

Tallinna Ülikool

Informaatika Instituut

Suupilli tabulatuuride XML märgendus

Bakalaureusetöö

Autor: Raido Seene

Juhendaja: Vladimir Tomberg M.Sc.

Juhendaja: Jaagup Kippar M.Sc.

Autor: „ „2011

Juhendaja: „ „2011

Juhendaja: „ „2011

Instituudi direktor: „ „2011

Tallinn 2011

Autori deklaratsioon

Deklareerin, et käesolev bakalaureusetöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikad ja mujalt pärinevad andmed on viidatud.

.....

.....

(kuupäev)

(autor)

Sisukord

Mõistete seletused	4
Sissejuhatus	5
1 Suupillimängu notatsiooniprojekt	6
1.1 Projekti taust.....	6
1.2 Projekti eesmärgid	8
1.3 Töö eesmärgid	9
1.4 Suupilli eripärad	10
1.5 Suupilli notatsiooni näiteid.....	11
1.6 Olemasolevad lahendused ja lähenemised	13
2 Märgistuskeeled.....	17
2.1 XML	18
2.2 XML skeemid.....	19
2.3 XML skeemide reeglid	21
2.4 XML andmete visualiseerimine.....	22
3 Arendus.....	24
3.1 Skeemide disainimine.....	26
3.2 Suupilli ontoloogia	34
3.3 Tabulatuuriformaadi rakendamine	35
Kokkuvõte	38
Summary.....	39
Kasutatud kirjandus	40

Mõistete seletused

Tabulatuur – (*tablature*) on praktiline noodikiri, milles helikõrguste asemel tähistatakse instrumendi mänguvõtteid, keeli, auke, nuppe ning kõike muud, mida antud pilli mängimiseks kasutatakse. Oma lihtsuse ning instrumendispetsiifilisuse tõttu on tabulatuur kergesti arusaadav ning õpitav.

Bend – on suupilli mänguvõtte, mille käigus suupillil tekitatud heli tooni muudetakse madalamaks kui mängitavast august kostuma peaks. Tehnikat kasutatakse suupillil puuduvate nootide mängimiseks.

Overbend ja **Overdraw** – on suupilli mänguvõtted, mille käigus suupillil tekitatud heli tooni muudetakse kõrgemaks kui mängitavast august kostuma peaks. Tehnikat kasutatakse suupillil puuduvate nootide mängimiseks.

Assert – on XML skeemis 1.1 kasutatav element, mis kontrollib XML dokumendis esinevate elementide ning atribuutide väärtuste paikapidavust teiste väärtuste või tingimuslausetega.

Sissejuhatus

Käesolev bakalaureusetöö on ülevaade ning uurimus sobiva *XML* märgenduse väljatöötamisest suupillimängu notatsiooniprojekti tarvis. Suupillimängu notatsiooniprojekti näol on tegu Haridustehnoloogia keskuse teaduri Vladimir Tombergi algatusega, mille eesmärkideks on ühtse *XML* standardi loomine suupillilugude ülesmärkimiseks ning ülestähendusi kasutavate rakenduste loomine algajatele suupillimängijatele pilliõppe lihtsustamiseks.

Töö jaguneb kolmeks osaks, millest esimeses antakse ülevaade suupillimängu notatsiooniprojektist üldisemalt. Selle eesmärkidest ning suupilli ehituse eripäradega kaasnevatest probleemidest. Töö teine osa pühendub suupillimängu notatsiooni probleemide lahendamisele ning sobiva ülesmärkimise viisi leidmisele. Käesoleva töö kolmas osa annab ülevaate notatsiooniskeemide valmistamisest ning nende rakendamisest projekti edasistes faasides.

Bakalureusetöö autor liitus projektiga huvist muusika ning selle visualiseerimise vastu. Samuti peab töö autor toonitama, et käesolev bakalaureusetöö ei hõlma suupillimängu notatsiooniprojekti mitte oma täies tervikus, vaid pühendub just suupilli notatsioonile ning *XML* süntaksi väljatöötamisele.

1 Suupillimängu notatsiooniprojekt

Suupillimängu notatsiooniprojekti näol on tegu Tallinna Ülikooli Haridustehnoloogia keskuse teaduri Vladimir Tombergi algatatud projektiga, mille eesmärkideks on uurida võimalusi suupillimängu ülesmärkimiseks sobiva *XML* süntaksi loomisel ning luua väljatöötatud süntaksit realiseerivad ning kasutavad rakendused.

Vajadused ühtse standardi loomiseks suupillilugude ülesmärkimiseks ning neid töötlevate rakenduste järele tekkisid seoses seni kinnitamata üldkasutatava standardi puudumise tõttu suupillimängu ülestähendamisel. Kuigi suupillilugude ülesmärkimiseks algajatele pillimängijatele kasutatakse valdavalt küll erinevaid tabulatuuride variante või tabulatuuri ja noodikirja segusid, puudub tabulatuuride talletamiseks ühtne standard, mis võimaldaks erinevaid tabulatuure kasutada ühes rakenduses ning erinevatel suupillidel mängides.

Lihtsustamaks suupillimängu iseõppija tegevust on vajalik ülesmääritud muusikapala mängijale meelepärases vormingus kuvamine ning suurel hulgal lugude kättesaadavus ühest allikast. Muusikapalade leitavus peab olema lihtsustatud otsinguga ning kättesaadav erinevatel seadmetel alates arvutis ja lõpetades mobiiltelefoniga.

Tabulatuuri kasutavad enamasti algajad ja esmakordselt pillimängu proovijad, kuna neil puuduvad teadmised traditsioonilise noodikirja valdamiseks, mille tõttu soovitud muusikapala kättesaadavus tabulatuuris ning meelepärases vormingus on motivatsiooniks pillimängu õppimisel. Eelnevat arvesse võttes ongi suupillimängu notatsiooniprojekti eesmärgiks digitaalse tabulatuuri kui õpiobjekti loomise, levitamise, otsimise ning kasutamise lihtsustamine. (Tomberg, 2011)

Suupillimängu notatsiooniprojektis osalevad Vladimir Tomberg – projekti algataja, eestvedaja ja juhendaja, Jaagup Kippar – juhendaja, Raido Seene – programmeerija ja bakalaureusetöö kirjutaja, Eugene Ivanov – rahvusvaheline ekspert, konsultant.

1.1 Projekti taust

Suupillimängu notatsiooniprojekti käigus langetatud otsuste ja valikute selgitamiseks ning suupilli eripäradest tekkivatest probleemidest arusaamiseks peab esmalt andma lühikese ülevaate suupillist ning suupillimängu ülesmärkimiseks kasutatavast tabulatuurist.

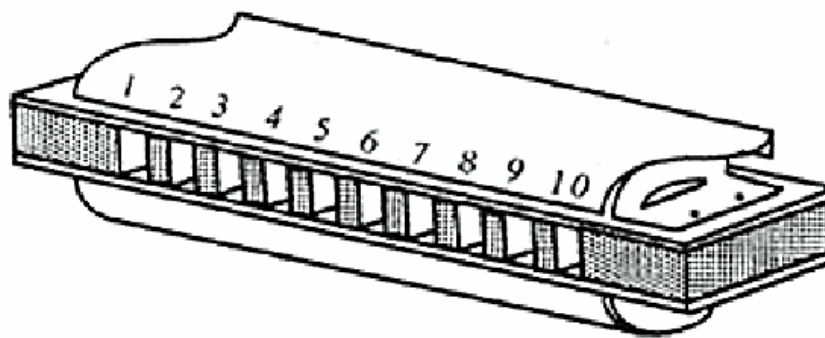
Suupill on eriline pillide seas juba oma ülesehituselt, olles ainuke puhkpill, mida saab mängida nii õhu puhumisel kui ka tõmbamisel läbi pilli. Pilli mängimisel tekib heli väikeste metallist ribade ehk keelte vibreerimisel. Suupillid jagunevad kahte suurde perekonda, diatoonilised ja kromaatilised suupillid. Peale diatoonilise ja kromaatilise suupilli leidub veel teisigi liike, nagu tremolo- ja ka orkestraalsuupill. (Myriad Software, 2004)

Diatoonilisel suupillil asub igas avas kaks keelt, millest üks võngub avasse puhumisel, teine tõmbamisel. Suupilli iga auguga on võimalik luua kaks kuni neli erinevat tooni. Kromaatilise suupilli puhul asub igas suupilli augus neli keelt, ning keeli millel mängitakse, valitakse suupillil paikneva nupu abil. Kromaatilisel suupillil saab mängida kõiki pooltoone üle mitme oktaavi, samal ajal kui diatooniline suupill võimaldab mängida vaid etteantud noote. Soovitud augu mängimiseks võib mängija mittesoovitud augud katta keelega. Samuti on võimalik akordide mängimiseks puhuda või tõmmata mitmest august korraga. Edasijõudnud mängijatel on võimalik mängida ühel augul kõrgemaid ning madalamaid noote kui sellele augule ette on nähtud, seda mänguvõtet nimetatakse *bend*'iks, *overbend*'iks ja *overdraw*'ks. (Myriad Software, 2004)

Erinevalt klaverist on diatoonilisel suupillil palju erinevaid häälestusi. Peale standardse *Richter*'i häälestuse leidub veel üle saja erineva variatsiooni. Alternatiivsed häälestused standardsele on loodud selle puuduste kompenseerimiseks. Eesmärkideks on kerge mängitavus ja suurem ilmekus mõne kindla stiili mängimisel, kromaatiline mängimine ilma *overbend*'e kasutamata ning lisamaks rohkem akorde ja intervale. (Tomberg, 2011)

Suupilli erilisus seisneb ka võimaluses selle mängu üles märkida tabulatuuri abil. Tabulaatuur on lihtsustatud viis erinevatele pillidele mõeldud muusika ülesmärkimiseks. Vanimatel andmetel kasutati tabulatuuri lautomängu ülesmärkimiseks. Tänapäeval kasutatakse tabulatuuri nii kitarril, bassil, trummidel ja paljude teiste pillide mängu noteerimiseks. Erinevalt standardsest noodikirjast pole tabulaatuur seotud kindla helistikuga. Nootide asemel kasutatakse tabulatuuris keeli, auke, nuppe ning kõike muud, mida antud pilli mängimiseks kasutatakse. Helistiku muutmine reeglina tabulatuuri ei muuda, kuna pilli häälestust saab muuta või on pill juba valmistatud kindlas helistikus. Selle tõttu on muusikapalade ühekordsel õppimisel neid lihtne mängida erinevates helistikes. Samuti on tabulaatuurid suhteliselt lihtsalt mõistetavad ja tihti kohandatud vaid ühele kindlale pillile, mille tõttu on neid üleliigse informatsiooni väljajätmise tõttu lihtne lugeda. (Tomberg, 2011)

Tabulatuuri põhiliseks eeliseks on tema lihtsus. Erinevalt noodikirjast ei vaja tabulatuuri kasutamine ning õppimine eelnevat muusikalist teavet ega õpet ning on kergesti omandatav ka laste poolt. Seetõttu on see laialdaselt kasutuses algajate pillimeeste seas, kelle noodikirja tundmise tase ei võimalda suupilli mängida otse noodist. Tabulatuuri kasutamise kasuks räägivad ka enamike diatooniliste suupillide korpustele graveeritud avade numbrid, samuti kasutatakse avade numbreid ka suupilli tabulatuuris, näidetest täpselt millist auku millise noodi kõlamiseks kasutatakse (vaata joonis 1). Samuti erinevalt noodist suudab tabulatuur kirjeldada ka erinevaid vaid suupillimängule omaseid tehnikaid. (Tomberg, 2011)



Joonis 1 Diatooniline suupill nummerdatud aukudega (Barret, 1997)

Kuna suupillimängu notatsiooniprojekti eesmärk on lihtsustada suupillimängu õppimist, keskendutakse antud projektis just nootide visualiseerimisele kasutades tabulatuuri. Seda protsessi saab aga veelgi lihtsustada rakendustega, mis õpitavas loos esile kerkivaid tabulatuure visualiseerivad ekraanile kuvatud suupillil, kasutades näiteks erinevaid värve õhuvoolu suundade kujutamiseks.

