | pb)+)  >
<!ELEMENT  head     (#PCDATA)            >
<!ELEMENT  lg       (#PCDATA | l)*       >
<!ELEMENT  l        (#PCDATA)            >
<!ELEMENT  pb       (EMPTY)              >

<!ATTLIST div
           n     CDATA  REQUIRED
           id    ID     #IMPLIED  >
```

Slika 2: Primer DTD XML

2.3 Pridruženi standardi

Uspešnost XML lahko pripišemo tudi obstoju pridruženih standardov, ki XML dodatno osmislijo, in prosto dostopnim orodjem, ki te standarde implementirajo; v tem poglavju omenimo samo nekatere najpomembnejše.

XML je prvenstveno jezik za opisovanje lastnosti, in ne videza dokumenta. To je v splošnem izredno dobra lastnost, saj je za uporabnost dokumentov precej bolj pomembno, kaj določen del besedila pomeni in ne kakšen je njegov zunanji videz. Kljub temu pa je neki dokument prej ko slej potrebno prikazati bodisi na zaslonu ali pa na papirju. Jezik XSLT definira način, kako preoblikovati dokumente XML v druge formate, bodisi XML ali drugačne, ki vsebujejo formate za stavljenje, vendar pa nanje niso omejeni. XSLT je tako uporaben ne samo kot prikazovalnik, pač pa tudi kot splošno orodje za konverzijo iz nekega XML formata (DTD) v poljuben format, ki ga potrebuje določena aplikacija, ki naj bi dokument procesirala. Posebej privlačna je konverzija iz enega XML DTD v drugega, saj odpira možnost poljubnega preoblikovanja, selekcije in združevanja dokumentov XML. Pri tem pa je jezik XSLT tudi sam zapisan v XML, s čimer lahko uporabimo orodja XML za pisanje transformacij XSLT, obstajajo pa tudi prosto dostopni programi, ki implementirajo konverzije XSLT.

Kot je bilo že omenjeno, mehanizem DTD omogoča preverjanje pravilnosti dokumentov, kar je izredno koristno pri izdelavi in označevanju novih jezikovnih virov. Vendar pa imajo DTD-ji tudi vrsto pomanjkljivosti, predvsem to, da so zapisani v svojem jeziku, in ne v XML, ter da podpirajo samo relativno enostavne podatkovne modele. V DTD npr. ni načina, da bi določili, da mora biti vsebina nekega elementa število ali da je vrednost atributa datum. Zato je bilo razvitih več shem, ki presežejo te omejitve – tu naj omenimo samo dve. Prva je RELAX NG, ki je postala standard ISO in je osnovana na regularnih izrazih, druga pa XML Schema, ki je predlog konzorcija W3C. Shemi sta komplementarni, saj ima vsaka svoje prednosti. Obstajajo tudi orodja, ki implementirajo preverjanje skladnosti dokumentov XML po specifikaciji v teh shemah in prevajanje DTD-jev v izraze iz teh dveh shem.

2.4 Iniciativa za zapis besedil TEI

Iniciativa za zapis besedil TEI (Text Encoding Initiative) je bila ustanovljena leta 1987 pod pokroviteljstvom več mednarodnih združenj, nastala pa je z namenom,

da se standardizira zapis besedil, ki bi se uporabljala pretežno v znanstvene namene, oz. da se razvije skupni način označevanja kompleksnih struktur besedil. S tem bi se zmanjšala razdrobljenost obstoječih načinov digitalnega zapisa, poenostavila računalniška obdelava in spodbudilo razširjanje ter izmenjevanje elektronskih besedil. Kmalu pa je postalo očitno, da bo zadosti prožna shema zapisa lahko nudila rešitve za splošne probleme zapisa besedil.

TEI je prvi osnutek svojih priporočil (TEI P1) izdal leta 1990, drugega pa leta 1992. Medtem ko sta bila tako P1 kot P2 še osnutka, predstavlja leta 1994 izdan TEI P3 zaključek prve faze dela TEI. TEI je kot osnovo svojega zapisa vzel SGML. TEI P3 je nabor fragmentov definicij tipov dokumentov, ki za široko paleto zvrsti besedil določa konkretne oznake in njihovo strukturo. Skorajda bolj pomembnih pa je 1200 strani dokumentacije, ki podaja pomen posameznih oznak, opisuje posamezne podsklope ter izpelje način za njihovo kombiniranje ter nadgradnjo. TEI P3 kot njegovi nasledniki so sicer izšli v knjižni obliki, so pa tudi prosto dostopni prek svetovnega spleta. Leta 1999 je izšla popravljena izdaja TEI P3, ki je odpravila nekaj tipografskih in drugih napak, 2002 pa je bil izdan TEI P4 (Sperberg-McQueen in Burnard 2002), ki nudi enakovredno podporo za SGML in XML ter in popravi razne napake iz P3, pri čemer pa ohranja skladnost s TEI P3.

