T A R T U Ü L I K O O L

MATEMAATIKA-INFORMAATIKATEADUSKOND

Arvutiteaduse instituut

Keeletehnoloogia õppetool

Informaatika eriala

Siim Viiklaid

„Histoloogiasõnastiku” teisendamine TBX kujule

Bakalaureusetöö (6 EAP)

Juhendaja: vanemteadur Heiki-Jaan Kaalep

Autor: ……..…...…… “…..“ juuni 2011

Juhendaja: ……..……. “…..“ juuni 2011

Lubada kaitsmisele:

Professor ……...…..…… “…..“ juuni 2011

TARTU 2011

Sisukord

Sissejuhatus

TBX (*TermBase eXchange*) on märgistuskeelel XML põhinev terminibaasi standard, mis on mõeldud põhiliselt terminoloogilise informatsiooni vahendamiseks. TBX poolt defineeritud raamistik võimaldab terminiandmete analüüsi, muutmist, esitust ja levitamist erinevates arvutisüsteemides [1].

Käesoleva töö eesmärgiks on Ülo Hussari koostatud TeX vormingus oleva histoloogiasõnastiku teisendamine TBX standardile vastavasse formaati. TBX kujul olevat sõnastikku on lihtne vajadusel edasi teisendada erinevatesse esitusformaatidesse, näiteks HTML veebileheks või väljatrükitud sõnastikuks.

Lisaks teisendamisele on töö eesmärgiks veel sõnastiku eestikeelsest osast ristviidete otsimine. Ristviite saab tekitada olukorras, kus mingi mõiste definitsioonis on kasutatud terminit, mis tähistab mõnda muud selles sõnastikus olevat mõistet. Ristviidete lisamine sõnastikule teeb selle kasutamise lõppkasutajale mugavamaks.

TBX standardiga kirjeldatud failiformaati kasutatakse näiteks Keeleveebis avaldatud sõnastike teisendusprotsessides [2] ning ka käesolev töö on osa sellest projektist. Varasemalt on bakalaurusetööna sõnastikke TBX formaati teisendanud Mait Kommusaar [3] ja Lauri Eskor [4].

Töö koosneb praktilisest ja kirjalikust osast. Praktiliseks pooleks on sõnastiku teisendamiseks vajalike programmide kirjutamine, teisenduse läbiviimine ja lõpptulemuse silumine. Töö kirjalik osa koosneb terminibaaside põhimõtte ja TBX lühitutvustest ning teisendusprotsessi kirjeldusest.

Esimeses peatükis antakse ülevaade terminibaasi põhimõttest ja ülesehitusest ning TBX vastavusest sellele.

Teises peatükis on kirjeldatud algandmete ülesehitust ja iseärasusi ning nende mõju teisendamisprotsessile.

Kolmandas peatükis kirjeldatakse teisendamise protsessi: metoodikat, valitud vahendeid ja lahendusi teises peatükis esitatud probleemidele.

Neljandas peatükis on lühidalt kirjeldatud teisendatud sõnastiku ülesehitust.

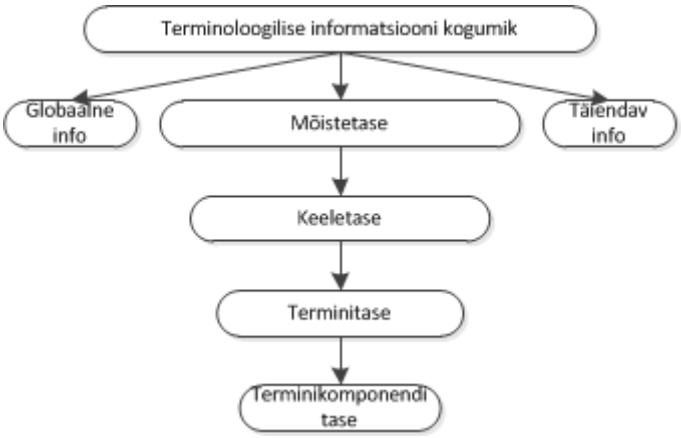
Lisana on CD-l esitatud töötlemiseks kasutatud programmid, sõnastiku lähtefailid, teisendatud sõnastik ja TBX standardi definitsioon.

1. TBX kui terminibaas

1.1. Terminibaasi põhimõte

Terminibaas on terminiandmeid sisaldav andmebaas. Terminibaasi põhimõttest aru saamiseks peab teadma, mida tähendab mõiste ja mis on selle erinevus terminist. Mõiste ei kujuta endas mingit kindlat sõna või fraasi, vaid on tähendus: inimese peas olev, reaalsuse abstraheerimise teel saadud kujutlus [5]. Termin on aga sõna või sõnaühend, mis vaid tähistab mõistet. Üks termin võib tegelikult viidata mitmele mõistele – näiteks tee kui rajatis ja tee kui jook. Mõistet võib nimetada erinevate terminitega, ka erikeelsetega, kuid selle olemus jääb ikkagi samaks.

Terminibaaside mudel (joonis 1) lähtubki mõistetest mitte kindlatest sõnadest. Ühte mõistesse kuuluvad terminid loetakse automaatselt võrdväärseteks (sünonüümideks) ning ka erinevad keeled on samal tasemel. See võimaldab näiteks terminibaasi teisendamisel sõnastikuks valida lähtekeeleks ükskõik millise keele.