1.2 Projekti eesmärgid

Suupillimängu notatsiooniprojekti eesmärgiks on luua ühtne standard muusikapalade noteerimiseks suupillidele suupillimängu õppe eesmärgil, mis võimaldaks üles märkida kogu vajaliku info loo mängimise kohta alates puhutavatest aukudest kuni rakendatavate efektideni. Samuti tuleb koostada vorming enamkasutatavate suupillitüüpide ning nende häälestuste talletamiseks, mille abil on võimalik tabulatuure vormindada soovitud suupillil mängitavaiks. Suupilli lugude talletamiseks tuleb aga luua tabulatuuride hoidla ning rakendused tabulatuuride hoidla haldamiseks ja kasutamiseks. Tabulatuuride konverteerimiseks soovitud

suupilli seadistustel mängitavaks tuleb välja töötada spetsiaalsed rakendused. (Tomberg, 2011)

Noteerimise standard loob omakorda võimalused erinevate rakenduste väljatöötamiseks, mis kõik töötaksid ühtse standardi alusel loodud tabulatuuride peal. Õppijate võimalus ise luua, muuta ja jagada oma tabulatuure suurendaks märkimsiväärselt võrgusisest suupillimängijate kogukonna koostööd. (Tomberg, 2011)

Projekti käigus on algsete plaanide nimistusse lisandunud ka kaugemale ulatuvaid ideid. Nende seas kasutajaliidese tabulatuuri visualiseerimise valik, kus vastavalt soovile oleks võimalik valida eri tabulatuuri liikide kui ka sünkroonmängu võimaluse vahel. Viimase puhul kuvatakse ekraanile suupilli kujutis, millel näidatakse mängijale õige ajastusega mängitavaid auke, õhuvoolu suunda ja vajadusel rakendatavaid efekte. Õppimist lihtsustavaks mooduseks saab muusikapala tabulatuurifailile lisada viite video- või helisalvestusele, mida saab vastava rakenduse abil kasutada pildi või helifailiga sünkroonseks harjutamiseks. Uurimisele tuleb ka võimalus notatsioonide sisselaadimine teistest vormingutest ja salvestamine teistesse vormingutesse, nagu *MIDI* ja *MusicXML*. Need võimalused laiendaksid ja kergendaksid standardi kasutuselevõttu.

1.3 Töö eesmärgid

Kuna bakalaureusetöö raames pole võimalik tervet kavandatud projekti käsitleda ja ka töö esitamise ajaks pole suupilli notatsiooniprojekt kaugeltki valmis, on käesoleva bakalaureusetöö eesmärkideks just *XML* põhise tabulatuuriformaadi väljatöötamine, mis võimaldaks talletada kõiki klassikaliste suupillide mänguvõtteid ja neil pillidel mängitavaid muusikapalasisid. Samuti on süntaksi väljatöötamisel vajalik tulevaste rakenduste mitmekesisusega arvestamine, et loodav süntaks kajastaks notatsiooni visuliseerimiseks vajaminevat informatsiooni võimalikult kompaktselt, arvutile arusaadavalt ning ilma üleliigset infot sisaldamata.

Loodava notatsiooni üheks oluliseks kriteeriumiks on tema hilisem kättesaadavus ja arhiveerimise võimalikkus. Selle tegevuse optimeerimiseks on notatsiooniga vaja ühendada metaandmete süsteem.

Tulevase suupillilugude hoidla loomiseks tuleb tagada kõigi salvestatavate andmete õigsus ja süntaksile vastavus. Notatsioonide õigsuse kontrolliks tuleb luua skeemid, mis peavad arvestama suupillide erinevusi, ning neist tulenevaid erinevaid mänguvõimalusi ja rakendatavaid mänguvõtteid.

1.4 Suupilli eripärad

Põhiliseks suupilli eripäraks, mis vajab notatsiooni loomisel tavalisest noodikirjast hoopis erinevat lähenemist, on suupilli nii õhu puhumisel ja ka tõmbamisel tekkivad helid. Seega tabulatuuri kasutamise puhul tuleb lisaks pilli aukudele üles märkida ka õhuvoolu suund, millega vastavat heli tekitatakse. Kõnealune eripära on ka põhjus, miks on raske kui mitte just võimatu hakata suupillimängu õppima noodikirjast. Seega olenemata notatsiooni viisist on vajalik lisamärke õhuvoolu suuna ning kestvuse kohta.

Teiseks vaid suupillidele omaseks nähtuseks on erinevate pillide rohkus, ning veel suurem häälestuste rohkus. Sellest lähtudes ei saa süntaksi loomisel lähtuda mingist kindlast helistikust, ega pilli helirea ülesehitusest. Suupilli aukude toonivahe ning ka järjekord võib vastavalt häälestusele olla väga erinev. Seega notatsiooni ülestähendamisel tuleb luua süsteem, mis sobiks nii erinevate suupillide kui ka nende häälestustega. Info suupilli tüüpide ning häälestuste kohta nende rohkuse tõttu tuleb aga hoida lahutatuna muusikapala tabulatuurist. (Missin, 2007)

Kolmandaks suupilli eripäraks on talle rakendatavate mänguvõtete rohkus. Paljud neist on kasutatavad vaid kindlat tüüpi pilli puhul, teised aga rakendatavad üksikshaaval või kindla mustri alusel korraga. Luues skeemi tabulatuuriformaadi õigsuse kontrolliks, tuleb arvestada mänguvõtete rakendamise reeglitega.

Hoolimata eelnevalt loetletud tabulatuuri eelistest algajale muusikule, leidub sellel ka silmapaistvaid puuduseid. Tabulatuuril puudub täpne noodi kirjeldus. Tal on liiga palju erinevaid visualiseerimise võimalusi, mis teeb nende arusaamise arvutile keeruliseks. Ning graafilist tabulatuuri pole lihtne levitada, kuna selles sisalduvaid spetsiifilisi sümboleid ja nooli on digitaalselt raske trükkida. Selle tõttu on vajalik visuaalse ning muusikalise info üksteisest lahutamine. Ülesmärgitud suupilliloo notatsioon ei tohiks endas sisaldada üleliigset tabulatuuri visualiseerimise infot, vaid ainult antud loo olulist muusikalist infot, mille alusel on tarkvara võimeline ise visualiseeringut genereerima. Milline hakkab antud notatsiooni

visuaalne tabulatuur välja nägema oleneb juba tarkvarast ning kasutaja eelistustest. (Tomberg, 2011)

Kõige olulisemaks eripäraks, mida antud projekti raames püütaksegi parandada on suupillimängu noteerimisviiside rohkus. Iseenesest pole probleemiks just notatsioonide visuaalne erinevus, vaid tihtilugu väga erinevad talletatavad infohulgad ja viisid. Kui muusikalise info talletamine toimub ühtse skeemi järgi, on sama notatsiooni kasutamine võimalik erinevates rakendustes, ning ka notatsiooni visualisatsioonid võivad olla erinevad. Järgnevalt näiteid erinevatest suupillimängu notatsiooni viisidest.

1.5 Suupilli notatsiooni näiteid

Üldlevinud standardi puudumise tõttu kasutavad muusikapalade tabulatuuride loojad igauks natukene erinevaid võtteid lugude ülesmärkimisel. Põhjuseks on ülesmärkimise ning lugemise mugavus, mille suhtes on paljudel autoritel erinevad arvamused. Samuti puudub lihtsamate lugude ülesmärkimisel lihtsalt vajadus spetsiifiliste mänguvõtete kasutamiseks, mis mõjutab samuti tabulatuuri struktuuri. Tulemuseks on vaid suupilli aukude numbritest ning kergesti mõistetavatest õhuvoolu suundasid tähistavatest märkidest tabulatuur (vaata joonis 2).

3	3	4			3	2	2	2	1	1
B	D	D	D ^b	D	B	D	D	D	B	D

6	-5	6	-5	6	-5	6	-5
a	-	a	-	t	h	e	t
a-way a-bove the chim-ney tops							

6	-5	6	-6	-7
t	h	a	t	s
that's where you'll find me				

4	7	-7	6	-6	-7	7
s	o	m	e	o	v	e
some-where o-ver the rain-bow						

4	-6	6
b	l	u
blue-birds fly		

Joonis 2 Lihtsad tabulatuurid. Allikad 'Beginner Blues Harmonica' (Barry, 2006) ja 'Somewhere Over The Rainbow' (Harptabs, 2004).

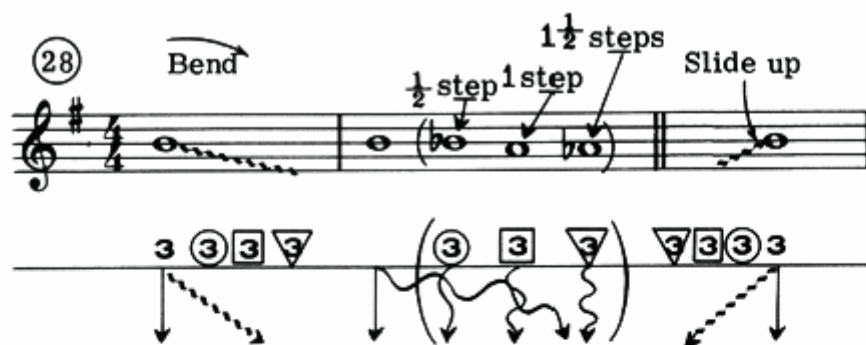
Tihtilugu seotakse tabulatuuris kasutatavad suupilli augud ka nootidega tähistamiseks kõlavate helide pikkuseid (vaata joonis 3), samas kui teistes lahendustes kasutatakse selleks erinevaid noolte pikkuseid (vaata joonis 4).

Ex. 53

Joonis 3 Nootidega tabulatuurid. Allikad *'All-American Harp'* (McCoy, 1998), *'Teach Yourself Blues Harmonica'* (Gelling, 2000), *'Rock Blues Country Harp'* (Rost, 1993) ja *'Play Irish Music on the Blues Harp'* (Power, 2000).

Joonis 4 Nooled. Allikad *'The Sourcebook of Sonny Terry Licks for Harmonica'* (Ball, 1995) ja *'Sourcebook of Little Walter/Big Walter Licks for Blues Harmonica'* (Ball, 2000).

Põhiliseks tabulatuuride erinevuse allikaks ongi lugude keerukuse varieerumine. Lihtsamate lugude ülesmärkimiseks piisab tihtilugu vaid numbritest ning kergestimõistetavatest märkidest, samas kui keerulisemate puhul tuleb kasutusele võtta erimärgid. Näiteks *bend'*ide kasutamine (vaata joonis 5).