Leta 2000 je bil ustanovljen konzorcij TEI, <http://www.tei-c.org/>, ki naj bi skrbel za razvoj standarda TEI. V okviru konzorcija je bil tako izdan TEI P4, ob tem pa so bila identificirana področja, kjer bi bil TEI potreben temeljitejše prenove. Zato je bilo ustanovljenih več delovnih skupin, ki imajo nalogo revidirati P4, da bo lahko leta 2005 izdana nova različica TEI P5. Ta ne bo več ohranjala skladnosti s TEI P3 oz. P4, namesto DTD-jev bo uporabljala Relax NG, omogočala pa bo tudi precej boljšo parametrizacijo kot prejšnje inačice.

Več kot 120 projektov, ki pokrivajo prek 30 jezikov, je do sedaj uporabljalo priporočila TEI za zapis raznovrstnih virov, npr. za jezikovne korpuse, za slovarje, knjižnične kataloge ipd. TEI je bil vpliven tudi pri zapisu srednjeveških rokopisov in tekstnokritičnih izdaj; tako na mrežnih straneh TEI najdemo opise projektov, ki so ustvarili digitalne zapise Dantejevih del, Canteburijskih zgodb, starobolgarske (starocerkvenoslovanske) literature, srednjeveške nordijske literature in še mnogo drugih.

S TEI P3 ali P4 skladen DTD ustvarimo za potrebe konkretnega projekta s kombinacijo naborov oznak, ki jih definira TEI. Središčne oznake (*core tags*) so ob-

vezne v vsakem TEI DTD. Središčne oznake tako določajo elemente, ki so na voljo v vseh dokumentih TEI (npr. oznake za naslove in odstavke), ter glavo dokumenta, ki vsebuje bibliografske in druge podatke o dokumentu. Osnovni nabori oznak (*base tag sets*) opisujejo različne zvrsti besedil, ki so med seboj razmeroma dobro ločene; vsak DTD lahko vsebuje natanko en osnovni nabor znakov. TEI definira osnovne nabore za prozo, poezijo, gledališče, zapis govora, tiskane slovarje ter terminološke baze. Dodatni nabori oznak (*additional tag sets*) zajemajo raznovrstna dodatna označevanja, ki predstavljajo določeno interpretacijo besedila ali pa nebesedilne elemente besedil, kot so navzkrižne povezave (npr. za stvarna kazala) ali pa slike. Takih naborov je vsega skupaj devet, med njimi so nabor za analitične mehanizme, npr. skladenjsko analizo, nabor za dokumentiranje uredniških posegov, nabor za imena in datume in tudi nabor za opis primarnih virov in tekstnokritični nabor. Končno lahko v TEI definiramo uporabniško določene oznake (*user defined tagset*), kjer dodamo lastne oznake ali spremenimo oznake, ki jih definira TEI.

Kot primer je na sliki 3 podana preambula XML, ki parametrizira TEI za gradivo pri našem projektu; ta najprej definira, da je korenski element dokumenta `<TEI.2>`, in naslov URL, kjer je dostopen DTD TEI P4, nato pa določi, da uporabljamo XML različico TEI z osnovnim naborom za prozo ter dodatnimi nabori za slike, povezovanje in transkripcijo primarnih virov, na koncu pa še določi datoteki, kjer najdemo uporabniško določene oznake.

```
<!DOCTYPE TEI.2 SYSTEM "http://www.tei-c.org/P4X/DTD/tei2.
dtd" [
  <!ENTITY % TEI.XML       "INCLUDE">
  <!ENTITY % TEI.prose     "INCLUDE">
  <!ENTITY % TEI.figures   "INCLUDE">
  <!ENTITY % TEI.linking   "INCLUDE">
  <!ENTITY % TEI.transcr   "INCLUDE">
  <!ENTITY % TEI.extensions.ent SYSTEM "tei2-ext.ent">
  <!ENTITY % TEI.extensions.dtd SYSTEM "tei2-ext.dtd">
]>
```

Slika 3: Primer TEI DTD

2.5 DTD za tekstnokritične izdaje

Parametrizacija TEI na sliki 3 je obenem kar tista, ki smo jo uporabili za definicijo tipa dokumentov našega gradiva za projekt znanstvenokritičnih izdaj. Edine »skrite karte« v tem DTD-ju so uporabniško določene oznake – te v našem pri-

meru vsebujejo definicijo nekaj novih elementov in atributov ter naštete vrednosti nekaterih atributov, ki imajo v TEI za vrednost poljuben niz. Tako so npr. vrednosti atributa rend (prikaz) pri elementu emph fiksirane na vrednosti sup, it, u, dblu (nadvrstično, ležeče, podčrtano, dvojno podčrtano).