Joonis 1. TMF Metamudel [1]

1.2. TBX

TBX (*TermBase eXchange*) on struktureeritud terminoloogilise informatsiooni hoidmiseks ja vahendamiseks loodud formaat. Tegu on LISA/OSCAR'i[1] poolt defineeritud suhteliselt populaarse standardiga (aastast 2008 ISO 30042), mis põhineb omakorda TMF (*Terminological Markup Framework*) standardil. TBX dokument on terminibaas, mis on kirjeldatud märgistuskeeles XML ja vastab TBX standardi poolt defineeritud ülesehitusele [1].

[1] LISA - *Localization Industry Standards Association*. Alates 28.02.2011 likvideeritud [8]
OSCAR - *Open Standards for Content/Container and Reuse*. LISA komitee.

TBX dokument, olles vastavuses TMF metamudelile (joonis 1), on terminoloogilise informatsiooni kogumik, milles on kirjeldatud üks või rohkem mõistet ja täiendav info dokumendi kohta.

Igas mõistetasemes on üldised andmed mõiste kohta (näiteks definitsioon, kontekst, lisainfo jms) ja üks või rohkem keeleüksust.

Keeletasemes asub üks või rohkem terminitaseme üksust ühes kindlas keeles, lisaks muu info, näiteks definitsioon.

Terminitasemes peab olema täpselt üks termin ning see võib valikuliselt sisaldada muud termini kohta käivat infot (piirangud termini kasutamise kohta, info grammatika kohta, viited teistele terminitele jpm).

Terminikomponendi taseme kasutamine on TBX formaadis valikuline. See sisaldab näiteks administratiivset informatsiooni termini kohta.

2. Sõnastiku esialgne struktuur

Algandmed asusid neljas eraldi tekstifailis, milles olid samad sissekanded, kuid erinevates keeltes: eesti, inglise, saksa ja vene keeles. Sissekande all mõistame siin ühe kindla mõiste ühes kindlas keeles kirjeldatud informatsiooni (terminid, definitsioon ja lisainfo). Sissekanded olid kirjeldatud tekstide töötluseks mõeldud küljenduskeeles TeX.

Käesolev histoloogiasõnastik sobib TBX kujule teisendamiseks, sest selle vastab terminibaasi omale (vt. peatükk 1.2. „TBX"):

- sõnastik on jaotatud mõisteteks (sissekanded on mõistete mitte kindlate sõnade kaupa)
- mõiste on jaotatud keelte vahel (keeled on samal tasemel)
- keeleüksus koosneb definitsioonist ja terminitest, kusjuures terminid on võrdväärsed (ehk sünonüümid)

Tüüpiline sissekanne oli nagu järgnevas näites (siin ja edaspidi on numbrid ridade ees mõeldud vaid viitamiseks):

1. `[\pl erżtrotsżżdid] ehk \idx{{\bg punalibled}} ({\it erythrocyti})~---`
2. `kaksikn÷gusad tuumata, hemoglobiini sisaldavad vererakud, diameeter`
3. `5--7,5~$\mu$m.  Tsirkuleerivad veres umbes 120~pleva, mille jłrel nad`
4. `fagotsżteeritakse p÷rna, maksa ja punażdi makrofaagide poolt. Nende`
5. `kogupind inimese tsirkuleerivas veres on umbes 3800~m$^2$, żletades`
6. `seega kehapinda umbes 2000~korda.`

Iga sissekanne algas terminiga (ülaltoodud näites *erżtrotsżżdid* real 1), millele võis järgneda selle sünonüüm (*punalibled* real 1).

Eestikeelsetes sissekannetes võis esineda mõiste ladinakeelne termin (*erythrocyti* real 1), kusjuures teiste keelte sissekannetes polnud ladinakeelset terminit eraldi välja toodud.

Terminid, sünonüümid ja ladinakeelsed terminid olid ümbritsetud läbi kogu sõnastiku enamjaolt samade TeX'i vorminduskäskudega ning neile järgnes alati sümbolijada ~---, mis märgib tühikut ja mõttekriipsu.

Definitsioonides (read 2 kuni 6) oli tavatekst, mis vaheldus TeX'i käskudega, näiteks tähistab märgijada *μ* (real 3) sümbolit μ (müü).

Mitte-ASCII tähemärgid eesti- inglise- ja saksakeelsetes sissekannetes olid kujutatud valede sümbolitega (*erżtrotsżżdid* - *erütrotsüüdid*), kuna autor ei suutnud nende failide kodeeringut kindlaks teha. Seetõttu tuli kõik valed sümbolid asendada (selliseid sümboleid

oli siiski vähe: õ, ä, ö, ü ja Ö). Venekeelse faili esialgne teisendusvorming oli IBM866 ja sellega lisatööd polnud.

Sissekannete struktuur oli võrdlemisi lihtne, kuid vajaliku info sõelumist raskendasid variatsioonid ja erandid sissekannete struktuuris ning TeX'i käsud.