Joonis 5 Bend'id. Allikad 'Blues Harp for Diatonic and Chromatic Harmonicas' (Duncan, 1996) ja 'The Harp Handbook' (Baker, 1999).

Kuigi mainitud tabulatuurid on disainitud spetsiaalselt suupillilugude ülesmärkimiseks, ei ole ühegagi neist võimalik tähistada kõiki olemasolevaid suupilli mänguvõtteid. (Tomberg, 2011)

1.6 Olemasolevad lahendused ja lähenemised

Hetkeseis esmapilgul suhteliselt kirjul internetimaastikul on suupilli tabulatuuride suhtes siiski väga vaene. Leidub küll suurel hulgal nii harrastajate poolt kogutud lugusid kui ka suuremaid professionaalsemaid tabulatuurikogumikke, kuid enamasti on nende veebilehtede miinuseks ühe loo tabulatuuri kättesaadavus vaid kindlale suupilli liigile, helistikule ning häälestusele. Samuti on leitavad tabulatuurid integreeritud veebilehte, ning notatsioonide allalaadimist pole peaaegu kusagil võimaldatud. Samas kui enamikel juhtudel esitatakse tabulatuuride info vaid tekstina (vaata joonis 1), puudub ka otsene vajadus algandmete kättesaamiseks, kuna need ei erine märgatavalt visuaalsest esitusest. Tabulatuuri salvestamise

soovi korral saab notatsiooni kopeerida mõnda tekstiredaktorisse. Vaid mõnel üksikul lehel leidub tabulatuure mis erinevad lihtsast numbrilisest infost ning õhuvoolu suundade tähistest ning mis on kättesaadavad näiteks *PDF* vormingus. Teretulnud on ka lahendused, kus lisaks tabulatuurile võib muusikapalast kuulata ka *MIDI* salvestist. (HarpTab, n.d.; Harpin' an' a Whinin', n.d.)

Suuremad suupillide tabulatuurileheküljed võimaldavad muusikapalasi otsida loo nimetuse, autori ning suupilli tüübi ja helistiku järgi. Pettumuseks võib aga osutada leitud loo kokkusobimatus kasutaja pilliga. Edasi on kasutajal kolm võimalust: jätkata otsinguid, lootuses leida endale sobilik tabulatuur kusagilt mujalt, püüda leitud tabulatuuri järgi improviseerida ning lugu siiski oma pillil maha mängida, mis aga algajate suupillimängijate puhul oleks võimatu või proovida tabulatuuri ise teisendada oma helistikule sobivaks.

Tabulatuuride teisendamine ühest helistikust teise sama häälestuse ning suupilli tüübi puhul on võimalik spetsiaalsete tabelitega, kus on märgitud suupilli aukude vasted erinevate helistike korral. (Molino, 2007)

Üks populaarsemaid lehekülgi suupillilugude otsijate seas on *harptabs.com*, oma väga suure lugude kogumikuga, kuid siiski omab lehekülj halba ülesehitust ning suhteliselt kesist lugude metaandmetega varustatust. Samuti pole võimalik tabulatuuride teisendamine ning erinevad häälestused on toetatud minimaalselt ning vaid erinevate suupillitüüpide standardhäälestustega. (Harptabs, n.d.)

Andreas Gräfe veebilehel leidub siiski ka üks rakendus, mis võimaldab kasutaja poolt sisestatud tabulatuure teisendada ühest helistikust teise, kuid siingi vaid ühe suupilli ning häälestuse piires. Rohkem võib leida rakendusi suupilli tuuningutest ning liikidest tulenevate häälestuste vaatamiseks ning võrdlemiseks, nagu *Harp-o-matic* ja *Harp Keys*, kuid neilgi puuduvad võimalused tabulatuuride teisendamiseks (vaata joonis 6). (Gräfe, n.d.; Alekseenko, 2007; Ivanov & Grigorov & Ladutin, n.d.)

nagu mobiiltelefonid, mis ei vaja või ei pruugi toetada kõik samu funktsionaalsusi ning visualisatsioonivõtteid, nagu arvuti.

Samuti räägib märgistuskeelte kasuks *MusicXML*, mis on saavutanud muusikalise informatsiooni talletamises ning andmevahetuses suure edu, ning mida kasutatakse rohkem kui 140 rakenduses. Miks aga ei saa suupilli notatsiooniprojekti jaoks kasutada *MusicXML*'i? Kuna *MusicXML* on mõeldud põhiliselt noodikirja talletamiseks ning püüab endas hõlmata enamust muusikalise info võimalustest, on ta teistest pillidest palju erinevama ehitusega suupilli jaoks liiga mahukas. Samas võtab *MusicXML*'is väga suure mahu enese alla suupillimänguks mittevajalik informatsioon ning puuduvad võimalused suupilli eripärast tingitud mänguvõtete ning nendega kaasneva informatsiooni talletamiseks. Selle tõttu nõuaks *MusicXML*'i kasutamine olemasoleva süntaksi tõsist ümberehitamist. Silmas pidades suupilli üldiselt lihtsat tööpõhimõtet mis ei nõua noodikirja keerulise süsteemi kasutamist ning suupilli eripäradest tingitud spetsiifilisust, sai projekti raames langetatud otsus täiesti uue, vaid suupillile mõeldud süntaksi loomise kasuks (vaata peatükk 3). (Recordare, n.d.)

2 Märgistuskeeled

Märgistuskeeled on keeled, mis märgistavad teksti selliselt, et seda oleks võimalik arvutis töödelda. Kuna märgistuskeelte märgised on kirjutatud selliselt, et nad eristuksid ülejäänud tekstis, on enamus märgistuskeeli ka inimesele loetavad. Näiteks enamlevinud märgistuskeelte, *XML*, *HTML* ja *XHTML*, märgised on '<' ja '>'. Ning tekst mis asub kahe märgise vahel on osa märgistuskeele käsklustest. (Kyrnin, n.d.)

Kuigi märgistuskeeli leidub iga valdkonna ja teema kohta, alates ärist kuni inimestevaheliste suheteni, on suupillimängu notatsiooni projekti jaoks olulisel kohal just muusika valdkonnale kohandatud märgistuskeeled.

Esimesteks katseteks muusikalist infot, eelkõige noodikirja digitaalsel kujul säilitada ja levitada olid *NIFF* (*Notation Interchange File Format*) ja *SMDL* (*Standard Music Description Language*). Kuigi *NIFF* loodi spetsiaalselt muusikalise noodikirja andmevahetuse standardiks ning võeti kasutusele paljudes notatsiooni ja muusika skanneerimise programmides, piiras standardi graafiline lähenemine muusikalise teabe talletamisel selle kasutamist muusika esituslikel ja töötlevatel seadmetel. Probleemid ei puudunud ka *SMDL*'il, millest sai ametlik standard, kuid tema nõrkuseks jäi püüdlus hõlmata kõike, mida muusika vallas kasutada võidak. Kuigi *SMDL* osutus oma eesmärkides edukaks, oli tema lähenemine nii abstraktne ja kohmakas, et ärilist rakendamist pole standard veel tänapäevani saavutanud. (Stewart, 2003)

Sarnaselt eelnevatele on järgnenud ka hilisemaid katsetusi samadel radadel, üheks tulemuseks *MuTaTeD* (*Music Tagging Type Definition*), mis kujutas endast kahe eelneva standardi kokkusulatamist. Ning *MNML* (*Musical Notation Markup Language*), mille eesmärgiks oli *HTML*'i sarnane märgistuskeel muusikalise info interneti teel levitamiseks. (Cover, 2006)

Töökindlam ning tänapäevani püsiv lahendus muusikalise info ülesmärkimisel saabus koos *XML*'i (*Extensible Markup Language*) loomisega. *XML*'ist sai kiiresti tunnustatud standard mistahes liiki andmete esitamises ja tarkvara- ning seadmetevahelises andmevahetuses. Kõige täiuslikum ja töökindlam püüdlus kasutada *XML*'i muusika talletamiseks on *MusicXML*. Nagu ka paljud teised muusikaalased märgistuskeelte notatsioonid, on ka *MusicXML*'i juured pärit akadeemilistest ringkondadest ning põhineb kahe akadeemiliselt välja töötatud töökindla

tarkvara, *MuseData* ja *Humdrum*, põhifunktsionaalsustel. *MusicXML*'i edu seisneb muusikalise kodeerimise oluliste põhifunktsionaalsuste toetamises, mitte igakülgses katvuses, nagu *MML* (*Music Markup Language*), mis oma ulatusliku raamistikuga on ainuke *XML*'il põhinev keel kirjeldamiseks väga laiaulatuslikku muusikat. Tulemuseks aga sarnasus *SMDL*'iga, kus standard on liiga keeruline, et olla kasutuses. Oma lähenemiselt on aga *MusicXML*'ist saanud *XML*'i rakendusviis, mis on leidnud tee paljudesse ärivaradesse. (Stewart, 2003; Cover, 2006)

Kuna *XML* standardi näol on tegu levinuima märgistuskeelega digitaalse muusikalise info edastamisel, mis on oma toimimist ka tõestanud, ning *XML*'ina talletatud andmeid on kerge kokku pakkida, levitada, ning korrektsel metaandmete kasutamisel ka otsida ning arhiveerida, sobib *XML* märgenduskeeltest ideaalselt digitaalseks õppematerjaliks.

Samuti räägib *XML*'i kasuks tema vaba kasutus, mis soodustab levikut ja kättesaadavust. Enamus muusikaalaseid *XML* skeeme ning *DTD* faile on kättesaadavad *royalty-free* litsentsi all. Ning neid on võimalik siduda mistahes tarkvaraga. (Stewart, 2003)

2.1 XML

XML on märgistuskeel, mis on kasutusel dokumentide ning andmete kirjeldamiseks standardiseeritud tekstipõhises vormingus, mida oleks lihtne transportida üle internetiprotokolli. Kuna *XML* on avatud andmevahetuse formaat, mis on disainitud info ülekande ning kättesaadavuse lihtsustamiseks, ei ole tegu just kõige kompaktsema vorminguga. Tihti on *XML*'ina talletatud infol suurem maht kui tema alguses vormingus. *XML*'i kasutamise muudab aga mugavaks tema suhteliselt lihtne loetavus nii inimesele kui ka arvutile. Samuti on seda võimalik kasutada erinevates protsessides, koos erinevate programmeerimiskeelte ning tarkvaradega. (Amiano & D'Cruc & Ethier & Thomas, 2006; Benz & Durant, 2003)

Ka paljud disainieesmärgid, mille alusel *XML* on loodud, kattuvad suupillimängu notatsiooniprojektis nõutavate kriteeriumitega:

- Lihtne kasutatavus üle interneti.
- Erinevate rakenduste toetamine.

- Lisaelementide ja kõige mitteolulise vähesus ning võimalus mittevajalikku välja jätta.
- Vormingu mõistetavus ning arusaadavus inimesele ning võimalus notatsiooni kerge vaevaga luua.