Vendar pa TEI DTD, kot je podan na sliki 3, dopušča mnogo več elementov in atributov, kot jih potrebujemo v naših izdajah. To preobilje za samo preverjanje pravilnosti oznak v dokumentu ni kritično, je pa neugodno za uporabo urejevalnika XML, ki ponuja preveč možnih oznak uredniku besedil. Zato smo v postopku izdelave e-izdaj napisali tudi »mali«, striktni DTD, ki je specializiral in omejil TEI DTD za potrebe naših izdaj. Končna javna izdaja pa spet uporablja integralni TEI DTD.

DTD, kot je podan na sliki 3, zahteva dostop do medmrežja, kjer se na naslovu http://www.tei-c.org/P4X/DTD/ nahajajo datoteka za definicijo fragmentov TEI DTD; tei2.dtd je datoteka, v kateri se potem sklicuje na ostale potrebne datoteke. Za procesiranje je dostikrat bolj enostavno imeti celoten DTD v eni datoteki, shranjeni lokalno – TEI ponuja to možnost z uporabo TEI Pizza Chef, mrežnega servisa, kjer v formularju izberemo module TEI, ki jih želimo uporabiti, po možnosti še uporabniško definirane oznake, nakar nam servis vrne celoten DTD v eni datoteki.

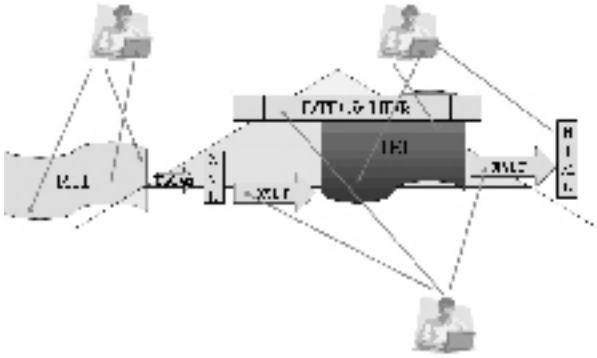
3 Metodologija izdelave digitalnih izdaj

V tem poglavju opišemo proces izdelave digitalnih izdaj v projektu. Naša metodologija (Erjavec, Ogrin, 2004) je v marsičem rezultat sestave projektne skupine, saj je eden od partnerjev imel ekspertizo predvsem na področju izdelave znanstvenokritičnih izdaj, drugi pa predvsem na področju jezikovnih tehnologij in uporabe standardov. Pomembne so bile tudi prioritete pri izdelavi prvih pilotskih izdaj.

3.1 Iz zapisa Word v TEI

Izhodišče nam je bilo v večini primerov delo, ki je bilo pred tem že izdano kot knjiga in na voljo v digitalni obliki, tipično v formatu urejevalnika Word. Izdelava digitalne izdaje tako zajema predvsem konverzijo izhodiščnega formata, orientiranega k izgledu dokumenta, v zapis TEI, kjer so oznake eksplicitne in pomensko usmerjene. Postopek izdelave je ilustriran na sliki 4 – tu predstavlja vodoravna os proces izdelave dokumentov (izdelava obsežnejših traja dlje), navpična pa količino koristne informacije, ki jo dokumenti vsebujejo – piramida tako ponazarja vložek dela, ki je

potreben za pretvorbo izvornega formata v standardizirani zapis (up translation), in nato razmeroma enostavno pretvorbo kanonične oblike v predstavitev HTML (down translation). Sodelujoči v tem procesu so po eni strani avtorji oz. uredniki, po drugi pa programerji. Prvi so zadolženi predvsem za vsebino, pri čemer jih lahko delimo na tiste, ki se omejijo na uporabo urejevalnika Word, in one, ki se seznanijo z XML in TEI in uporabljajo urejevalnike XML. Programerji pa so zadolženi za implementacijo transformacij XSLT in za izdelavo DTD-ja in glave TEI.



Slika 4: Metodologija izdelave digitalne izdaje TEI in HTML

Po zapisu originalnega dokumenta v urejevalniku Word, ki je shranjen v njegovem formatu RTF, je prva stopnja obdelave konverzija v osnovni zapis XML. Za to pretvorbo obstajajo razna orodja; mi trenutno uporabljamo komercialni program UpCast podjetja Infinity Loop. Nato so za vsako izdajo posebej napisane pretvorbe, ki v več korakih spremenijo k izgledu usmerjen zapis XML v zapis TEI. Pretvorbe so napisane v jeziku XSLT (za pretvorbo elementov) in Perl (za pretvorbo vzorcev v besedilu v elemente in vrednosti atributov). Za vsak material tudi že zelo zgodaj napišemo skripto XSLT, ki format TEI pretvori v HTML, ki ga lahko potem vidimo v poljubnem brskalniku.

