

НАМЈЕНСКИ ЈЕЗИК И ОКРУЖЕЊЕ ЗА МОДЕЛОВАЊЕ НОТНОГ ЗАПИСА И
ГЕНЕРИСАЊЕ СПЕЦИФИКАЦИЈА ЗА МУЗИЧКИ СОФТВЕРA DOMAIN SPECIFIC LANGUAGE AND A FRAMEWORK FOR MUSICAL NOTATION
MODELING AND GENERATING MUSIC SOFTWARE SPECIFICATION

Оливера Благојевић, Факултет техничких наука, Нови Сад

Област – ПРИМЈЕЊЕНЕ РАЧУНАРСКЕ НАУКЕ
И ИНФОРМАТИКА

Кратак садржај – У овом раду је описан намјенски језик и окружење за моделовање нотног записа, као и израда генератора и генерисање спецификација за музички софтвер. За креирање језика је коришћена текстуална конкретна синтакса. Осим језика, креирани су и генератори, који имају за циљ генерисање фајлова формата који одговарају софтверима за креирање и читање нотног система. Креирани намјенски језик поједностављује рад са софтверима за компоновање музике и чини писање нотних система знатно лакшим. За развој метамодела коришћено је окружење Eclipse Modeling Framework, а за дефинисање ограничења модела коришћен је OCL језик опште намјене. За креирање текстуалне конкретне синтаксе коришћен је радни оквир Xtext, а за генераторе програмски језик Xtend.

Кључне ријечи: музика, музичка нотација, нотни запис, намјенски језици, развој софтвера базиран на моделу

Abstract – The paper describes domain-specific language and framework for music score modeling, as well as generator design and generating specifications for music software. Textual concrete syntax was used for creating the language. Besides the language, generators were created, which goal is to generate files that have formats that fit sheet music creation and reading softwares. Created domain-specific language simplifies the work with softwares for music composing and makes writing sheet music much easier. Eclipse Modeling Framework was used for metamodel development as well as general purpose OCL language to define model constraints. Xtext framework was used to create textual concrete syntax, and Xtend programming language was used for generators.

Keywords: music, music notation, sheet music, Domain-Specific Languages, Model-Driven Software Development

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Милан Челиковић, доцент.

1. УВОД

Нотни запис је ручно записана или одштампана форма музичке нотације, која користећи музичке симболе означава тонове, ритмове или акорде пјесме или инструменталног музичког дјела. На енглеском језику нотни запис музике је „sheet music“, јер је употребом термина „sheet“ (срп. лист) намјера била направити разлику између писаних или штампаних музичких форми и звучних записа. Нотни запис је основни облик у коме се класична музика биљежи тако да је могу учити и изводити соло пјевачи, инструменталисти, или музички ансамбли.

Ручно писање нотних записа углавном одузима много времена, подложно је грешкама и понекад непрегледно. Да би се потешкоће приликом ручног писања нотног записа избјегле, данас на интернету постоје различити музички софтвери. Међутим, уколико корисник музичког софтвера није унапред упознат са софтвером или није довољно вјешт приликом коришћења савремених технологија, ови софтвери могу бити веома компликовани за коришћење и нејасни.

Истраживањем и анализом музичких софтвера је утврђено да сви музички софтвери омогућавају увоз и извоз фајлова који садрже записану музичку нотацију у одређеном, изабраном, формату. Могуће је покренути музички софтвер, учитати фајл формата који одређени софтвер подржава и добити исписан нотни систем, без коришћења софтвера за исписивање нотног записа. Софтвер се након увоза фајла може искористити за додатне опције и за репродукцију композиције, а измјењени фајл се потом поново може сачувати у жељеном формату. На овај начин би се уштедило вријеме за исписивање музичких нотација, а предности музичких софтвера би се и даље могле искористити. Циљ овог рада је проналажење начина за генерисање фајлова који садрже жељени нотни запис и који могу бити увезени у музичке софтвере и тиме рјешење могућих проблема приликом употребе музичких софтвера. Да би се ово остварило, креирани су намјенски језик за моделовање нотног записа, назван *MusicDSL*, и њему одговарајући генератори, који у себи садрже спецификације за музички софтвер.