Viimasele aitab kaasa kodeeringu ning kompressseeringu puudumine. (Amiano & D'Cruc & Ethier & Thomas, 2006)

XML dokumentide valideerimist ehk nende vormistusliku õigsuse kontrollimist saab peale *XML* standardile omase süntaksi teostada ka struktuuri ja sisu baasil. Dokumendi vastavuse kontrollimine nõutavatele vormistusreeglitele käib *XML* skeemide (*XML Schema*) abil. (Skeemidele eelnenud standardina pole kasutuselt kadunud ka *DTD (Data Type Definition)*). (Benz & Durant, 2003)

Alates *XML*'i sünnist on erinevate notatsioonide talletamiseks võetud kasutusele just see formaat. *XML*'i soosib muusikalise struktuuri esitamise võimaluste rohkus, alates ühe noodi märgistamisest, kuni tervete noodiridade ning meloodiliste käikude diatoonilise ja kromaatilise noodivahe ülesmärkimiseni. Samuti lubab *XML* isoleerida erinevaid muusikalisi komponente. Tänu sellisele objektorienteeritud lähenemisele saab määratleda tarkvara käitumist vastavalt antud muusikalise elemendi omadustele ning vastavaid elemente käsitleda ühtemoodi olenemata tema esinemise asukohast, kas muusikapala sees või eraldiseisvana mõnes teises teoses. Peale muusikalise struktuuri võimaldab *XML* väljendada ka muusika semantikat. Mainitud omadused korvavad eelnevate standardite ülesehituslikud puudused ning muudavad märgenduse tunduvalt loetavamaks. (Stewart, 2003)

2.2 XML skeemid

XML skeem ehk *XSD (XML Schema Definition)* on *W3C (World Wide Web Consortium)* ametlikult kinnitatud *XML* dokumentide struktuuri definitsiooni standard, väljendamaks *XML* dokumentide piiranguid ning struktuurilist ülesehitust. Erinevalt *XSD* eellasest *DTD*'st (*Document Type Definition*) järgib skeem *XML* dokumendi vormistusreegleid, mis teeb tema arusaamise sama lihtsaks kui *XML* failist arusaamise. Samuti lubab skeem palju täpsemat kontrolli dokumendis esitatud info üle. *XML*'i formaati järgides, ning väga detailse struktuuri kontrolli tõttu kipuvad skeemi failid olema mahukamad kui *XML* failid, mida nad kirjeldavad. Sellest olenemata on skeemid tihtilugu arendajatele kergemini loetavad ja järgitavad kui *DTD*

failid, mille paljud elemendid esinevad varjatult ning segaselt kirjeldatult. (Benz & Durant, 2003; W3C, n.d.)

Kuigi peale *XML* skeemide on kasutusel veel *Relax-NG* ning vananenud *DTD*, on *XML* skeemid siiski enimkasutatud ning levinuim lahendus *XML* dokumentide struktuuri kontrollimiseks. *XML* skeem defineerib dokumendis lubatud elemendid, atribuudid, nende põlvnemise ning esinemise järjekorra ja arvu. Erinevalt näiteks *DTD*'st võimaldab skeem määrata palju täpsemalt elementide sisu ning nende andmetüübi, mis tagab lihtsama skeemide mõistmise. Samuti on *XML* skeemi suurimaks eeliseks tema vormistus *XML* kujul, mis teeb selle kasutuse arusaadavamaks ning lihtsamaks. (W3C, n.d.; w3chools, n.d.)

Kuna *XML* skeemid tagavad peamiselt dokumentide struktuurilise ning sisu andmetüübilise korrektsuse, on dokumendis talletatud infost tingitud piirangud jäänud seni spetsiaalselt vastava dokumendi valideerimiseks disainitud rakenduse hooleks. Kuna validaatorid on võimelised kontrollima vaid dokumentide struktuuri jäi programmeerijate ülesandeks sisust tingitud reeglite realiseerimine. Sellisel eesmärgil valmistatud validaatorid suudavad aga kontrollida vaid spetsiaalselt selle rakenduse jaoks kirjutatud dokumente ning juhul kui dokumentide struktuuri pidi muutma, tuli ümber programmeerida ka validaator.

Selle probleemi lahendamiseks töötab *W3C XML* skeemide töögrupp hetkel uue *XML* skeem 1.1 väljatöötamise kallal. Sisuliselt ollakse uue standardi loomisega valmis, ning standardit saab juba kasutada ning valideerida üksikute validaatoritega, kuid ametlikult soovitusliku standardi staatuse omandamiseks tuleb skeemil läbida veel intensiivne testimine. Hetkeseisuga on skeemi versioon 1.1 standardi kandidaadi staatuses. Peale erivalidaatoreid mittenõudvate funktsioonide väljatöötamist parandatakse uue versiooni puhul ka eelnevas standardis esinenud vigu ning lisatakse uusi võimalusi. Hoolimata uuendustest jääb skeemi struktuur ning ülesehitus sarnaseks vanema versiooniga, ning on tagasiulatuva ühilduvusega. (W3C Ubiquitous Web Domain, n.d.)

Kuna suupilli mänguvõtetes esineb palju reegleid, mis piiravad teatud võtete korraga kasutamist või esinemist teatud liiki suupillide puhul, on suupillimängu notatsiooniprojekti jaoks loodava notatsiooni süntaksi juures uue skeemi uuendused väga olulised. Kuna tabulatuuride süntaks saab olema eraldiseisev rakendustest, mis seda visualiseerivad, siis tuleb kõik mänguvõtteid keelavad ning lubavad reeglid formuleerida süntaksis endas.

2.3 XML skeemide reeglid

Oma struktuurilt ning dokumendi ülesehituse reeglilt on *XML* skeem 1.1 sarnane oma eelkäijaga, mida kinnitab ka uue skeemi tagasiulatuv ühilduvus vanema versiooniga. Juurde on lisandunud aga suurel hulgal uusi võimalusi, mis lubavad töödelda ning piiranguid seada vastavalt *XML*'is talletatud informatsiooni eripäradele. Selleks on *XML* skeem 1.1 reeglitesse integreeritud ka *XPath* reegleid, mis võimaldavad *XML* dokumendi valideerimisel selles palju dünaamilisemalt liikuda ning kontrollida palju enam kui elementide järgnevust. Uute reeglitega saab määrata elementide ning nende väärtuste sõltuvust üksteisest, elementide ning atribuutide võimalikke väärtusi ja palju muudki. Järgnevalt ülevaade lisandunud võimalustest. (Costello, 2010)

XML skeem 1.1 uued elemendid võimaldavad luua seoseid elementide, atribuutide ning nende väärtuste vahel ja sellega lubada ning keelata teiste atribuutide ning elementide olemasolu või nende väärtusi. Samuti kontrollida teiste elementide arvu, ning teostada ka keerulisemaid arvutus- ja võrdlustehteid (*assert* ja *assertion*). Olulisteks uuendusteks on veel ka elemendi tüüpide valik lähtuvalt mingist muutujast (*alternative*) ning skeemis mitteesinevate elementide lubamise võimalus (*openContent*). Lisandunud on ka võimalus üle laadida teises skeemi globaalselt deklareeritud elemente (*override*). Samuti on palju olemasolevaid elemente ning atribuute muudetud töökindlamaks, võimalusterohkemaks, ning kasutuselt mugavamaks. Suurel hulgal on lisandunud andmetüpe aja ning ajakulu määramiseks (*dateTimeStamp*, *yearMonthDuration* ja *dayTimeDuration*) ning suurema täpsusega murdarvude väljendamiseks (*precisionDecimal*). Samuti on lisandunud ka andmetüüp mis lubab kasutada eneses kõiki lihtandmetüpe (*atomicType*) või võimalus deklareerida täiesti uusi (*vendor*). Atribuutidele on lisandunud võimalus luua terve dokumendi ulatuses kehtivaid atribuute ning deklareerida neid ka alamelementidele päritavaiks. (Costello, 2010)

Tulevikku suunatud uuenduseks on aga versiooni kontroll, mis asub eraldi nimeruumis ja lubab skeemis määrata millist versiooni validaatorid toetavad antud funktsionaalsusi. Juhul kui validaator teatud liiki käsklusi ei toeta, jätab ta need lihtsalt tähele panemata. (Costello, 2010)

Kuigi uuenduste nimekiri on palju pikem ning põhjalikum, on suupilli notatsiooniprojekti süntaksis vajaminevatest uutest reeglitest kõige olulisem *assert* suupilli mänguvõtete esinemise võimalikkuse kontrollimiseks.

2.4 XML andmete visualiseerimine

XML'is paiknevate andmete visualiseerimiseks on võimalusi ning keeli mitmeid. Milliseks kujunevad suupillimängu notatsiooniprojekti raames kasutatavad lahendused kasutajaliidese loomisel, on veel vara prognoosida. Kuid järgnevalt lühike ülevaade võimalustest, mille töösse rakendamist tasub kaaluda.

XSLT on *XML*'i disainikeel, mille abil on võimalik *XML* dokumendi formaati muuta teiseks *XML* dokumendiks, *HTML*, *XHTML* leheks või tekstidokumendiks. *XSLT*^oga uut faili luues saab *XML* dokumendis leiduvaid elemente ära jätta, juurde lisada ning nende esitamise järjekorda muuta ja sorteerida. Kasutamist leiaks see suupillimängu notatsiooniprojektis lihtsa tabulatuuride väljundi loomisel, kus andmed esitatakse tabelite või andmejadadena *HTML*'is. Tabulatuuride esitamisel saab kasutada erinevaid värve näiteks erinevate mänguvõtete ja õhuvoolude märgistamiseks. Siiski on tegu suhteliselt lihtsa lahendusega ning keerulisemaid mänguvõteteid, kus tabulatuuride kujutamiseks kasutatakse ka spetsiifilisi märke ning pilte, pole kõige optimaalsem *XSLT*^oga lahendada. (w3schools, n.d.)

Flash on multimeedia platform interaktiivse sisu ning animatsioonide loomiseks nii veebilehtedele kui ka eraldiseisvate rakendustena arvutitele, televiisoritele ja telefonidele. Võrreldes *XSLT*^oga saab *Flash*'is luua palju interaktiivsemaid lahendusi, mis on võimelised kasutama *XML* dokumente andmete allikana. Selle poolest on aga *Flash* palju mahukam, ning nõuab kasutatavas seadmes *Flash Player*'i olemasolu. (Adobe, n.d.; w3schools, n.d.)

Canvas on *HTML5* element, mida kasutatakse graafika või lihtsamate animatsioonide renderdamiseks veebilehel. *Canvas*'ele joonistamiseks kasutatakse skriptikeeli, kõige tavalisemalt *JavaScript* keelt. *Canvas* element pole toetatud paljude vanemate veebilehitsejate poolt, kuid töötab nende uuematel versioonidel nagu *Firefox 1.5* ja hilisem, *Opera 9* ja hilisem, *Safari* uuemad versioonid, *Chrome* ja *Internet Explorer 9*. (Mozilla Developer Network, 2011)

SVG on märgistuskeel kahemõõtmeliste vektorgraafiliste rakenduste ning piltide kirjeldamiseks. *SVG* ga on pilte võimalik optimaalselt renderdada kõigi suurustega seadmetel. Võimalustelt asub *SVG XSLT* ning *Flash*'i vahepeal, temaga saab teha suhteliselt keerulisi graafilisi elemente kuid mitte nii interkatiivseid kui *Flash*'iga. Tööpõhimõttelt on *SVG* aga parem kui *Canvas*, jättes loodud elemendid meelde, võimaldades nende hilisemat muutmist, samal ajal kui *Canvas* joonistab kõik elemendid üksteise peale allolevaid meelde jätmata. (W3C, n.d.)