Proces izdelave končne inačice zapisa v TEI je cikličen. Izhod pretvorbe (TEI, predvsem pa iz njega narejeni HTML) uredniki ovrednotijo, pri čemer lahko odkrijejo tri tipe napak: (1) napake v osnovni datoteki Word, (2) napake v konverziji ali (3) napake v izbiri označevanja. Za (1) popravimo datoteko v Wordu, za (2) pretvorbene skripte in za (3) semantiko DTD-ja. Nato ponovno pretvorimo original

in cikel ponovimo. Ta proces je bil v začetku precej dolgotrajen, z večjim številom iteracij; delno je bil razlog prepletanje izdelave posameznih faz med partnerjema, vendar se je ciklus bistveno skrajšal z naraščajočimi skupnimi izkušnjami. Po izdelavi prvega čistopisa v TEI (skupaj z glavo) lahko digitalni original v formatu Word zavržemo in nadaljnje izboljšave delamo neposredno na gradivu v TEI. Takšen pristop hitre izdelave prototipov je spodbujal skupno delo in izmenjavo ekspertize.

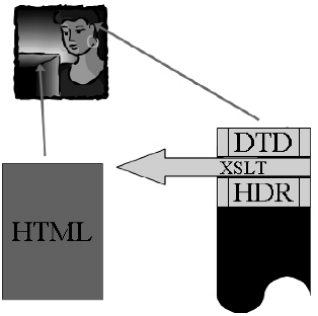
3.2 Prikaz materialov v HTML

Zapis v TEI je seveda potrebno tudi prikazati, in to na način, ki naj bi bil kar najbolj zvest virom (še posebej pri diplomatičnih prepisih), dostopen, izkoriščal pa naj bi tudi možnosti, ki jih ponujata medmrežje in XML. Kot smo omenili, že na samem začetku procesa izdelave posamezne izdaje naredimo v uredniške namene stil XSLT, ki realizira prikaz v HTML. Ta prikaz je že razmeroma kompleksen, saj po eni strani prikaže strukturo in avtorske oziroma uredniške posege v besedilo (poudarki, popravki), po drugi pa izrazi paralelizem posameznih prepisov in faksimila. Kot izredno koristen se je pokazal pogled na gradivo, ki poleg faksimila prikaže po odstavkih ali celo po vrsticah poravnana dva prepisa, npr. diplomatični in kritični ali pa diplomatični in prevod, saj tak pogled hitro razkrije nekonsistentnosti med variantnima zapisoma.

Ta uredniški izpis potem postane, z manjšimi dodatki, tudi izpis namenjen »končnim uporabnikom«, t. j. bralcem. Trenutno torej omogočamo en sam pogled na vsako TEI-izdajo. Ta pristop je ilustriran s sliko 5 in ima prednost v svoji enostavnosti, saj je prehod iz TEI v HTML realiziran z eno samo XSLT pretvorbo. To pomeni, da lahko k vsaki izdaji v TEI enostavno dodamo še izdajo v HTML, novejši brskalniki pa omogočajo celo neposreden prikaz XML s pripisanim XSLT. Ker je izdajo moč v celoti prepisati na računalnik to pomeni, da jo lahko beremo na svojem računalniku z uporabo samo spletnega brkljalnika. Slabost takšne enolične statične rešitve pa je njena neprilagodljivost različnim zahtevam in tipom uporabnikov – z uporabo programskih rešitev (npr. v okviru strežnika HTTP) je mogoče te omejitve preseči, vendar pa prisilijo bralca k sprotni uporabi medmrežja oz. k instalaciji programja.

Pomemben del izpisa v HTML za bralca, ki ga zanima o izdaji zvedeti več, je prikaz glave TEI. Preslikava z XSLT namreč preslika elemente glave v njihov lokaliziran opis, npr. `<respStmt>` v Responsibility statement oz. Izjavo o odgovornosti, pri tem pa še poveže vsako oznako z njeno definicijo iz kazala elementov v Priporočilih P4. Ker glava TEI vsebuje tudi seznam vseh elementov, ki se

uporabljajo v telesu dokumenta, vsebuje HTML-inačica glave v vsaki izdaji tudi neposredno dostopno dokumentacijo za vse elemente XML, ki se uporabljajo v tej izdaji. In ker je kopija celotnih Priporočil P4 del posamezne izdaje, je mogoče do te dokumentacije dostopati tudi brez povezave z medmrežjem.