Поред уводног поглавља и закључка, овај рад садржи још шест поглавља. Поглавље *Преглед постојећег стања у области* представља преглед доступних музичких софтвера и преглед формата које фајлови ових програма користе, а које *MusicDSL* генерише. У

оквиру поглавља *Теоријске основе намјенског језика и трансформација модела* су дати основни појмови и концепти за спецификацију намјенских језика и теоријске основе трансформација модела. Поглавље *Опис коришћених технологија* даје опис технологија које су коришћене за креирање *MusicDSL* намјенског језика и генератора. У *Архитектури система* су приказани дијелови описаног система. У поглављу *Намјенски језик за моделовање нотног система* је дат детаљан опис концепата и креирања *MusicDSL* намјенског језика. Поглавље *Генерисање фајлова* представља изглед генератора, опис тока генерисања и коначни резултат.

2. ПРЕГЛЕД ПОСТОЈЕЋЕГ СТАЊА У ОБЛАСТИ

Брзи развој десктоп рачунара 1980-их година је допринио стварању много софтвера за писање нотних записа. То је тада био много јефтинији начин креирања нотних записа и дијелова за оркестралну музику, који су користили млади композитори, музички педагози и студенти композиције. Нотни записи у тадашњим музичким софтверима су били читљиви али нису изгледали професионално и због тога их комерцијални издавачи музике нису користили. Током 1990-их година већина ових софтвера је изашла из употребе, јер су се појавили нови, лакши за коришћење и бољег квалитета записа. Развојем *Windows* оперативног система појављивало се све више музичких софтвера, са опцијама увоза, измјене и штампања музичке нотације, који могу да испишу једноставну композицију, комад за одређен инструмент, или комплексну оркестралну музику, високог квалитета записа. Скоро сви музички софтвери користе своје формате фајлова за чување креираних композиција. Да би се креиране композиције могле користити у различитим софтверима, већина музичких софтвера дозвољава увоз и извоз фајлова у стандардним форматима за размјену фајлова.

2.1. Формати за размјену дигиталних нотних записа

Као што је претходно наведено, сви софтвери за креирање и читање нотних записа имају опцију увоза и извоза музичких нотација у фајлове одређених формата, који се касније могу користити у неком другом софтверу, чиме је обезбијеђена једноставна размјена дигиталних нотних записа. Најчешће омогућен формат дигиталног нотног записа је *MusicXML*. Скоро сви софтвери подржавају увоз овог формата, али не и извоз. Неки софтвери дозвољавају чување нотних записа само у *XML* формату фајла (*Noteflight*) или *MSCX* (*MuseScore*).

Extensible Markup Language (XML) или прошириви мета језик за означавање текстуалних докумената, је стандардни скуп правила за дефинисање формата података у електронској форми прописан од стране *World Wide Web* групе (*W3C*). Слиједећи правила *XML* стандарда корисници дефинишу сопствене формате података, које могу користити за њихово складиштење, обраду и размјену. *XML* документи имају хијерархијску структуру и концептуално се могу тумачити као структура стабла, која се назива

XML стабло. *XML* документи морају да садрже основни елемент („родитељ“ свих осталих елемената). Сви елементи у *XML* документу могу да садрже поделене, текст и атрибуте. Стабло представљено *XML* документом почиње од основног, коријенског, елемента и грана се до најнижег нивоа елемената [1]. *MusicXML* је стандардни отворени формат за размјену дигиталних нотних записа, који се може бесплатно користити уз *W3C* спецификацију. Дизајниран је за дијељење нотних записа између апликација и за архивирање таквих фајлова за будућу употребу. Од датума издавања овај формат подржава преко 200 апликација [2].

MSCX је формат дигиталног нотног записа креиран за *MuseScore*. Уведен је тек у верзију *MuseScore* 0.9.5, јер је прије тога било тешко ријешити конфликт са повезивањем фајла у *Windows*-у. *MSCX* фајлови чувају некомпресовану верзију нотних записа. Компресована верзија се чува у *MSCZ* фајловима, што је стандардни формат за *MuseScore* композиције. *MSCX* фајлови могу бити преведени у двадесет језика у *MuseScore*-у. [3] Због тога што је *MSCX* некомпресовани текстуални формат, може једноставно ручно бити уређиван у било ком текстуалном едитору.