3 Arendus

Kuna suupillimängu notatsiooniprojekti näol on tegu Vladimir Tombergi huviprojektiga, mitte põhitööga, oli juba algselt selge, et sisult mahukas projekt teostub suhteliselt pika ajavahemiku vältel. Samuti ei kannata pillimängu noteerimisel kasutatava süntaksi väljatöötamine mõtlematut kiirustamist, mis võib tulevikus kaasa tuua probleeme ununenud elementide lisamisega või nõuaks halvimal juhul lausa süntaksi ümbertegemist. Kuna käesoleva töö autoril oli vaja paralleelselt süntaksi loomisega tegeleda ka bakalaureusetöö kirjutamisega, sai esialgseks bakalaureusetöö raames täidetavaks eesmärgiks määratud suupillimängu tabulatuuride XML-põhise süntaksi ning reeglite loomine ja formuleerimine. Terve projekti raames on peale tabulatuuriformaadi ning selle struktuuri ja reegleid tagavate skeemide loomist kavas arendada välja ka formaati kasutavad süsteemid ning rakendused.

Kui projekti käivitades oli juba selge, et tulevastes lahendustes tahetakse kasutada just XML põhise märgistuskeelt, jäi siiski esialgu lahtiseks rakendatava süntaksi päritolu. Kas optimaalsem oleks võtta kasutusele mõni oma toimimist tõestanud süntaks nagu seda on *MusicXML* või luua tühja koha peale täiesti uus just talle ettenähtud eesmärgi lahendav formaat? Kuna *MusicXML* on oma olemuselt mõeldud just muusikalise notatsiooni ülesmärkimiseks ning esimestest Vladimir Tombergi tabulatuurialastest uuringutest selgus ka suhteliselt suur noodikirja kasutamine koos tabulatuuridega, tundus juba eksiteeriv süntaks parima lahendusena. *MusicXML*'i ning suupilli ehituslike eripärade lähemal uurimisel selgus, et kuigi *MusicXML* on väga sobilik noodikirja talletamiseks, ei ole tegu kõige parema lahendusega suupillimängu notatsiooniprojekti jaoks. (Recordare, n.d.; Tomberg, 2011)

Kuna projekti üheks eesmärgiks on liigse visualiseerimisalase informatsiooni piiramine miinimumini nagu ka kõige muu ebavajaliku mitte talletamine, osutus *MusicXML*'i suurimaks probleemiks tema mahukas. Kuna *MusicXML* hoiab endas just visuaalset informatsiooni muusikapaladest noodikirja loomiseks koos kõigi märkide ning võimalike lisadega, on juba süntaksi ülesehitus notatsiooniprojekti eesmärkidega vastuolus. Üheks võimaluseks *MusicXML*'i kasutamisel oleks tema kohandamine suupillimängu talletamiseks, piirates ebaolulise informatsiooni mahtu. Kuid lisaks süntaksi kohandamisele ning piiramisele tuleks laialdaselt juurde luua ka täiesti uusi funktsionaalsusi ning elemente. Nimelt ei sobi *MusicXML* muutmata kujul suupillimängu ülesmärkimiseks tingituna just viimase ehituslikest eripäradest (vaata peatükk 1.4). Ülejäänud puhkpillidest erinevalt õhuvoolu suundade

vaheldumise kasutamine pilli mängimisel, ning lai valik mänguvõtteid, mida ei ole võimalik liigitada teistel pillidel kasutatavate hulka, tingivad tahes tahtmata uute spetsiaalselt suupillile mõeldud *XML* elementide ning nendevaheliste reeglite loomist. Samuti osutuks problemaatiliseks erinevate pillide ning nende häälestuste kasutamine muusikapalade konverteerimisel. Teostatavaks tööks kujuneks lõppkokkuvõttes vaid väikese osa *MusicXML*'ist eraldamine ning selle ümber siiski sisuliselt uue süntaksi loomine. Selle asemel et sooritada nii ulatuslikke muudatusi ning vältida doonoriks oleva süntaksi eripäradest tulenevaid piiranguid osutub parimaks lahenduseks luua täiesti uus süntaks just selliste funktsionaalsuste ning oluliste võimalustega nagu on vaja suupillimängu talletamiseks. (Recordare, n.d.)

Pärast uue süntaksi loomise kasuks otsustamist avanes suurem võimalus valida tulevaste keelte ning tehnikate vahel. *MusicXML* kasutab nimelt endiselt oma standardi kontrollimiseks *DTD* skeeme, ning alles 2008. aastal lisandusid sinna juurde ka *XML* skeemid. Viimasel ajal on aga kasutuskõblikuks muutunud vastset valmiv *XML* skeemi versioon 1.1 mis võimaldab eelmisest versioonist märksa rohkem vabadust ning võimalusi (vaata peatükk 2.2). Võttes arvesse suupilli eripärasid ning rohkeid mänguvõtteid, mis seavad üksteisele kasutuspiiranguid lähtudes rakendusviisist, osutus uue skeemi versioon oma suuremate võimalustega ideaalseks skeemi kandidaadiks tulevase süntaksi struktuuri kontrollimiseks. Samuti võimaldaksid uue skeemi reeglid formuleerida kõiki suupillimängu reegleid skeemisiseselt, kasutamata valideerimiseks eraldi rakendust sisulise informatsiooni õigsuse kontrollimiseks. Kitsaskohtadeks, mida versiooniga 1.0 poleks saanud kontrollida on elementide olemasolu piiramine lähtuvalt mingi teise elemendi või atribuudi olemasolust. Samuti poleks võimalik elementide arvu ning näiteks elementides esinevate väärtuste arvulise kõrvutipaiknemise tagamine. Kui aga kõnealuseid reegleid poleks võimalik sisse kirjutada süntaksisse, tähendaks see reeglite komplekti, mis tuleb realiseerida tabulatuure loovates ning visualiseerivates rakendustes. Kuna aga standardi loomine tähendab iseseisvat töötava süsteemi loomist, ei ole mõeldavgi olukord, kus teatud reeglite järgimine lükatakse kasutaja õlgadele. Kui reeglid, mis määravad tabulatuurifaili ehituse, ei ole kättesaadavad ning iga programmeerija peab need ise välja mõtlema ning oma rakendusse istutama, muutub standard ise mõttetuks. Samuti muutuvad ka erinevate rakenduste poolt loodud suupilli notatsioonifailid ebastabiilseks ning erinevates rakendustes avamatuks ning loetamatuks neis esinevate reeglite erinevuse või hoopis nende puudumise tõttu. Ning sellist olukorda püütaksegi kõnealuse standardi loomisel vältida. Just tabulatuuride erinevates rakendustes

kasutamise võimatus ning nende struktuuriline erinevus ongi peamisteks põhjusteks, miks kõnealune projekt üldse alguse sai. (Costello, 2010; Recordare, n.d.)

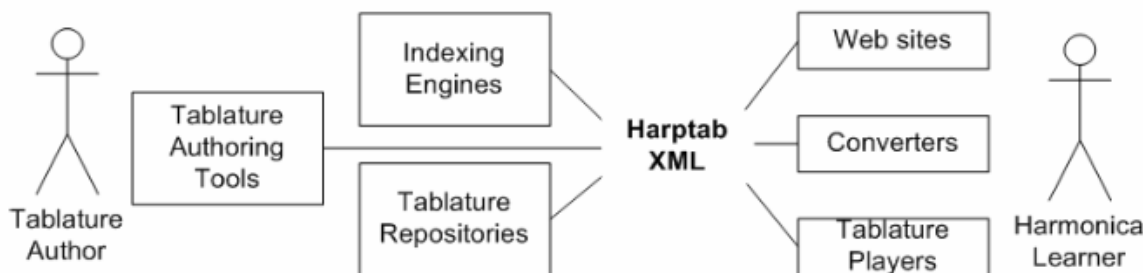
3.1 Skeemide disainimine

Uurides *XML* skeemi versiooni 1.1 selgus, et olenemata tema ametlikult veel mitte standardi staatusest saab uut skeemi juba rakendada ning kasutada. Hetkeseisuga on kättesaadavad kaks rakendust mis võimaldavad skeemi valideerimist. *Saxon 9.3* on serveripoolne *Java* rakendus, mis toetab skeem 1.1 funktsionaalsusi täielikult ning võimaldab kontrollida nii skeemi enda korrektsust kui ka *XML* dokumendi vastavust skeemile. *Xerces 2* on aga *Java API XML* validaatorite loomiseks, mis toetab osaliselt skeem 1.1 funktsionaalsusi. Nii kaua kuni *XML* skeem 1.1 saab ametlikuks standardiks ning hakkab ilmuma ka rohkem erinevaid validaatoreid, sobib katsetusteks ning esimesteks prototüüpideks ideaalselt *Saxon-EE 9.3.0.5. Saxon Enterprise Edition*, mis nende tootevalikust ainukesena viimast skeemi toetab on küll tasuline, kuid rakenduse katsetamiseks on võimalik kompaniist tellida kolmekümnapäevane litsents. Esialgu sellisest lahendusest piisab, kuid tulevikus tuleb arvestada mõne vabavaralise alternatiiviga, mis rahuldaks ka kõik valideerimise vajadused. (Saxonica, n.d.; Apache Xerces, n.d.)

Esmaseks ideeks lugude ülesmärkimisel oli suupilli aukude tähistamine numbritega ning õhuvoolu suunast sõltuvate toonide märgistamine numbrite jadana tooni kõrgenemise järjekorras. Õhu puhumise ning tõmbamise noodid jaguneksid erinevatesse rühmadesse, ning tooni järjekorra number näitaks samaaegselt ka tooni paiknemist pillil. Selline tähistamise viis muutunuks aga liiga keeruliseks ning raskesti mõistetavaks. Peagi jõuti erinevate katsetuste läbi arusaamiseni, et kui sooviks on luua süntaks, mida saab kasutada lugude talletamiseks ning esitamiseks erinevatel suupillidel ja häälestustel, mida on Pat Missin'i andmetel üle kahesaja, tuleb suupillil paiknevad toonid ning häälestused täielikult eraldada muusikalisest informatsioonist. Loogiliselt järgnevas lahenduseks sai kahest failist koosnev süsteem. Ühes failis asub suupilliloo muusikaline informatsioon ehk mänguvõtted ning augud, millel mängitakse. Teises failis asuksid kõik võimalikud suupillide häälestused ehk suupilli ontoloogia. Ning selleks, et tabulatuurifailis kajastatud info muutuks mängitavaks, tuleb tabulatuuridele lisada viide ontoloogias paiknevale häälestusele, mida kasutati selle loo ülesmärkimiseks. Kui muusikaline info esitatakse samale pillile, mida kasutati loo ülesmärkimisel, lisatööd ei teki. Juhul kui seda lugu soovitakse esitada aga mõnel teisel pillil

või häälestusel, saab tabulatuure ümber konverteerida just ontoloogiafailis paikneva info põhjal. (Missin, 2007)