Slika 5: Uporaba izdaje v zapisu TEI

3.3 Zapis in izpis posebnih znakov

Problem, na katerega bo naletela vsaka e-izdaja besedil starejšega slovenskega slovstva, posebej še, če vsebuje več prepisov, je kodiranje in omogočitev prikaza posebnih znakov. Že pri starejših pisavah se srečamo z nenavadnimi črkami (ſ, ʃ), problem pa je toliko hujši pri starejših rokopisih, kot so npr. Brižinski spomeniki, saj dostikrat ni enostavno ugotoviti, ali je zapis neke črke posebnost pisarja ali pa uveljavljena dvojnica. Poleg tega lahko kritični aparat vsebuje tudi prevode in primerjave (tudi s starocerkvenoslovanščino, npr. Ѣ) ali pa fonetični prepis, ki spet vpelje v izdaje nove znake (npr. ə, ɦ), ki imajo dodatno kompleksnost zaradi nad- in podvrstičnih znamenj (diakritik, npr. ǽ, â).

Formalni način, kako poljubni znak zapisati v dokument, obstaja že iz SGML; v ta namen se uporabljajo entitete (nadomestni nizi), ki si jih definiramo sami. Kot del samega standarda SGML pa je obstajal tudi nabor podanih definicij za nabore znakov, ki pokrivajo evropske pismenke, in razne druge simbole, npr. matematične, tiskarske, diakritične itn. S temi entitetami lahko npr. zapišemo Čáčka kot &C caron;áčka. Vendar pa standard ni predpisal, v kaj naj se te entitete pravzaprav preslikajo, in še pred nekaj leti bi bila predstavitev takšnih znakov – v kombinaciji na zaslonu računalnika in na način, ki bi bil neodvisen od računalniške platforme – nemogoča. S široko uveljavitvijo nabora znakov Unikod pa je to po-

stalo vsaj do določene mere izvedljivo. Z vpeljavo Unikoda v XML odpade potreba za zapis znakov s pomočjo mnemonikov in so lahko zapisani kar direktno oz. kot znakovne entitete XML. Seveda ni nujno, da bo vsak računalnik imel instalirano pisavo (*font*), ki bo vseboval znak za vsako kodo Unikod, uporabljeno v izdaji, vendar je skrb za to na plečih proizvajalcev operacijskih sistemov, pri tem pa je že sedaj tipična pokritost zadosti velika.

Kljub temu pa ostajajo problemi, saj vseh potrebnih znakov v Unikodu vseeno ni. Za nekatere od znakov smo pristali pri pragmatični rešitvi, kjer uporabimo kodo, ki po videzu svojega znaka, ne pa po definiciji, ustreza zahtevanemu znaku. Tako smo npr. za veliki S s kljukico, ki ga v Unikodu ni – čeprav ta vsebuje mali dolgi S, tj. ſ, katerega par je S s kljukico –, uporabili 222B, torej kodno točko za integral, ∫, ki je seveda matematični operator, čeprav se po videzu približa zahtevani črki. Mogoče bi bilo v tem primeru bolje uporabiti entiteto kot mnemonik, denimo &Slong;, vendar pa ima ta rešitev slabost, da so entitete definirane v DTD, tako da njihova uporaba onemogoča dokumentom, da bi jih distribuirali neodvisno od DTD, torej samo kot dobro oblikovane dokumente XML, saj so pravilni dokumenti XML sicer koristni za razvoj, za samo distribucijo pa je dodatna obremenitev z DTD pogosto nepotrebna; prav zato smo posebne znake raje shranili neposredno v naboru Unicode, pa četudi pod »napačno« kodo. Rešitev z »ugrabitvijo« drugih znakov je mogoča samo v izoliranih primerih, za generalno rešitev novih znakov pa je potrebno poseči po drugih sredstvih.

Za znake, ki niso definirani v Unicode, pa jih vseeno potrebujemo, vsebuje Unicode zasebno področje (*Private Use Area*), kjer kodne točke lahko definiramo po svoje. Seveda s tem samo omogočimo zapis posebnih znakov v Unicode, ne pa tudi njihov prikaz oz. definicijo. Na našo srečo je bil, prav tako na ZRC SAZU, razvit nabor posebnih znakov, namenjenih predvsem jezikoslovcem, ki se ukvarjajo s zapisom zgodovinskih in dialektoloških besedil in fonetičnih prepisov. Sistem ZRCola (Weiss, 2004) vsebuje takó definicijo Unicode kodnih točk v zasebnem področju za te znake kot tudi font za njihov izpis in Word makro za vnos. Nabor ZRCola vsebuje veliko število znakov, potrebnih za zapis raznovrstnega gradiva, ki smo ga ali pa ga nameravamo obdelati v našem projektu – za tiste, ki jih ne, pa upamo, da jih bo možno dodati v nabor ZRCola.