2.1. Variatsioonid

All on välja toodud mõned näited erinevatest sissekannete ülesehitustest. Sõnastiku reaalsed andmed on tähistatud rasvase tekstiga.

```
1. [\pl TERMIN]~--- DEFINITSIOON
2. [\pl TERMIN] ({\it TERMIN LADINA KEELES})~--- DEFINITSIOON
3. [\pl TERMIN ehk SÜNONÜÜM] ({\it TERMIN LADINA KEELES })~--- DEFINITSIOON
4. [\pl TERMIN] ehk \idx{{\bg SÜNONÜÜM}}~--- DEFINITSIOON
5. [\pl TERMIN] ehk \idx{{\bg SÜNONÜÜM}} ({\it TERMIN LADINA KEELES})~--- DEFINITSIOON
6. [\pl TERMIN] ({\it TERMIN LADINA KEELES})~--- DEFINITSIOON Vrd.~SEOTUD TERMIN
```

Lisaks ülaltoodud variatsioonidele esines ka nende kombinatsioone ja erandlikke sissekandeid.

2.2. Erandid

Erandite tõttu ei saanud kogu teisendusprotsessi automatiseerida, vaid pidi tegema manuaalselt parandusi algandmetes või tulemusfailis. Mõned tüüpilised erandid olid:

1. Esines sissekandeid, milles terminite korduv osa oli märgitud asendava sidekriipsuga või erinev osa sulgudega eraldatud. Näiteks on järgneva bloki puhul terminiteks „interstitsiaalvedelik" ja „koevedelik" mitte „interstitsiaal-" ja „koevedelik".

```
[\pl interstitsiaal-] ehk \idx{{\bg koevedelik}}
```

Siin on aga terminiteks „pseudotiinus" ja „ebatiinus":

```
[\pl pseudo- (eba-) tiinus]
```

2. Üksikute sissekannete puhul võis termini kasutamine oleneda sellest, kas juttu on loomast või inimesest.

```
[\pl rasedus] (inimesel), \idx{{\bg tiinus}} (loomadel)
```

3. Sissekandega seotud terminid olid definitsioonist eraldatud erinevate lühenditega:

- eesti keeles „Vrd." (võrdle) või „Vt." (vaata)
- inglise keeles „*Cf.*" (*confer*)
- saksa keeles „*Vgl.*" (*vergleich*)
- vene keeles „*Ср.*" (*сравни*) või *„См"* (*смотри*)

Seotud termineid võis olla erinev arv või puudusid need üldse ning kõik neist polnud tingimata sõnastikus esindatud.

4. Üksikute mõistete puhul oli ühe või mitme keele sissekanne puudu.

5. Esines suurel hulgal muid raskestiavastatavaid vigu ja erandeid (nt. *it* asemel *ita*, ~--- asemel ~--, kirjavead, puuduvad või liigsed sulud jpm)

2.3. TeX'i käsud

Definitsioonides ja vähemal määral ka terminites esines TeX'i käske vormingute või erisümbolite kirjeldamiseks. Näiteks *AA* kirjeldab sümbolit *Å* ning _ (alakriips) näitab, et sellele järgnev sümbol peab olema vormindatud allindeksina. Pidi arvestama ka üksteise sees asetsevate käskudega, näiteks on järgnev käsuhulgas kokku kolm taset:

```
${\norm {\hbox{mm}}^3}$
```

3. Teisendamine

3.1. Metoodika

Autor valis teisenduse läbiviimiseks evolutsioonilise prototüüpimise (*evolutionary prototyping*) metoodika, mis on üks väledatest tarkvaraarendusmetoodikatest [10].

Evolutsioonilise prototüüpimise puhul valmib alguses rakenduse tuum, mida järk-järgult vastavalt uutele nõuetele edasi arendatakse. Esialgne prototüüp on funktsioneeriv süsteem, milles on implementeeritud vaid rakenduse põhilisemad omadused. Klient või lõppkasutaja saab seega juba varajases staadiumis rakendust testida ning selle kohta tagasisidet anda. Saadud info põhjal lisatakse prototüübile funktsionaalsust või tehakse parandusi, misjärel antakse prototüüp uuesti kliendile kasutada. Protsessi korratakse niikaua, kuni prototüüp on arenenud lõplikuks süsteemiks. Praegusel juhul olid arendaja ja klient samas rollis – pärast teisenduse läbiviimist sai autor kohe sõnastikust ülevaate ning oli näha, mis on veel tegemata.

Evolutsioonilist prototüüpimist kasutatakse siis, kui algul pole täiesti selge, milline lõplik produkt olema peaks või on oht, et arenduse käigus võib tekkida palju uusi nõudeid. See kehtib ka praeguse projekti puhul. Varasemate sarnaste teisendustööde puhul [3][4] kulus palju aega vigade otsimiseks ja ka käesolevas sõnastikus oli palju variatsioone, erandeid ja erisümboleid, mida on esmapilgul raske tuvastada (vt. peatükk 2. „Sõnastiku esialgne struktuur"). Prototüüpimine annab võimaluse juba töö algusest alates näha teisenduse väljundit, mis lihtsustab oluliselt vigade leidmist ja uute vajalike nõuete kindlakstegemist.