2.2. Преглед доступних софтвера за креирање нотних записа

Данас су софтвери за креирање нотних записа потпуно доступни и постали су неопходни скоро као што су софтвери за унос и обраду текста. На тржишту их има много, а овдје ће бити подијељени у три групе: *offline* софтвери, *online* софтвери и апликације. Биће наведени они који су бесплатни и који дозвољавају увоз и извоз фајлова претходно поменутих формата.

Међу *offline* софтверима најпознатији су: *MuseScore*, *Improvvisor*, *Sibelius*, *Denemo*, *Anvil Studio* и *IBOS MusicXml Reader*.

У групи *online* софтвера најпознатији су: *Noteflight* и *Flat*. Од апликација тренутно су доступне: *forScore* и *Orpheus Sheet Music Pro*.

3. ТЕОРИЈСКЕ ОСНОВЕ НАМЈЕНСКОГ ЈЕЗИКА И ТРАНСФОРМАЦИЈЕ МОДЕЛА

Језик представљен у раду је намјенски, а имплементиран је захваљујући трансформацијама из модела у текст. Како би поступак израде рјешења за проблем описан у раду био јаснији, а самим тим и циљ рада, потребно је дати теоријске основе ових појмова.

Језици специфични за домен (енг. *Domain-Specific Modeling Languages, DSL*) су дизајнирани посебно за одређен домен, контекст, или компанију, како би олакшали задатке људима који треба да опишу ствари у том домену. Ако је језик усмјерен на моделовање назива се још и доменски специфичним језиком за моделовање (енг. *Domain-Specific Modeling Language, DSML*). *DSL* се користио у рачунарској науци много прије него што је сам акроним настао. Примјери оваквих језика су: *HTML* језик за израду *web* страница, *Logo* за једноставно цртање, *VHDL* за описивање хардвера, *Mathematica* и *MatLab* за математику, *SQL* за приступ базама

података, итд. [4] Језик специфичан за домен је намијењен за употребу у оквиру конкретно дефинисаног домена. Сваки намјенски језик је дефинисан кроз три кључна елемента:

- Апстрактна синтакса описује структуру језика и начин на који се различити примарни елементи могу међусобно комбиновати, независно од било које конкретне репрезентације или кодирања.
- Конкретна синтакса описује одређену репрезентацију језика моделовања, укључујући кодирање и/или проблеме визуелне презентације. Конкретна синтакса може бити: текстуална – подаци се кодирају низом карактера, као код програмских језика и графичка конкретна синтакса – подаци се кодирају пропсторним распоредом графичких и текстуалних елемената.
- Семантика – описује значење елемената дефинисаних у језику и значење различитих начина комбиновања ових елемената.

Моделу нису и изоловани ни статички ентитети. Као дио процеса инжењеринга заснованог на моделу, модели су: обједињени, поравнати, преправљени и пречишћени. Све ове операције над моделима су имплементирание као трансформације модела. Могу бити трансформације из модела у модел (енг. *Model-To-Model*, *M2M*) или трансформације из модела у текст (енг. *Model-To-Text*, *M2T*). У *M2M* трансформацијама улазни и излазни параметри трансформације су модели, док је у *M2T* трансформацијама улазни параметар модел, а излазни текст.

4. ОПИС КОРИШЋЕНИХ ТЕХНОЛОГИЈА

Изабрани начин креирања намјенског језика за моделовање нотног система је креирање намјенске текстуалне синтаксе на основу метамодела. Помоћу *M2T* трансформација се аутоматизује генерисање музичких фајлова креирањем шаблона. У наставку овог поглавља су описане технологије којима је постигнуто наведено.

За креирање метамодела, његове динамичке инстанце и генерисање кода из модела коришћен је *Eclipse*-ов радни оквир за моделовање (енг. *Eclipse Modeling Framework*, *EMF*). *EMF* нуди могућности генерисања кода за креирање алата и других апликација базираних на структурисаном моделу података. Из спецификације модела описане у *XMI* формату, *EMF* нуди и алате и *runtime* подршку за израду скупа *Java* класа из модела, заједно са скупом додатних класа које омогућавају читање и уређивање модела на основу команде, и основни едитор.