4 Jezikoslovne obdelave

V dosedanji obravnavi izdaj v projektu je bil poudarek na standardizaciji digitalnega zapisa, ki pa v marsičem vseeno samo eksplicira informacijo, ki je že prisotna v knjižni obliki. Digitalne izdaje pa lahko preučujemo tudi po njihovi jezikovni plati, na njihovi osnovi izdelamo slovarje in konkordance in omogočimo poizvedovanje po jezikoslovnih kriterijih. Takšna obravnava znanstvenokritičnih edicij je podobna izdelavi in jezikoslovni uporabi besedilnih korpusov, za realizacijo pa bi uporabili tudi podobne metode.

Izdelava korpusa poteka v določeni meri enako kakor izdelava digitalne edicije – iz izvornega digitalnega zapisa se najprej preide v zapis TEI, čeprav ima osnovni zapis besedil TEI v korpusu običajno precej manjšo gostoto in kompleksnost oznak kakor pa tekstnokritični. Potem se korpusu dodajo jezikovne oznake, in sicer oznake za besede, tem pa se pripišejo oblikoskladenjske oznake in osnovne oblike (leme). Za označevanje se uporabljajo programi, naučeni na vnaprej označenih korpusih in izboljšani z raznimi hevristikami (Brants, 2000, Erjavec et al. 2005). Končni rezultat takšnega plitkega jezikoslovnega označevanja podajamo na sliki 6.

```
<s id=   "Osl.1.2.2.1">
<w lemma="biti    " ana="Vcps-sma  ">Bil</w>
<w lemma="biti    " ana="Vcip3s--n ">je</w>
<w lemma="jasen   " ana="Afpmsnn   ">jasen</w>
<c>,</c>
<w lemma="mrzel   " ana="Afpmsnn   ">mrzel</w>
<w lemma="aprilski" ana="Aopmsn    ">aprilski</w>
<w lemma="dan     " ana="Ncmsn     ">dan</w>
<w lemma="in      " ana="Ccs       ">in</w>
<w lemma="ura     " ana="Ncfpn     ">ure</w>
<w lemma="biti    " ana="Vcip3p--n ">so</w>
<w lemma="biti    " ana="Vmps-pfa  ">bile</w>
<w lemma="trinajst" ana="Mcnpnl   ">trinajst</w>
<c>.</c>
</s>
```

Slika 6: Primer jezikoslovno označenega besedila

Do sedaj smo izdelali že večje število korpusov slovenskega jezika (Erjavec et al. 1998, Erjavec 2002, Erjavec 2004, Erjavec, Vintar 2004, Erjavec et al. 2005), in jih tudi označili, vendar pa prinaša jezikoslovno označevanje besedil v starinski

slovenščini in v tekstnokritičnem aparatu čisto nove izzive. Pri označevanju besed, ki jih v sodobni slovenščini ne najdemo ali pa se pišejo drugače, je problem, da niso zajete v računalniškem slovarju sistema, ki jih zato ne zna pravilno analizirati; da pa sistem prilagodimo drugačni leksiki, so potrebni večji posegi (Van Eynde et al., 2002). Tekstnokritični aparat pa uvede še dodatno kompleksnost v označevanje, saj v nekaterih primerih (popravki besedila) ni povsem jasno, kaj naj bi označevali (izvorni zapis ali popravek), večje število prepisov pa tudi sili k njihovi usklajeni in paralelizirani obravnavi.

Ko so besedila jezikoslovno označena, je mogoče iz njih avtomatsko generirati konkordance in slovarje. Dandanes se kot vir konkordance skorajda razume program, ki na uporabnikovo poizvedbo vrne vse konkordance iz korpusa, ki ustrezajo iskalnemu pogoju, vendar pa je bilo tradicionalno pojmovanje poln izpis vseh besednih oblik v korpusu, skupaj s kontekstom. Tehnična izvedba prvega ali drugega vsaj v osnovi ne predstavlja problemov, saj je mrežni konkordančnik že implementiran na IJS (Erjavec, Vintar, 2004) in tudi že podpira večje število eno- in dvojezičnih slovenskih korpusov. In ker so znanstvenokritične izdaje po številu besed razmeroma majhne, vsaj v primerjavi s tipičnim korpusom, bi bilo tudi možno narediti statične polne konkordance, in jih v obliki HTML vključiti v izdajo.