Esiteks aitas prototüüpimine kiirelt leida probleeme, mille peale poleks enne sõnastiku lõppkuju nägemist tuldud, näiteks:

- ristviidetest tuleb eraldada kirjavahemärgid
- definitsiooni lõpus võib esineda loetelu seotud terminitest
- sünonüümid pole alati samamoodi esitatud
- sissekannetel on palju variatsioone

Veel oli prototüüpimisest kasu erandliku ülesehitusega sissekannete leidmiseks. Olles jaganud sissekande teisenduse käigus alamosadeks (termin, sünonüüm, definitsioon ja lisainformatsioon) olid valesti eraldatud osad paremini märgatavad.

```
[\pl foliaat-] (ehk \idx{{\bg lehis-) papill}} ({\it papilla foliata})~---
keelepapillide liik, mis asetseb keele kżljel keele-suulaekaare ees. Koosneb mitmest
transversaallehest ({\it folium papillae}), mille kżlgedel asetsevad arvukad
maitsmispungad. See keelenłsa tżżp on inimesel vłhem vłlja arenenud kui mitmetel
loomadel (hobusel, seal, koeral, kżżlikul). Foliaatpapillid puuduvad młletsejalistel.
```

Eelneva näite teevad erandlikuks sulud sõnapaari „ehk lehis-” ümber, mistõttu loeb regulaaravaldis terminiks sõna „foliaat-” ning definitsiooniks kogu ülejäänud sissekande. Sellist definitsiooni algust on aga inimsilmaga kerge märgata:

```
(ehk \idx{{\bg lehis-) papill}} ({\it papilla foliata})~---
```

Samuti oli lõpptulemusest lihtne eristada teisendamata TeX'i tähiseid, näiteks järgnevaid:

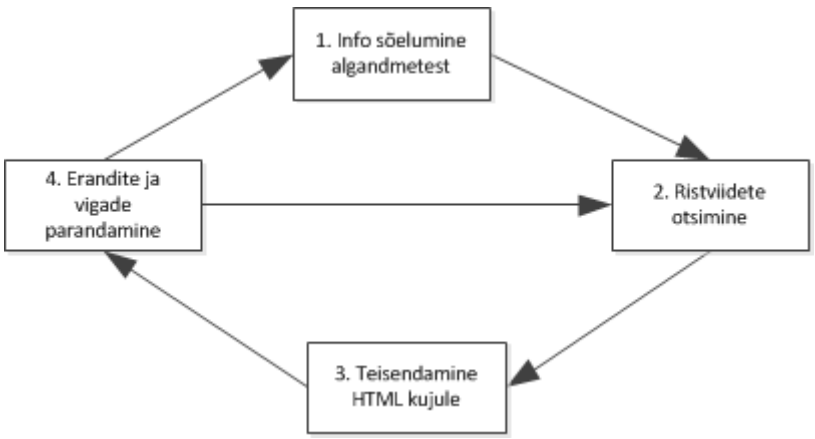
- `$\mu$` - μ
- `\AA` või `{\AA}` - Å
- `\%` - %

Kuna evolutsiooniline prototüüpimine on väle tarkvaraarendusmetoodika, siis kulub sellega vähem aega nõuete analüüsile ja tarkvara disainimisele - autor saab kohe tööle asuda programmi nende osadega, mille töötamisest on juba ettekujutus olemas.

3.2. Teisenduse üldplaan

Teisendamisele eelnes algandmete kopeerimine üheks failiks ja ülejäänud teisendusprotsess jagunes neljaks osaks (joonis 2):

1. Vajaliku informatsiooni sõelumine algandmetest
2. Ristviidete leidmine ja lõpptulemuse genereerimine
3. Lõpptulemuse esitamine mugaval kujul
4. Vigade ja erandite parandamine algandmetes



Joonis 2. Tööprotsess

Iga etapi jaoks valiti sobivaim töövahend olenevalt etapi iseärasustest. Protsess oli vastavalt metoodikale tsükliline – pärast sõnastiku teisendamist lihtsastiloetavale HTML kujule sai sealt välja lugeda vigu ja erandeid, mille parandamiseks tuli muuta algandmeid või teisendusprogramme.

3.3. Informatsiooni sõelumine algandmetest

Vajaliku informatsiooni eraldamiseks algandmetest valmis skript programmeerimiskeeles Perl. Perli valimisel osutus otsustavaks faktoriks selle väga arenenud regulaaravaldiste käsustik, mis võimaldab vähese vaevaga keerulisi andmemustreid kirjeldada.

Skripti tööpõhimõte oli itereerida läbi sissekannete ning vastavalt sissekande struktuurile valida sobilik regulaaravaldis sellest andmete saamiseks. Lisaks sellele realiseeriti siin etapis TeX'i erisümbolite ja mitte-ASCII sümbolite teisendus.