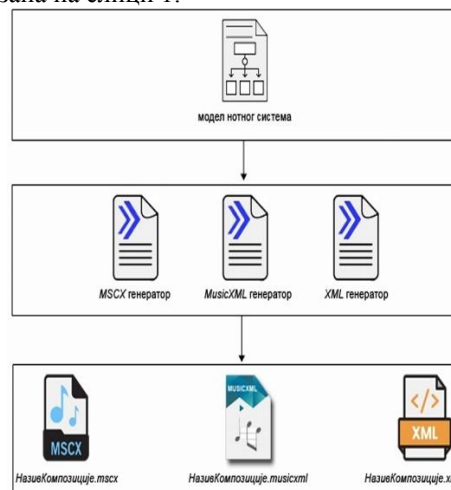
У метамоделу рађеном у *EMF Ecore*-у није било могуће представити сва ограничења намјенског језика, нека су требала бити додатно дефинисана, и за то је коришћен језик за ограничења објеката (енг. *Object Constraint Language*). Правила за ограничења су писана на нивоу апстрактне синтаксе, како би сама имплементација ограничења била једноставнија.

Текстуална граматика намјенског језика је аутоматски генерисана, а затим прилагођена и генерисани су њени артефакти у *Xtext* радном оквиру. *Xtext* аутоматизује креирање текстуалних едитора намјенских језика у виду *Eclipse plugin*-а. Коришћена је опција креирања пројекта на основу постојећег модела (енг. *Xtext Project from Existing Ecore Models*), како би се могле користити инстанце креираног *Ecore* метамодела.

За реализацију генератора који креирају .xml, musicxml и .mscx фајлове коришћен је програмски језик *Xtend*. *Xtend* је програмски језик вишег нивоа генералне намјене за *Java* виртуелну машину. Синтактички и семантички има коријене у *Java* програмском језику али се фокусира на сажетију синтаксу и неке додатне функционалности као што су: детекција типа података, методе проширивања (енг. *extension method*) и *operator overloading* (различити оператори имају различите имплементације, у зависности од њихових аргумената).

5. АРХИТЕКТУРА СИСТЕМА

Први корак израде софтверског рјешења описаног у раду је дефинисање архитектуре жељеног система, а затим креирање система који одговара дефинисаној архитектури. Почетна ставка архитектуре система је модел нотног система, који се састоји од свих елемената нотног система који су потребни за генерисање жељених фајлова. На основу модела је креиран и намјенски језик *MusicDsl*, а потом су направљени генератори помоћу којих добијамо фајлове жељених формата. Архитектура система је приказана на слици 1.



Слика 1: Архитектура система

Систем се састоји од: модела нотног система, на основу кога су креирани генератори (*MSCX*, *MusicXML* и *XML*), који генеришу фајлове истоимених формата.

6. НАМЈЕНСКИ ЈЕЗИК ЗА МОДЕЛОВАЊЕ НОТНОГ ЗАПИСА

Коријенски концепт апстрактне синтаксе је композиција (*Composition*), гдје корисник касније уноси уопштене информације о композицији коју креира (назив, аутор, инструмент за који је

предвиђена, итд). Композиција у себи садржи линијске системе (*Staff*), може имати највише два и то је ограничење које је било могуће намјестити у *EMF*-у. Сваки линијски систем садржи кључ (*Clef*), тактну вриједност (*TimeSignature*) и тоналитет (*Key*). Код неких софтвера за креирање нотног система (нпр. *Flat.io*) је могуће само једном унијети тактну вриједност и тоналитет и то се не мијења, остаје исто за оба кључа.

Код других софтвера за креирање нотног система за сваки линијски систем се те вриједности поново уписују (нпр. *Sibelius*), тако да су приликом генерисања шаблони прилагођавани томе. Осим ова три елемента, које линијски систем мора имати, он такође садржи и апстрактну класу са својим осталим елементима (*StaffElement*) коју наслеђују такт (*Measure*), пауза (*Rest*), понављање (*Repetition*) и нота (*Note*). Типови ових елемената су наведени у еnumerацији. Нота сама за себе садржи украсе, своју артикулацију и везе, који се налазе унутар концепта нотација (*Notation*).