Seega paralleelselt *XML* skeemi uusima versiooni uurimisega ning tabulatuuriformaadi loomisega tuli hakata tegelema ka suupillide ontoloogia ehk tabulatuuride konverteerimise instruksiooni loomisega, mis on põhimõtteliselt kõikide võimalike suupillide häälestuste skeemid. Ontoloogia abil on võimalik muusikapalade tabulatuure konverteerida ühelt häälestuselt teisele mängitavaks. Tabulatuuriformaadi ning ontoloogia olemasolul tuleb looma hakata rakendusi ning süsteeme, mis loodud formaate toetaks. Võimalused, mis tabulatuuride kasutamisega peavad kaasnema, on nende loomine, indekseerimine metaandmetega ning tabulatuuride kõigile kättesaadavasse hoidlasse salvestamise võimalus. Suupillimängu õppijal peab aga olema ligipääs kõigile hoidlas paiknevatele lugudele. Võimalus tabulatuure otsida, vaadata ning konverteerida endale sobivasse formaati. Ning salvestatud lugusid maha mängida reaalajaliseks harjutamiseks koos visuaalsete instruksioonidega, võimalusel ka koos taustamuusikaga või videoga (vaata joonis 7). (Tomberg, 2011)



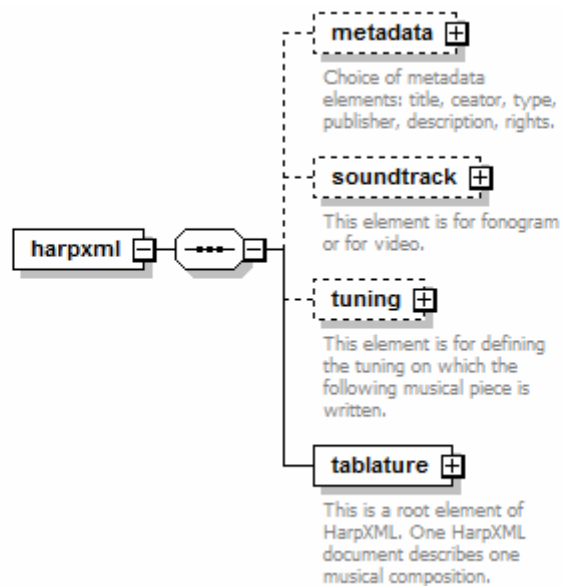
Joonis 7 Loodava tabulatuuriformaadi kasutamine. (Tomberg, 2011)

Suupilli ontoloogia info kokkukogumine kui vägagi spetsiifiline töö, jäi selle ala asjatundja Eugene Ivanovi kanda. Samal ajal kui Vladimir Tomberg ning Eugene Ivanov tegid tööd tabulatuurifaili süntaksi jaoks oluliste mänguvõtete ning pillimängu talletamist vajava info kogumisega, sai käesoleva töö autor alustada metaandmete uurimisega.

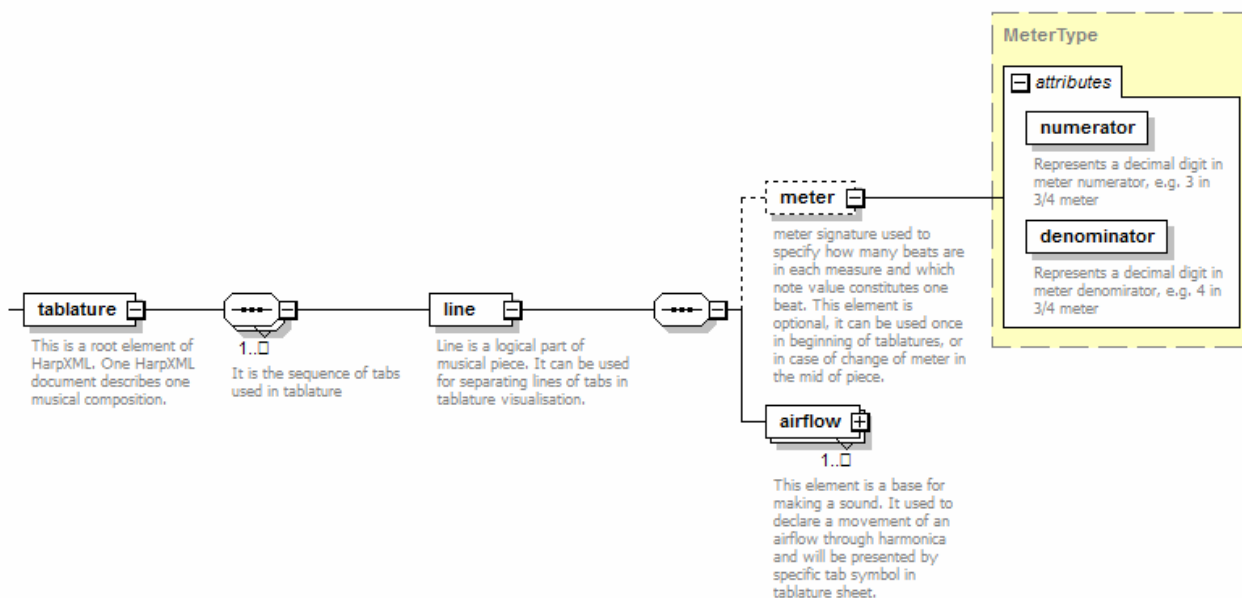
Et vältida tabulatuuriformaadi liigset koormamist, on mõistlik kasutada vaid kõige vajalikumaid metaandmeid ning neid hoida eraldiseisvana dokumendi tabulatuuride osast. Samuti peab jääma võimalus metaandmeid mitte lisada, kui tabulatuuride looja neid ei tea või ei soovi jagada. Kõige mõistlikumaks lahenduseks kujunes olemasoleva ning laialt kasutatava metaandmete süsteemi *Dublin Core* kasutamine. *Dublin Core* on metaandmete süsteemi

standard füüsiliste andmeressursside kirjeldamiseks (raamatud ning digitaalsed materjalid). *Dublin Core* moodustub kuni viieteistkümnest võimalikust elemendist kirjeldamaks kõike ühte teost puudutavat, alates tema nimest kuni õigusteni. Tabulatuuriformaadis kasutamiseks sai valitud neist kuus: pealkiri, autor, tüüp ehk žanr, levitaja ehk tabulatuurifaili looja, kirjeldus ja õigused. Kõiki elemente on võimalik kasutada ning soovi korral kõrvale jätta, samuti võib metaandmed täiesti ära jätta. Metaandmete vormistuslikku õigsust kontrollitakse aga üle interneti kättesaadava *Dublin Core* metaandmete skeemi alusel. Lisaks standardist kasutatule tuli juude lisada ka informatsiooni mida *Dublin Core* ei võimalda esitada. Nendeks on tabulatuuride video- või helisalvestusega sidumiseks mõeldud informatsioon: salvestuse ning esituse liik, salvestuse *MIME* tüüp, veebiaadress ja helistik, ning suupillimängu osa algus- ja lõpuaeg. Ning suupilli ontoloogiat puudutav info, mis kujutab endas muusikapala notatsiooni loomisel kasutatud suupilli häälestuse nime ning *ID*'d ontoloogiafailist. Nagu metaandmete puhulgi, peab kogu formaadi raames olema võimalik suupilliloo ülesmärkimiseks mitte esmatahtsaid elemente ära jätta. (Dublin Core Metadata Initiative, n.d.)

Loodav tabulatuurifaili süntaks hakkas võtma kuju juba neljas osas: metaandmed, salvestuse andmed, andmed ontoloogiaga sidumiseks ning tabulatuuride osa (vaata joonis 8). Kui valdavas enamuses oli selgunud vajalike elementide hulk ning omavahelised suhted, mis iseloomustaksid suupillimängu muusikalisi andmeid, sai asuda *XML* skeemi kokkupanemise kallale. Tabulatuuriosa põhilisteks elementideks, mis märgistavad suupilli mängimist, kujunesid elemendi *tablature* sees paiknevad õhuvoolusid tähistavad *airflow* elemendid. Hiljem töö käigus skeemi optimeerides lisandus element rea tähistamiseks, mis jagab õhuvoolud ridadeks ning muudab sellega tabulatuuri lugemise lihtsamaks. Igale tabulatuuri reale on võimalik lisada taktimõõt loo rütmi määramiseks (vaata joonis 9). Kuid põhiline info õhuvoolude suuna, mänguvõtete ning kasutatud aukude kohta paiknevad *airflow* elemendis.

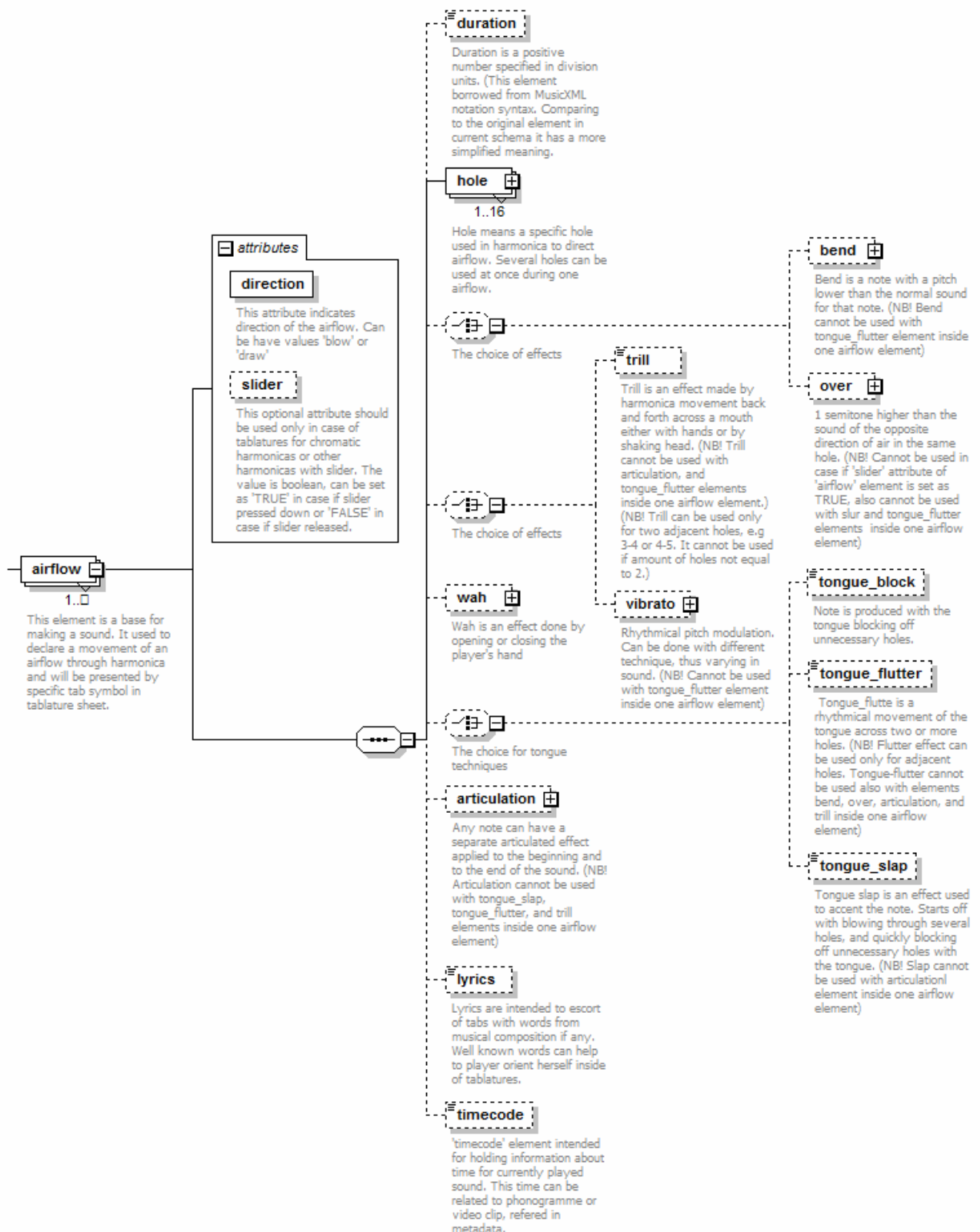


Joonis 8 Tablaturifaili XML skeemi jagunemine.