Povezan s konkordancami je slovar, ki vsebuje besede ali besedne zveze iz edicije. V pomembnejših delih je verjetno smiselno izluščiti kar statični slovar vseh vsebovanih besed, v drugih pa mogoče samo besede, posebej pomembne za konkretno edicijo, npr. osebna imena v pismih. Dejstvo, da imamo na razpolago več med sabo poravnanih prepisov nekega besedila, omogoča tudi izdelavo vzporednih slovarjev, ki primerjajo leksiko dveh ali več prepisov. Metodologija (pol)avtomatske izdelave takšnih slovarjev je podobna tisti, ki se uporablja pri izdelavi dvojezičnih slovarjev s pomočjo vzporednih korpusov (Vintar, 2002, Krek, 2003), kar še dodatno osmisli variantne prepise. Takšni slovarji se potem tudi zapišejo v formatu TEI (slovarski modul), vanje pa se lahko vključijo kot primeri tudi konkordance iz besedila ali pa kar kazalci v samo besedilo.

Takšna analitična orodja bi pripomogla k razkrivanju notranje kompleksnosti, znanstvenih potencialov in zgodovinske vrednosti besedil, ki jih objavljamo v naših digitalnih izdajah.

5 Zaključki

V prispevku smo opisali izdelavo standardiziranih digitalnih znanstvenokritičnih edicij slovenskega slovstva. Poudarek v prispevku je bil na opisu standardov, ki smo jih uporabili za e-edicije, na postopku pretvorbe digitalnih izvirnikov v ta standardizirani zapis in v možnosti uporabe jezikovnih tehnologij za nadaljnjo analizo besedil.

V prihodnosti projekta nas pričakuje še dosti drugih izzivov. Poleg večanja števila monografij bi v zbirki izboljšali predstavitev materialov, predvsem z možnostjo prilagoditve mrežne predstavitve posameznemu uporabniku. S formo v HTML bi bilo tako možno izbrati, kako, če sploh, si želimo videti faksimile, kateri prepisi nas zanimajo, ali želimo videti popravke ali čistopis itn. Na ta način bi se videz materialov lahko prilagajal posameznim tipom uporabnikov, od zahtevnih, ki jih zanima vse bogastvo znanstvenokritične izdaje, do osnovnih, kot so npr. učenci, ki jim večinoma zadošča en sam, poenostavljen vidik besedila.

Sama spletna stran projekta je trenutno precej enostavna, s kazalci na posamezne publikacije, ki pa so med seboj ločene. Z večanjem števila izdaj pa se bo pojavila tudi potreba po poenotenem iskanju po materialih. Tu je najprej potrebno poskrbeti za enako zapisane metapodatke o izdajah, tj. za vsebino njihovih glav TEI. Trenutno smo v ta namen uporabljali kar standardno glavo TEI, čeprav za podroben opis rokopisov obstaja razširjena glava, ki je bila izdelana v okviru projekta MASTER (Manuscript Access through Standards for Electronic Records), ki pa naj bi tudi postala del standardne glave TEI v inačici P5 – njena izdaja je predvidena v letu 2005. Takrat bomo z materiali prešli na novi TEI in omogočili tudi izvoz »kataložnih listkov« v repozitorije metapodatkov, kot je denimo OLAC (Open Language Archives Community).

Literatura

Brants, T. (2000) TnT – A Statistical Part-of-Speech Tagger. V *Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000.* Seattle, WA: ACL, str. 224–231.

Van Eynde, F., Zavrel, J., Daelemans, W. (2002). Part of Speech Tagging and Lemmatisation for the Spoken Dutch Corpus. V *Proceedings of the Third International Conference on Language Resources and Evaluation, LREC'02*, str. 1427–1434.

Durusau, P., Brook O'Donnell, M. (2001). Implementing Concurrent Markup in XML. *Extreme Markup Languages, Conference Proceedings*, Montréal.

Erjavec, T., Gorjanc, V., Stabej, M. (1998). Korpus FIDA. *Konferenca Jezikovne tehnologije za slovenski jezik*. Ljubljana: Institut »Jožef Stefan«, str. 124–127.

Erjavec, T. (2002). *The IJS-ELAN Slovene-English Parallel Corpus*. International Journal of Corpus Linguistics, 7(1), str. 1–20.

Erjavec, T. (2003). Označevanje korpusov. *Jezik in slovstvo* 48 št. 3/4, str. 61–76.

Erjavec, T., Evans, R., Ide, N., Kilgarriff, A. (2003). From Machine Readable Dictionaries to Lexical Databases: the Concede Experience. V *Proceedings of the 7th International Conference on Computational Lexicography, COMPLEX'03*, Budimpešta.

Erjavec, T. (2004). MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. V *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC'04*, str. 1535–1538.

Erjavec, T. and Džeroski, S. (2004). *Machine Learning of Morphosyntactic Structure: Lemmatising Unknown Slovene Words*. Applied Artificial Intelligence 18(1), str. 17–40.

Erjavec, T., Ogrin, M. (2004). E-Slomšek: elektronska znanstvenokritična izdaja retorske proze 19. stoletja po standardu XML TEI. V *Jezikovne tehnologije : zbornik B 7. mednarodne multi-konference Informacijska družba IS 2004*, Ljubljana: Institut »Jožef Stefan«, str. 87–93.