Skripti väljundiks on fail, kus sõnastiku andmed on esitatud lihtsa XML struktuurina. XML formaadiga saab edukalt kujutada sõnastikuandmete hierarhilist struktuuri. Kuna tegu on standardse märgistuskeelega, siis on selliseid andmeid teisenduse järgmises etapis lihtne sisse lugeda. Näide ühest sissekandest:

```
1.  <entry>
2.    <et>
3.      <term>atreetiline folliikul</term>
4.      <la>folliculus atreticus</la>
5.      <defin>degenereeruv ovariaalfolliikul.</defin>
6.    </et>
7.    <en>
8.      <term>atretic follicle</term>
9.      <defin>degenerating ovarian follicle.</defin>
10.   </en>
11.   <de>
12.     <term>atretischer Follikel</term>
13.     <defin>degenerierender Ovarialfollikel.</defin>
14.   </de>
15.   <ru>
16.     <term>атретичный фолликул</term>
17.     <defin>дегенерирующий овариальный фолликул.</defin>
18.   </ru>
19. </entry>
```

Ladinakeelne termin (real 4) on lihtsuse mõttes sissekande eestikeelse osa (read 2 kuni 6) alamosa. Järgmises teisendusetapis seatakse ladinakeelne osa teiste keeltega võrdseks.

3.4. Ristviidete otsimine

Kõige keerulisem teisenduse etapp ehk ristviidete otsimine realiseeriti programmeerimiskeeles Python. Pythonil on olemas ulatuslik standardteek, kus on realiseeritud kõik teisendamiseks vajalikud moodulid:

- *xml.sax.handler* XML failide lugemiseks
- *re* regulaaravaldiste jaoks
- *subprocess* UNIX'i kesta kasutamiseks
- *difflib* sõnede sarnasuse leidmiseks

Lisaks lubab Python kasutada objekt-orienteeritud programmeerimisstiili, mis lihtsustab hierarhilise ülesehitusega sõnastiku töötlemist. Samas on keele lihtne süntaks sobilik sõnetöötluseks.

Ristviite saab tekitada, kui mõiste definitsioonis või lisainformatsiooni sektsioonis on kasutatud terminit, mis asub mõne muu mõiste koosseisus. Sõnastiku suuruse tõttu (774 mõistet viies erinevas keeles) pidi algoritmid kavandama nii, et nende keerukus poleks liialt suur. Samuti pidi arvestama järgnevate teguritega:

1. Terminid võivad olla erinevates vormides. Näiteks sõnad „koed" ja „kudedesse" on mõlemad sõna „kude" algvormid, kuid nende kirjapildid erinevad teineteisest märgatavalt.

2. Terminid võivad olla mitmesõnalised, kusjuures mitmesõnaline termin võib endas omakorda sisaldada muid termineid.

```
epiteliokoriaalne platsenta
```

Ülaltoodud näites sisaldab kitsama tähendusega mõistet märkiv termin „epiteliokoriaalne platsenta" endas laiema tähendusega mõistet märkivat terminit „platsenta". Sellistel juhtudel tuleb eelistada täpsemat mõistet.

3. Definitsioonides võivad terminid olla ümbritsetud kirjavahemärkidega.

```
(aksonid),
```

Eeltoodud näites on sõnal ümber sulud ja järgneb koma, kuid ristviite peaks tekitama vaid terminile „aksonid".

Ristviited asuvad teisendatud sõnastikus mõistete definitsioonides ning on muust tekstist eraldatud spetsiaalsete märgistega.

```
1.    <descrip type="definition">
2.  kohev sidekoeline <hi type="entailedTerm" target="eid-282">kest</hi>, mis seob sooni
    ja teisi &#245;&#245;nesorganeid naaberorganitega; sooneseina v&#228;liskest.
3.    </descrip>
```

Ülaltoodud näites on viide lisatud sõnale „kest" (rida 2). Märgise <*hi*> atribuut *type* väärtusega *entailedTerm* tähendab, et antud termin eksisteerib käesolevas dokumendis. Atribuut *target* märgib viidatava sissekande unikaalset identifikaatorit.

3.4.1. Ühesõnaliste terminite viited

Esimese probleemi lahendamiseks kasutati eesti keele morfoloogilist analüsaatorit. Tegu on Eesti Keele Instituudi tarkvaraga, mis muuhulgas võimaldab leida eestikeelsete sõnade algvorme ehk lemmasid [7]. Lemmasid saab kasutada ristviidete leidmiseks ühesõnalistele terminitele, mitmesõnaliste terminite puhul analüsaator ei tööta. Analüsaatorit kasutati läbi UNIX'i kesta Pythoni mooduli *subprocess* abil.

Kui analüüsida sõna, mis analüsaatori sõnastikus puudub, siis proovib programm algvormi oletada. Sellisel juhul ei väljastata vaid ühte lemmat vaid kõik oletatud sõnad. Järgnevalt on näide analüsaatori väljundist sisendi „adetsiduaadid" puhul.

```
adetsiduaadid
    adetsiduaad+d //_S_ pl n, //
    adetsiduaadi+d //_V_ d, //
    adetsiduaat+d //_S_ pl n, //
    adetsiduaati+d //_V_ d, //
```

Kuna analüsaatori sõnastikus puudub sõna „adetsiduaat", siis pakutakse algvormideks nelja erinevat varianti: „adetsiduaad", „adetsiduaadi", „adetsiduaat" ja „adetsiduaati". Algvormidest sai kõrvale jätta verbid (tähistatud „_V_"), kuna mõistete sissekannetes verbe polnud. Lisaks sellele tuli analüsaatori väljundist eemaldada üleliigsed tähised.