Joonis 9 'Tablature' elemendi jagunemine.

Elemendis *airflow* on kohustuslikeks elementideks vaid auk, *hole* mis määrab kasutatavad suupilli augud ning *airflow* atribuut *direction*, mis määrab õhuvoolu suuna (vaata joonis 10). Kõiki ülejäänud mänguvõtteid on võimalik kasutada vastavalt soovile, kuid järgides skeemi reegleid.



Joonis 10 'Airflow' elemendi jagunemine.

Enamik suupilli mängureegleid, mis piiravad ühte gruppi kuuluvate mänguvõtete samaaegset rakendamist saab määrata *XML*'i struktuurireeglitega. Näiteks *bend*'i või *over*'i kasutamine, *trill*'i ning *vibrato* rakendamine või erinevate keeletehnikate kasutamised on kontrollitavad *XML* skeem 1.0 struktuurireeglitega. Paigutades eelnimetatud mänguvõtted valiku alla (*choice*) saab selle elemendi järglastest korraga esineda vaid üks. Kuid üksikutele mänguvõtetele lasuvad nõuded, mis keelavad või piiritlevad mänguvõtete kasutamise lähtuvalt mõnest teisest täiesti eraldiseisvast efektist või kasutatud suupilli aukude arvust. Sellistel juhtudel tulevad abiks skeem 1.1 uued võimalused. Selliste reeglite realiseerimiseks, mis nõuavad terve dokumendi ulatuses eksisteerivate elementide kontrollimist, on tabulatuuriformaadis vajalikud *assert* elementide kasutamine (vaata koodinäide 1).

```
<xs:assert test="count(distinct-values(hole/@number)) =
    count(hole)">
    <xs:annotation>
        <xs:documentation>This rule prevents using multiple holes
            with same number in one airflow.</xs:documentation>
    </xs:annotation>
</xs:assert>
<xs:assert test="if (not((count(hole/@number) = 2) and
    ((hole[1]/@number - hole[2]/@number = 1) or (hole[1]/@number -
    hole[2]/@number = -1)))) then not(trill) else true()">
    <xs:annotation>
        <xs:documentation>This rule prevents using trill if there
            aren't used two adjacent holes in
            airflow.</xs:documentation>
    </xs:annotation>
</xs:assert>
<xs:assert test="if(hole/@slur) then not(over) else if
    (@slider eq 'true') then not(over) else true()">
    <xs:annotation>
        <xs:documentation>This rule prevents using over if there
            have been already used slur or the slider is
            'true'.</xs:documentation>
    </xs:annotation>
```

```

</xs:assert>
<xs:assert test="if(bend | over | trill | vibrato) then not
  (tongue_flutter) else true() ">
  <xs:annotation>
    <xs:documentation>This rule prevents using tongue_flutter
    if there have been already used bend, over, trill or
    vibrato.</xs:documentation>
  </xs:annotation>
</xs:assert>
<xs:assert test="if(trill | tongue_flutter | tongue_slap)
  then not(articulation) else true() ">
  <xs:annotation>
    <xs:documentation>This rule prevents using articulation
    if there have been already used trill, tongue_flutter or
    tongue_slap.</xs:documentation>
  </xs:annotation>
</xs:assert>
<xs:assert test="if(vibrato/@type eq 'tongue') then not
  (tongue_block | tongue_flutter | tongue_slap) else true() ">
  <xs:annotation>
    <xs:documentation>This rule prevents using any tongue
    technique if previously used vibrato type is
    tongue.</xs:documentation>
  </xs:annotation>
</xs:assert>
<xs:assert test="if(not((count(hole/@number) >= 2) and
  ((max(hole/@number) - min(hole/@number)) =
  (count(hole/@number) - 1)))) then not(tongue_flutter) else
  true() ">
  <xs:annotation>
    <xs:documentation>This rule prevents using tongue_flutter
    if there are not at least two adjacent holes in
    airflow.</xs:documentation>
  </xs:annotation>
</xs:assert>

```

Koodinäide 1 Suupilli mänguvõtete reeglite kontroll.

Assert elementi nõudvateks reegliteks kujunesid ühe õhuvoolu sees rakendatavate aukude mittekordumise kontrollimine. See reegel on vajalik dokumendi struktuurilise korrektsuse saavutamiseks ja ka selletõttu, et füüsiliselt on võimatu ühe õhuvoolu jooksul samasse auku puhuda rohkem kui korra. Sama õhuvoolu piires korduvate aukude leidmine osutus töö autorile sedavõrd keeruliselt realiseeritavaks, et abipalvega tuli pöörduda lausa *XML* skeem 1.1 uurija ning õppematrjalide autori Roger Costello poole. Saadud soovitude kohaselt sai kontroll-lause realiseeritud suhteliselt uut *XPath 2.0* funktsionaalsust kasutades (vaata koodinäide 2). Samuti tuli *assert* elementidega piirata mänguvõtte *trill* rakendamist juhul kui samas õhuvoolus ei kasutata kahte kõrvutiasetsevat auku, ning *overbend*'i ja *overdraw* keelamine kui eelnevalt on rakendatud mänguvõtet *slur* või kromaatilise suupilli puhul on alla vajutatud keeltevaliku nupp. Sarnane keeld eelnevalt kasutatud mänguvõtetega kehtib ka *tongue_flutter*'i ja *articulation*'i kohta. Keeletehnikate kasutamine on aga üldse keelatud juhul kui *vibrato* sooritamisel kasutatakse keelt. Viimane *assert* reegel keelab *tongue_flutter*'i kasutamise juhul kui pole kasutatud vähemalt kahte kõrvutiasetsevat auku. Viimase kontrollimiseks osutus parimaks lahenduseks esimese ning viimase augu asukoha vahe leidmine ning selle võrdlemine kasutatavate aukude arvuga (vaata koodinäide 3).

```
<xs:assert test="count(distinct-values(hole/@number)) =  
count(hole)"/>
```

```
<xs:assert test="count(hole[not(@number=preceding-  
sibling::hole/@number)]) = count(hole)"/>
```

Koodinäide 2 Suupilli aukude kordumise välistamine. Ülemine näide kasutades *XPath 2.0* funktsionaalsusi, alumine näide kasutades *XPath 1.0* funktsionaalsusi.

```
<xs:assert test="if(not((count(hole/@number) >= 2) and  
(max(hole/@number) - min(hole/@number)) =  
(count(hole/@number) - 1))) then not(tongue_flutter) else true()"/>
```

Koodinäide 3 Vähemalt kahe kõrvutiasuva augu olemasolu kontroll.

Skeemifailis mugavamaks liikumiseks ning seal eksisteerivatest elementidest parema ülevaate saamiseks sai kood jagatud lineaarse paiknemise asemel objektorjenteeritud programmeerimisvõtte alusel elemendi tüüpideks, mis asuvad eraldiseisvana faili lõpuosas.

Tabulatuurias on nähtavad kõik põhilised dokumendis rakendatavad elemendid. Uurimaks aga elementide ehitust ning nende sisu tuleb seda vaadata elemendi tüübi kirjeldusest.

3.2 Suupilli ontoloogia

Ontoloogia väljatöötamisega hakkas lähemalt tegelema Eugene Ivanov kui selle ala spetsialist. Kuna Pat Missin'i andmed pärinevad aastast 2007 ning on mõnevõrra vananenud, tuleb kokku koguda uus suupilli ontoloogia vastavalt hetkel kasutuses olevatele suupilli häälestustele. Kogutud info alusel saab moodustada ontoloogiafaili, kus asuvad kõik enimkasutatud suupillid koos võimalike häälestustega. Iga häälestust hakkab tähistama tema nimi ning *ID*, millega neid seotakse tabulatuurifailidega. (Missin, 2007)

XML'ina loodud ontoloogiafaili on tulevikus võimalik vastavalt vajadusele ka täiendada ning parandada. Esialgu hakkab ontoloogia sisaldama diatoonilise ning kromaatilise suupilli erinevaid häälestusi. Ontoloogia täiendamise lihtsustamiseks on võimalus välja töötada ka selleotstarbeline rakendus.

Suupilli ontoloogia hakkab ülesehituselt sisaldama vaid suupilli aukude baastoone. Seega kümne auguga suupilli kohta 20 tooni, pooled tekitatuna õhu puhumisest, teised õhu tõmbamisest. Kõigi keerulisemate mänguvõtete alla kuuluvate *bend*'ide, *overbend*'ide ning *overdraw*'de olemasolud on arvatavad kasutatava häälestuse baasil (vaata koodinäide 4). Selleks võrreldakse suupilli ühte auku õhu puhumisel ning õhu tõmbamisel tekkivaid tooni erinevusi, mille järgi määratakse *bend*'ide ning *overbend*'ide ja *overdraw*'de asukohad pillil. Kasutajaliideses kuvatakse mängijale *bend*'idena, *overbend*'idena ning *overdraw*'dena vaid neid toone, mida ei ole võimalik suupilli aukudel loomulikult viisil mängides saada.

```
if(blow - draw) <= 0 then harmonica draw side has bends and blow
    side has overbends.
if(blow - draw) >= 0 then harmonica blow side has bends and draw
    side has overdraws.
```

Koodinäide 4 *Bend*'ide, *overbend*'ide ning *overdraw*'ide arvutamise põhimõte.

Ontoloogia peab kajastama kõiki suupilli häälestusvõimalusi, kaasaarvatud niinimetatud ventiilide (*valves*) kasutamist, mis väldivad õhukadu mitterängitava pillikeele ava kaudu.

Ventiilide kasutamine välistab *overbend*'ide ning *overdraw*'de kasutamist, vähendades nende helikõrgust. Kuna ka ventiilide kasutamisest tingitud toonimuutusi saab suupillil arvutada, ei ole vaja neid eraldi ontoloogiafailis märkida. Küll aga tuleb loo juures ära märkida, olukord kui ventiile on kasutatud.