Erjavec, T., Vintar, Š. (2004). Korpus kot podpora slovarju informacijskega izrazja slovenskega jezika. *Uporabna informatika (Ljubljana)*, 12/2, str. 97–106.

Erjavec, T., Ignat, C., Pouliquen, B., Steinberger, R. (2005). Massive Multilingual Corpus Compilation: Acquis Communautaire and totale. (2005) *Proceedings of the 2nd Language & Technology Conference*, April 21–23, 2005, Poznań (v tisku).

Krek, S. (2003). Sodobna dvojezična leksikografija. *Jezik in slovstvo*, 43/1, str. 45–60.

Slomšek, A. M., Faganel, J., Ogrin, M., Erjavec, T. (2004). *Tri pridige o jeziku : elektronska znanstvenokritična izdaja v zapisu XML - TEI. Verzija 1.4*. Ljubljana: Inštitut za slovensko literaturo in literarne vede ZRC SAZU. ISBN 961-6500-64-3.

Sperberg-McQueen, C. M., Burnard, L. (ur.) (2002). Text Encoding Initiative: Guidelines for Electronic Text Encoding and Interchange, TEI P4, XML-compatible edition. TEI Consortium.

Thompson, H. S., McKelvie, D. (1997). Hyperlink Semantics for Standoff Markup of Read-only Documents. V *Proceedings of SGML Europe '97*, Barcelona.

Vintar, Š. (2002). Avtomatsko luščenje izrazja iz slovensko-angleških vzporednih besedil. V Zbornik 3. konference o jezikovnih tehnologijah, Ljubljana, str. 78–85.
Viscomi, J.: Digital Facsimiles (2002). Reading the William Blake Archive. *Computers and the Humanities*, 36, str. 27–48.
Weiss, P. (2004). Vnašalni sistem ZRCola. V *Jezikovne tehnologije : zbornik B 7. mednarodne multi-konference Informacijska družba IS 2004*, Ljubljana: Institut »Jožef Stefan«, str. 124–125.

Seznam pomembnejših spletnih naslovov

- http://nl.ijs.si/e-zrc/
  Elektronske znanstvenokritične izdaje slovenskega slovstva
- http://www.w3.org/XML/
  XML: eXtended Markup Language
- http://www.unicode.org/
  Unicode
- http://www.w3.org/TR/xslt
  XSLT: XML Stylesheet Language Transformations
- http://www.relaxng.org/
  RELAX NG
- http://www.w3.org/XML/Schema
  XML Schema
- http://www.tei-c.org/
  TEI: Text Encoding Initiative
- http://www.tei-c.org/Activities/MS/
  MASTER: TEI Taskforce on Manuscript Description
- http://www.language-archives.org/
  OLAC: Open Language Archives Community
- http://zrcola.zrc-sazu.si/
  Vnašalni sistem ZRCola za jezikoslovno rabo v programu MS Word
- http://nl2.ijs.si/
  Mrežni konkordančnik IJS
- http://nl.ijs.si/ME/
  MULTEXT-East: Večjezikovni korpusi in leksikoni
- http://www.fida.net/
  Korpus FIDA

Scholarly Digital Editions of Slovenian Literature: Standards and Challenges

Tomaž Erjavec, Ljubljana

The project Scholarly Digital Editions of Slovenian Literature is, on the technical side, based on the use of standards for digital text encoding. The fact that we do not tie our digital encoding into a particular application, but use open international standards and recommendations, should make the project results interchangeable among computer platforms and applications, ensure their clarity and make them proof against technological change.

In the paper we first introduce the main standards that we use in the project, namely the eXtended Markup Language, XML, which provides the basic markup infrastructure for the project and the Text Encoding Initiative Guidelines, TEI P4, which define a rich vocabulary of XML elements needed to encode complex text-critical editions. We also mention some related recommendations, in particular the XML transformation language XSLT, which provides the mechanism for converting one type of XML documents to another.

We then describe the standards-based methodology developed in the technical production of our digital editions. It consists of up-translating the materials from their digital source, usually Word, to the canonical standardised TEI format, and the down-translation into a format suitable for reading, i.e. HTML. We also discuss the problems – and our solutions – to encoding complex (archaic, phonetic) characters that appear in text-critical editions.

We next consider a challenge still ahead, namely the introduction of language technologies into the compilation and presentation of digital text-critical editions. We discuss the basic linguistic processing steps, i.e. word-level syntactic tagging and lemmatization, and the problems we are faced with if they are to be applied to complex editions containing archaic language. The utility of the linguistic markup for concordancing and lexicon extractions is also discussed. The paper concludes with a discussion and a broader view of our further work.