Otsing toimus algoritmiga, mis esmalt leiab kõikide ühesõnaliste terminite kõik tuletatud lemmad ja lisab nad paisktabelisse, mille võtmeteks on terminite algvormid ja väärtusteks viidad mõistete sissekannetele. Seejärel itereeritakse läbi kõikide definitsioonide kõikide sõnade. Kui vähemalt üks antud sõna lemmadest on paisktabelis võtmena olemas, siis saab tekitada ristviite.

3.4.2. Mitmesõnaliste terminite viited

Eesti keele analüsaator ei tööta mitmesõnaliste terminite puhul. Seetõttu pidi rohkem kui ühest sõnast koosnevate sõnade jaoks kasutama muud viisi.

Selliste terminite puhul itereerib algoritm läbi terminite ja otsib definitsioonidest ristviiteid n-sõnaliste sõnaühendite kaupa, kus n on termini sõnade arv. Näiteks termini „valged vererakud" puhul jagatakse otsingul sõnad kahesõnalistesse gruppidesse ning iga grupi puhul võrreldakse selle sarnasust sõnaühendiga „valged vererakud".

Võrdlemine toimub Python'i standardteeki kuuluva mooduliga *difflib*, mis kasutab sõnede sarnasuse leidmiseks modifitseeritud Ratcliff-Obershelp'i algoritmi [6]. Sõnede sarnasust kujutatakse ujuvkomaarvuga lõigus [0,1], kus 0 näitab, et sõnedes pole võrdseid jadasid ning 1 seda, et sõned on täpselt samasugused.

Viiteotsingulgoritm lisab viite juhul, kui sarnasus on suurem kui 0,9 ning väärtuste 0,8 kuni 0,9 puhul küsitakse kasutajalt, kas viide tekitada või mitte. Need väärtused määrati pärast mõningast katsetamist. Mõned üksikud automaatselt tekitatud viited on küll valed, kuid neid pole raske hiljem manuaalselt parandada.

Samuti peab arvestama asjaoluga, et mitmesõnalise termini üksikud osad võivad omakorda terminid olla, kuid sõnaühendi kohta võib vaid ühe viite tekitada. Selle vältimiseks otsitakse viited kõigepealt mitmesõnalistele terminitele ja alles siis ühesõnalistele. Viidete tekitamisel märgitakse sõnaühend spetsiaalselt ära, et ühesõnaliste terminite otsimisel need vahele jäetaks.

3.4.3. Kirjavahemärkide eemaldamine viidetest

Lihtsuse mõttes töötab otsingualgoritm sõnade kaupa, kusjuures eraldajaks on tühik. Niimoodi võib üks selline sõna olla järgnev:

```
(aksonid),
```

Vajaliku termini kättesaamiseks on realiseeritud regulaaravaldis, mis jagab sõna kolme ossa:

- Algus: `(`
- Termin: `aksonid`
- Lõpp: `),`

Terminiga tehakse vajalikud operatsioonid, st lisatakse viide:

Algus: `(`
Termin: `<hi type="entailedTerm" target="eid-11“>aksonid<hi>`
Lõpp: `),`

Lõpuks lisatakse komponendid tagasi kokku:

```
(<hi type="entailedTerm" target="eid-11“>aksonid<hi>),
```

3.4.4. Väljund

Selle etapi väljundiks on TBX spetsifikatsioonile vastav XML dokument, mis on täpsemalt kirjeldatud peatükis 4 “Teisendatud sõnastiku struktuur”.

3.5. Teisendamine HTML kujule

TBX dokumendi teisendamine HTML kujule oli käesoleva töö jooksul vajalik etapp, et teisendatud dokumenti visualiseerida ning avastamata vigu ja erandeid leida. HTML osutus valituks oma lihtsuse tõttu.

Teisendus realiseeriti programmeerimiskeeles XSLT (*Extensible Stylesheet Language Transformations*). XSLT programm on reeglistik, mis kirjeldab XML dokumendi struktuuri ja sisu teisendusreegleid. Levinuim kasutusala XSLT-le ongi XML-kujul olevate andmete teisendamine HTML-iks. Teisendus toimub jooksvalt brauseris, ei nõua kasutajapoolset sekkumist ja toimib kõikide populaarsemate brauseritega[9][2].

Tulemuseks on viieveeruline HTML tabel, milles on ridade kaupa sissekanded, igas veerus erinev keel.

3.6. Erandite silumine ja vigade parandamine

Algandmete ülesehituses olevate erandlike sissekannete tõttu pidi pidevalt jälgima teisenduse lõpptulemust, et arvestada varem märkamata jäänud nüanssidega. Lisaks peatükis 3.4. „Teisendamine HTML kujule" kirjeldatud meetodile aitas vigu otsida veel ristviidete logi, mis genereeritakse ristviidete otsimise käigus.

Vigade parandamine polnud teisendamise kindel etapp vaid pidev protsess. Parandused toimusid enamasti algandmetes, kuid oli vaja ka muuta programmifaile ja üksikutel juhtudel lõpptulemust. Viimast näiteks juhul, kui lemmatiseerija oletas (lisaks teistele oletustele) algvormiks mingi muu sõna (nt. sõna „peenis" lemmaks „peen") või tekkisid ristviited homonüümide vahel („keel" kui organ või „keel" kui suhtlusvahend).