Текстуална конкретна синтакса је креирана тако што је прво *Xtext* генерисао граматiku на основу коријенске класе *Composition*, а затим је граматика прилагођена жељеном изгледу језика. Приликом прилагођавања, промијењено је сљедеће:

- већина витичастих заграда је замијењена угластим;
- почетак такта се означава именом и редним бројем такта након којег слиједи ознака *begin*, а крај се означава ознаком *end*;
- вриједности еnumerација се уносе без испред унијетог назива;
- називи веза су уклоњени, јер нема потребе да их корисник уноси;
- промијењене су неке кључне ријечи, највише када је концепт *Note* у питању.

```
Composition "Yesterday" [
  author: "John Lennon and Paul McCartney"
  instrument: "Piano"
  tonality: "F major"
  soundTempo: 80
  Staff 1 [
    clef: violin
    key: -1
    time: beat 4, beatType 4
    Measure 1 begin
    note G :
      type eighth
      duration 1
      octave 4
      voice 1
      lyrics "Yes"
  end,
```

Листинг 1: Примјер почетног дијела модела нотног система композиције *Yesterday*

7. ГЕНЕРИСАЊЕ ФАЈЛОВА

За имплементацију генератора потребно је било направити *Xtext* шаблоне који одговарају спецификацији музичких софтвера и структури фајлова формата за размјену дигиталне музичке нотације. То значи да су осим шаблона који одговарају спецификацији музичких софтвера генерисане и функције за претраживање и

препознавање линијских система, тактова, типова музичких елемената, итд.

Из *MusicDslGenerator* генератора се позивају и покрећу *MusicXML*, *XML* и *MSCX* генератори. Генерисан је аутоматски приликом генерисања *Xtext* артефакта и касније прилагођен. Основни елемент по коме се филтрира и врши претрага осталих елемената је *Composition*. Елемент *Composition* је коријенски елемент и због тога једини из кога се може приступити свим осталим елементима, јер су унутар њега.

8. ЗАКЉУЧАК

У раду је описан намјенски језик *MusicDsl* који омогућава креирање фајлова који се могу учитати у музичке софтвере, без претходног програмерског предзнања корисника, довољно је само да познаје синтаксу језика. Језик посједује концепте којима је детаљно описан цијели нотни систем, тј. сви елементи који су неопходни за писање једне композиције. Из њега су даље креирани генератори, који након корисниковог уноса нотног записа путем намјенског језика генеришу фајлове жељених формата.

Намјенски језик *MusicDsl* би могао бити још унапређен, у зависности од тога у ком смјеру ће се мијењати музички софтвери. Иако ови софтвери упозоравају ако је у учитаном фајлу превише или премало нота у једном такту у односу на тактну вриједност, у будућности би се могла додати валидација унутар конкретне синтаксе, тако да корисник у старту не би могао унијети погрешан број тактних елемената. Ово би било веома корисно, али је тешко реализовати јер је подложно грешкама уколико се не наведу све комбинације. Апстрактна синтакса би вјероватно могла бити још проширена, уколико софтвери почну додавати још неке музичке елементе у своје опције. Ако корисник не уноси различите ноте за другу врсту кључа, онда би нека врста преводиоца из кључа у кључ такође била добра идеја за унапређење језика. Када би се појавили нови софтвери на тржишту, са новим форматима представљања музичких елемената у фајлу, израда још једног генератора би била неопходна.

ЛИТЕРАТУРА

- [1] *XML*, *Wikipedia* - <https://en.wikipedia.org/wiki/>
- [2] *MusicXML*, документација - <https://www.w3.org/2017/12/musicxml31>
- [3] *MuseScore*, документација - <https://musescore.org/en/handbook>
- [4] M. Brambilla, J. Cabot, and M. Wimmer, *Model-driven software engineering in practice*, Second edition. San Rafael, Calif.: Morgan & Claypool Publishers, 2017.

Кратка биографија:



Оливера Благојевић рођена је 1994. године у Добоју, у Босни и Херцеговини. Дипломски рад из области информacionих технологија одбранила је 2017. године на Слобомир П Универзитету. Мастер студије на Факултету техничких наука је уписала 2019. године.