3.3 Tabulatuuriformaadi rakendamine

Kuigi tabulatuuriformaat näib ise suuremas osas valmis olevat, ei katke sellega veel testimine ning vajadustel parenduste läbiviimine. Järgmiseks eesmärgiks suupillimängu notatsiooniprojekti valmimise suunas on esimese prototüübi valmistamine ning tööle saamine. Kuigi tööd võiks koheselt alustada ka ontoloogiafaili loomisega on siiski mõistlikum luua enne üks läbinisti töötav üksus kui mitu reaalist veel mittekasutatavat elementi. Seega esimeseks prototüübi katseliseks eesmärgiks jääb tabulatuurifailide loomine, failide avamine, võimalus neid üle võrgu valideerida ning valideeritud dokumendi esialgu lihtsakoeline visualiseering (vaata joonis 11). Hilisemas arendusfaasis lisanduvad prototübile andmebaas tabulatuuride talletamiseks ning ontoloogia alusel dokumentide konverteerimine teistesse häälestustesse. Koos tabulatuurifailide hoidlasse lisamise ning sealt kättesaamisega tuleb samaaegselt tööle saada ka metaandmete alusel dokumentide indekseerimine ning otsimine, et edaspidine ligipääs tabulatuurifailidele oleks kasutajale võimalikult lihtne ning kiire.

Kui projekti põhifunktsionaalsused, mis hõlmavad endas notatsioonide loomist ning kättesaamist, on valmis ning töötavad vigadeta, saab tööd alustada rakenduste välise kasutajaliidese kallal. Kasutajapoolseid rakendusi saab ning peab süsteemile looma mitmeid erinevaid. Esiteks eraldi ligipääs peab olema tabulatuuride loojatel ehk niinimetatud õpetajatel, kes lisavad hoidlasse notatsioone. Esimene prototüüp saabki suuresti olema notatsioonide loomise katsetamiseks. Lihtsam rakendus, mis võimaldab vaid tabulatuure otsida ning neid vaadata ja konverteerida on mõeldud pillimängu õppijale. Kolmas kasutajaliidese tüüp on ontoloogia täiendamiseks, millele ligipääs peab olema tagatud vaid selleks volitatud isikutel.

- Lines

Use Meter / + Add New Line

- Tabs

Duration / 4

↑ Blow	C	E	G	C	E	G	C	E	G	C
	1	2	3	4	5	6	7	8	9	10
↓ Draw	D	G	B	D	F	A	B	D	F	A
Slur	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- Bends

Bend	0	0	0	0	0	0	0	0	0	0
Over	0	0	0	0	0	0	0	0	0	0
Glide-in	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- Effects

Trill

Vibrato

Tonque

Articulation

- Additional

Timecode H M S

Lyrics

Enter tab/s

Visualisation

Joonis 11 Prototüübi kasutajaliidese esialgne kavand.

Kui kõik põhifunktsionaalsused ning nende kasutamine on võimaldatud jääb edasisteks eesmärkideks tabulatuuride esitamise funktsionaalsuse arendamine. Tabulatuuride esitamise funktsioon on mõeldud eelkõige suupillimängu õppeks sünkroonselt kas video või helisalvestusega. Õppijale kuvatakse soovitud muusikapala visuaalne salvestus või lihtsalt loo taustamuusika ning ekraanil kuvatakse reaajas mängitavad suupilli augud koos rakendatavate mänguvõtetega.

Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli suupillimängu notatsiooniprojekti raames rakendatava tabulatuuriformaadi parima reliseerimisviisi leidmine. Selle formaadi lihtsa ning loogilise indekseerimise väljatöötamine tulevastele rakendustele võimalikult optimaalseks töötlemiseks ning kasutajapoolse kättesaadavuse tagamiseks. Samuti standardi koostamine kõikide suupilli mängureeglite ning –võtete õigsuse järgimisega tabulatuurifailide loomiprotsessiks.

Optimaalseima noteerimisviisi otsingute käigus töötati välja ka esimene tabulatuuriformaadi standard mis on ka käesoleva bakalaureusetöö praktiliseks väljundiks. Valminud tabulaatuuriformaat lubab talletada suupillimänguks vajalikku muusikalist informatsiooni koos kõigi võimalike mänguvõtetega enimkasutatud suupilli tüüpidel. Koos muusikalise informatsiooniga saab salvestada andmeid kasutatud suupilli liigi ja häälestuse kohta ning lisada dokumendile metaandmeid ja viidata samasisulisele internetis paiknevale video- või helisalvestisele.

Bakalaureusetöö tulemusena kogutud teadmisi ning loodud standardit rakendatakse suupillimängu notatsiooniprojekti edasises arendustöös.

Valminud tabulatuuriformaadi *XML* skeem ning *XML* dokumendi näide koos tabulatuuriformaadi dokumentatsiooniga asuvad töö lisana *CD* plaadil. Samuti saab lisainfot ka veebiaadressilt <http://www.tlu.ee/~raido89/harpXML/>, kuhu projekti edasistes faasides ilmuvad esimesed tabulatuuriformaati kasutavad prototüübid.

Summary

XML Vocabulary for Harmonica Tablatures

Due to the lack of a unified standard in a field of describing harmonica tablatures, every musician marks down one's tablatures in a slightly different way. For novice harmonica players who have just started learning the instrument, a wide variety of tablature types is often very confusing. This situation is exacerbated by even greater variety of harmonica tunings.

This bachelor's thesis focuses on the process of developing suitable *XML* vocabulary for describing harmonica tablatures. The main goal of this work is to simplify the novice harmonica player's learning process by creating a standard for unitary way of describing the harmonica tablatures. By describing only the musical information of certain musical piece the user can choose what kind of tablature visualisation he prefers and can even convert tablatures to different tunings.

The main goals of this vocabulary is to make it possible to describe every kind of harmonica composition and every possible playing technique, to enable the conversion of tablatures and to make searching and indexing tablature files easier. An initial version of this *XML* syntax is the basis for the further development of learner friendly harmonica tablature applications.

Kasutatud kirjandus

1. Adobe. (n.d.). Adobe Flash Professional CS5.5. Viimati vaadatud 25. aprill 2011, aadressil <http://www.adobe.com/products/flash.html>
2. Alekseenko, S. (2007). Harp Keys. Viimati vaadatud 25. aprill 2011, aadressil <http://harmonica.ru/content/catalogs/soft/harpkeys.swf>
3. Amiano, M., D'Cruc, C., Ethier, K., Thomas, M.D. (2006). XML Problem – Design – Solution. Indianapolis, IN: Wiley Publishing, Inc.
4. Apache Xerces. (n.d.). Xerces2 Java Parser Readme. Viimati vaadatud 25. aprill 2011, aadressil <http://xerces.apache.org/xerces2-j/>
5. Baker, S. (1999). The Harp Handbook. [image], Music Sales America.
6. Ball, T. (2000). Sourcebook of Little Walter/Big Walter Licks for Blues Harmonica. [image], Centerstream Publishing.
7. Ball, T. (1995). The Sourcebook of Sonny Terry Licks for Harmonica. [image], Centerstream Publishing.
8. Barret, D. (1997). Building Harmonica Technique. [image], Mel Bay Publications, Inc.
9. Barry, B. (2006). Beginner Blues Harmonica. [image], Harmonicast.
10. Benz, B., Durant, J.R. (2003) XML Programming Bible. New York, NY: Wiley Publishing, Inc.
11. Costello, R.L. (2010). XML Schema 1.1. Viimati vaadatud 28. aprill 2011, aadressil <http://www.xfront.com/xml-schema-1-1/xml-schema-1-1.ppt?v=13>
12. Cover, R. (2006). XML and Music. Cover Pages, veebruar. Viimati vaadatud 25. aprill 2011, aadressil <http://xml.coverpages.org/xmlMusic.html>
13. Dublin Core Metadata Initiative. (n.d.). Viimati vaadatud 25. aprill 2011, aadressil <http://dublincore.org/>

14. Duncan, P. (1996). Blues Harp for Diatonic and Chromatic Harmonicas. [image], Mel Bay Publications, Inc.
15. Gelling, P. (2000). Teach Yourself Blues Harmonica - 10 Easy Lessons. [image], L.T.P. Publishing.
16. Gräfe, A. (n.d.). Lick Transposer. Viimati vaadatud 28. aprill 2011, aadressil <http://harptools.andreas-graefe.de/siteht/transpose/LickTransposer.aspx>
17. Harptabs. (n.d.). Viimati vaadatud 25 aprill 2011, aadressil <http://www.harptabs.com/>
18. Ivanov, E., Grigorov, S., Ladutin, K. (n.d.). Harp-o-matic. Viimati vaadatud 28. aprill 2011, aadressil <http://www.truechromatic.com/harp-o-matic.php>
19. Kyrnin, J. (n.d.). What are Markup Languages? Viimati vaadatud 25. aprill 2011, aadressil <http://webdesign.about.com/od/htmlxhtmltutorials/p/what-are-markup-languages.htm>
20. McCoy, C. (1998). All-American Harp. [image], Wise Publications.
21. Missin, P. (2007). Tuning and the Harmonica. Viimati vaadatud 25. aprill 2011, aadressil <http://patmissin.com/tunings/tunings.html>
22. Molino, M.W. (2007). Musical Notation and Diatonic Harmonica Tablature of Natural Notes of All 12 Keys. Viimati vaadatud 25. aprill 2011, aadressil <http://www.harptabs.com/displayfile.php?ID=21>
23. Mozilla Developer Network. (2011). Canvas tutorial. Viimati vaadatud 25. aprill 2011, aadressil https://developer.mozilla.org/en/canvas_tutorial
24. Myriad Software. (2004). Tablatures: Harmonica Tablatures. Viimati vaadatud 25. aprill 2011, aadressil <http://www.myriad-online.com/resources/docs/manual/english/tabharmonica.htm>
25. Power, B. (2000). Play Irish Music on the Blues Harp. [image]
26. Recordare. (n.d.). MusicXML. Viimati vaadatud 25. aprill 2011, aadressil <http://www.recordare.com/musicxml>

27. Recordare. (n.d.). MusicXML specification. Viimati vaadatud 25. aprill 2011, aadressil <http://www.recordare.com/musicxml/specification/xsd>
28. Rost, M. (1993). Rock Blues Country Harp. [image], Voggenreiter Verlag OHG.
29. Saxonica. (n.d.). Viimati vaadatud 25. aprill 2011, aadressil <http://www.saxonica.com/welcome/welcome.xml>
30. Stewart, D. (2003). XML for Music. Electronic Musician, detsember. Viimati vaadatud 25. aprill 2011, aadressil http://emusician.com/mag/desktop/emusic_xml_music/
31. Tomberg, V. (2011). Exploring the Potentials of the Dedicated XML Syntax for Diatonic Harmonica Tablatures. In Proceedings of ICMC-2011 conference, Huddersfield.
32. W3C The World Wide Web Consortium. (n.d.) About SVG: 2d Graphics in XML. Viimati vaadatud 25. aprill 2011, aadressil <http://www.w3.org/Graphics/SVG/About.html>
33. W3C The World Wide Web Consortium. (n.d.). Schema. Viimati vaadatud 25. aprill 2011, aadressil <http://www.w3.org/standards/xml/schema>
34. W3C Ubiquitous Web Domain. (n.d.). XML Schema. Viimati vaadatud 25. aprill 2011, aadressil <http://www.w3.org/XML/Schema>
35. w3schools. (n.d.). Flash Tutorial. Viimati vaadatud 25. aprill 2011, aadressil <http://www.w3schools.com/flash/default.asp>
36. w3schools. (n.d.). XML Schema Tutorial. Viimati vaadatud 25. aprill 2011, aadressil <http://www.w3schools.com/schema/default.asp>
37. w3schools. (n.d.). XSLT Tutorial. Viimati vaadatud 25. aprill 2011, aadressil <http://www.w3schools.com/xsl/default.asp>