See etapp nõudis ka asjatundjatega konsulteerimist. Näiteks oli erandlike sissekannete hulgas palju selliseid, mille termineid oli raske eristada.

```
[\pl limaskesta proopria] ehk \idx{{\bg płriskiht}}
```

Ülaltoodud terminipaari võib eelteadmisteta inimene tõlgendada mitut moodi:

- Limaskesta proopria on sama mis päriskiht
- Limaskesta proopria on sama mis limaskesta päriskiht
- Proopria on sama mis limaskesta päriskiht

Võõrkeelsete sissekannete puhul oli lisaks vaja keelelist abi.

[2] Eranditeks on brauserid Google Chrome ja Chromium, mille puhul ei suutnud autor avada ühtegi **lokaalset** XML faili, sealhulgas antud töö tulemit

4. Teisendatud sõnastiku struktuur

TBX standard on väga mahukas ja paindlik, võimaldades kirjeldada terminoloogiat paljudest aspektidest lähtuvalt. Selles peatükis kirjeldatakse spetsiifiselt just selle projekti jaoks valitud ülesehitust, kuna kogu TBX spetsifikatsiooni kirjeldamine väljuks antud töö piiridest.

4.1. Mõistetase

Järgnevalt on kujutatud tüüpiline näide sõnastiku sissekandest. Keeletasemete sisu on siin ruumi kokkuhoidmiseks asendatud *[---]* märgistega.

```
1.  <termEntry id="eid-5">
2.    <langSet xml:lang="et">
3.      [---]
4.    </langSet>
5.    <langSet xml:lang="en">
6.      [---]
7.    </langSet>
8.    <langSet xml:lang="de">
9.      [---]
10.   </langSet>
11.   <langSet xml:lang="ru">
12.     [---]
13.   </langSet>
14.   <langSet xml:lang="la">
15.     [---]
16.   </langSet>
17. </termEntry>
```

Igal mõistel peab olema unikaalne identifikaator (*eid-5* real 1). Terminiinfo on jaotatud osadeks keelte järgi (näiteks eesti keel ridadel 2 kuni 4).

4.2. Keeletase

Järgnevalt on toodud näide ülaltoodud sissekande eestikeelsest osast.

```
1.  <langSet xml:lang="et">
2.    <descrip type="definition">
3.  kohev sidekoeline <hi type="entailedTerm" target="eid-282">kest</hi>, mis seob sooni
    ja teisi &#245;&#245;nesorganeid naaberorganitega; sooneseina v&#228;liskest.
4.    </descrip>
5.    <ntig>
6.      <termGrp>
7.        <term>adventiitsia</term>
8.      </termGrp>
9.    </ntig>
10.   <ntig>
11.     <termGrp>
12.       <term>adventitsiaalkest</term>
13.       <termNote type="termType">synonym</termNote>
14.     </termGrp>
15.   </ntig>
16. </langSet>
```

Keeletase koosneb definitsioonist (read 2 kuni 4) ja terminitest (read 5 kuni 15). Igal mõiste keelejaotusel on märgitud *xml:lang* atribuudina keelekood.

Definitsioon (read 2 kuni 4) koosneb tavatekstist ning metamärgistest. Näiteks märgis <*hi*> real 3 kujutab endas ristviidet mõistele, mille identifikaator on *eid-282*. On veel eraldi märgiseid kursiivis, allindeksis ja ülaindeksis formaaditud teksti kohta.

XML standardi kohaselt peavad mitte-ASCII tähemärgid olema kodeeritud (nt. *väliskest* real 3).

Lisaks definitsioonile ja terminitele võib keeletasemes olla ka lisakommentaar <*note*> märgiste vahel:

```
<note>Vrd. <hi type="entailedTerm" target="eid-93">eferentne</hi></note>
```

4.3. Terminitase

Terminitaseme üksused asuvad <*ntig*> märgiste vahel (read 8 kuni 12 ja 13 kuni 18).

Kuigi terminid, nagu ka keeled, on terminibaasi seisukohalt omavahel võrdsed, on praegusel juhul eristatud põhitermin (*adventiitsia* real 7) ja teine selle sünonüümiks märgitud (*adventitsiaalkest* real 12). Selleks on mitu põhjust:

1. Algses sõnastikus oli põhitermin selgelt eristatud
2. Kui tulevikus on vaja sõnastikku teisendada tähestikuliseks, siis peab iga sissekande kohta olema üks võtmesõna
3. Terminid säilitavad siiski võrdsuse

Kokkuvõte

Töö esimeseks eesmärgiks oli teisendada algselt TeX formaadis olev „Histoloogiasõnastik" TBX standardi spetsifikatsioonile vastavaks XML dokumendiks. Lisaks sellele pidi tekitama sõnastiku kirjete vahele ristviited.

Teisenduse tulemit saab kasutada „Histoloogiasõnastiku" teisendamiseks erinevatesse esitusformaatidesse ning ristviidete olemasolu teeb sõnastiku kasutamise mugavamaks.

Töö kirjalikus osas toodi välja tähelepanekud teisendamise käigus tekkinud probleemide kohta ja nende lahendamiseks valitud meetodid. See informatsioon võib kasuks tulla tulevaste teisendustööde korral. Samuti on tehtud kokkuvõte terminibaaside ülesehituse põhimõttest.

Töö tulemit kasutatakse sõnastiku publitseerimisel keeleveeb.ee portaalis.

The conversion of ‘Histoloogiasõnastik’ to TBX

Bachelor’s thesis

Siim Viiklaid

Abstract

TBX (TermBase eXchange) is an XML-based standard for representing and exchanging terminological data in various computer environments. The objective of this thesis is to convert ‘Histoloogiasõnastik’ (dictionary of histology, created by Ülo Hussar) to a valid TBX document. TBX format allows easier ways for transforming terminological data to various representation forms, an HTML glossary for instance. Another objective of the thesis was to generate cross-references between entries of the ‘Histoloogiasõnastik’, which would make using the dictionary more convenient for the end-user.

TBX document is a termbase (terminological database). Termbase should be designed according to a particular model that allows converting it to different formats and prevents systematic errors during the creation of the database. ‘Histoloogiasõnastik’ reflects this model therefore making it possible to convert it to TBX.

The original data are in TeX format, entries being rather simple in their structure but containing a lot of different variations and exceptions. The methodology used for the conversion was cyclic in its nature, consisting of four main stages:

- parsing original files of the dictionary, outputting an XML representation of the data
- finding cross-references and forming the TBX structure, outputting the end product of the conversion
- transforming the TBX document to an HTML document, allowing easy inspection of the end result to detect errors and overlooked exceptions in the original data
- correcting mistakes of the conversion process and eliminating exceptions in the original data

The end result of the conversion is an XML document that is in accordance with the TBX specification and satisfies the main principles of a termbase design. The converted dictionary will be published in Keeleveeb, a portal that along with different linguistic resources also features other technical dictionaries similar to ‘Histoloogiasõnastik’.

Kasutatud allikad

[1] OSCAR Group. Systems to manage terminology, knowledge, and content - TermBase eXchange

http://mirror.transact.net.au/sourceforge/t/project/tr/transtandards/TBX/TBX-specification.pdf - viimati vaadatud 02.06.2011

[2] H.J. Kaalep "Korpusepäring keeleveebis", ettekanne konverentsil „Eesti keele keeletehnoloogiline tugi 2006-2010" 19. november 2007

http://keeletehnoloogia.cs.ut.ee/konverents/slaidid/kaalep-keeleveeb.pdf/ - viimati vaadatud 02.06.2011

[3] M. Kommusaar "Infotehnoloogia terministandardi projekti" teisendamine TBX kujule

Bakalaureusetöö, Tartu 2006

[4] L. Eskor "Sümboolikaleksikoni" teisendamine TBX kujule

Bakalaureusetöö, Tartu 2008

[5] A. Tavast. Terminibaasi koostamise põhimõtted - kiirülevaade insenerile

http://www.imprimaatur.ee/artiklid/kiirylev.html - viimati vaadatud 02.06.2011

[6] The Python Standard Library. *difflib* — Helpers for computing deltas

http://docs.python.org/library/difflib.html - viimati vaadatud 02.06.2011

[7] Eesti Keele Instituudi tarkvara. Morfoloogiline analüüs

http://www.eki.ee/tarkvara/analyys/- viimati vaadatud 02.06.2011

[8] LISA

http://www.lisa.org/ - viimati vaadatud 02.06.2011

[9] W3Schools. *XSLT Browsers*

http://www.w3schools.com/XSL/xsl_browsers.asp/ - viimati vaadatud 02.06.2011

[10] Steve McConnell. *Rapid development*, lk 147-150, 1996

Lisad

Lisa 1

CD, millel on sõnastiku algandmed, teisendamiseks valminud programmfailid, teisendatud sõnastik, TBX standardi spetsifikatsioon ja valideerimiseks kasutatud failid.

`thesis.pdf` – käesolev dokument
`algandmed` – sõnastiku algandmed
- `HISE.TEX` – eestikeelsed sissekanded
- `HISI.TEX` – inglisekeelsed sissekanded
- `HISS.TEX` – saksakeelsed sissekanded
- `HISV.TEX` – venekeelsed sissekanded

`histoloogia` – teisendatud sõnastik
- `hist.xml` – teisendatud sõnastik TBX kujul
- `style.xsl` – XSLT stiilifail TBX teisendamiseks HTML formaati

`tbx` – TBX standardipakett
- `TBX_2008_10_29.pdf` – TBX standardi spetsifikatsioon 29. oktoober 2008
- `TBX_RNGV02.rng` – TBX Relax NG Schema
- `TBXcoreStructV02.dtd` – TBX DTD
- `TBXXCSV02.xcs` – TBX vaikimisi XCS

`teisendus` – teisenduseks kasutatud programmid
- `paste.sh` – skript algandmete kopeerimiseks ühte faili
- `parse.pl` – skript algandmete sõelumiseks
- `Parser.py` – klass XML andmete sisselugemiseks Pythoni objektideks
  Allikas (02.06.2011):
  http://code.activestate.com/recipes/534109-xml-to-python-data-structure/
- `Converter.py` – skript ristviidete leidmiseks
- `Writer.py` – klass sõnastiku kirjutamiseks TBX dokumendiks