

# Maturitní práce

Informatika

Aplikace pro syntézu hudby z formátu MusicXML do  
formátu SMF (MIDI) pomocí GNU Octave

Alena Smutná

R8.A

Gymnázium Jana Keplera

13. března 2019

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Použité technologie</b>	<b>2</b>
2.1	XML . . . . .	2
2.1.1	MusicXML . . . . .	2
2.1.2	DOM, Apache Xerces . . . . .	2
2.2	MIDI . . . . .	3
2.2.1	General MIDI . . . . .	3
2.2.2	SMF . . . . .	3
2.3	GNU Octave . . . . .	3
<b>3</b>	<b>Teoretická část</b>	<b>4</b>
<b>4</b>	<b>Implementace</b>	<b>5</b>
<b>5</b>	<b>Testování</b>	<b>5</b>
<b>6</b>	<b>Návod k použití</b>	<b>5</b>
<b>7</b>	<b>Závěr</b>	<b>6</b>
	<b>Literatura</b>	<b>7</b>

# 1 Úvod

Tato práce se zabývá převodem hudebních skladeb ve formátu MusicXML (textový formát používaný pro přenos skladeb mezi programy pro tvorbu hudebních skladeb) na zvukový soubor ve formátu MIDI. Program je psán v jazyce GNU Octave, s využitím funkcí knihovny Apache Xerces2 Java Parser pro načítání a parsování XML dokumentu a funkce `xml2struct` z Mathworks File Exchange pro jeho konverzi z XML Document Object Model do datového typu struktura.

## 2 Použité technologie

### 2.1 XML

XML, nezkráceně *eXtensible Markup Language*, je značkovací jazyk, specifikovaný společností W3C, který je strojově zpracovatelný, a zároveň je čitelný pro lidi. Užívá se především pro přenos dat mezi různými aplikacemi a pro uchovávání dat, u kterých je důležitá struktura a obsah jednotlivých částí. XML dokumenty jsou textové dokumenty užívající kódování Unicode. Specifikace XML formátu nedefinuje jednotlivé značky, ty jsou definovány buď v jednotlivých konkrétních aplikacích (např. XHTML, RSS, SVG nebo MusicXML), nebo se dá definovat vlastní sada značek pomocí tzv. definičních jazyků (např. DTD).

#### 2.1.1 MusicXML

MusicXML je formát založený na XML pro zápis hudebních skladeb v západní notaci. Byl vytvořen pro snadný přenos skladeb mezi jednotlivými aplikacemi pro zápis skladeb. MusicXML soubory obsahují informace o vizuální podobě skladby i informace o použitých nástrojích, které se využívají při konverzi skladeb do zvukových formátů.

#### 2.1.2 DOM, Apache Xerces

W3C DOM, nezkráceně *Document Object Model*, je reprezentace HTML nebo XML souborů jako stromových struktur, kde každý objekt je reprezentován jako uzel. DOM parser načte celý soubor a uloží jej do paměti. Apache Xerces je knihovna pro zpracovávání XML dokumentů a následnou manipulaci s nimi.

## 2.2 MIDI

MIDI, nezkráceně *Musical Instrument Digital Interface*, je specifikace pro propojování různých hudebních zařízení. Specifikuje jednak hardwarové propojení zařízení, jednak komunikační a datový protokol. Komunikace probíhá pomocí tzv. zpráv, které popisují jednotlivé události (např. začátek či konec noty).

### 2.2.1 General MIDI

Specifikace General MIDI rozšiřuje MIDI specifikaci přenosových protokolů o polyfonii, zavádí standardizované zvukové programy (zvuky jednotlivých nástrojů) a přidává některé kontrolní zprávy.

### 2.2.2 SMF

SMF, nezkráceně *Standard MIDI File*, je formát souborů, zapsaných dle General MIDI specifikace, sloužících pro uložení hudebních dat ve formě událostí a nikoliv jednotlivých zvuků, jako je tomu u běžných formátů používaných pro uložení zvuků. Je určen pro přenos dat mezi jednotlivými zařízeními. SMF soubor se skládá z jednotlivých stop, které je možné přehrát současně a které obsahují zprávy, které popisují jednotlivé události. Do SMF souborů se dají ukládat i informace o skladbě či autorovi, pomocí tzv. meta-událostí. Každá událost se skládá z časového přírůstku (od poslední události), stavového bytu, který udává o jakou událost půjde, a datového bytu, který obsahuje parametry události.

## 2.3 GNU Octave

GNU Octave je matematicky orientovaný vyšší programovací jazyk, značně kompatibilní s jazykem Matlab, ale na rozdíl od něj je svobodný. K ukládání dat se nejčastěji používají matice a GNU OCTave podporuje velké množství maticových operací. GNU Octave podporuje i ukládání dat do tzv. datových struktur, pomocí kterých se dají reprezentovat stromy dat v paměti.

### 3 Teoretická část

SMF soubor (.mid) obsahuje nejprve hlavičku souboru, ve které je specifikován typ souboru, počet stop a rychlost přehrávání. Z MusicXML dokumenty tedy potřebuji získat informaci o celkovém počtu hlasů a o rychlosti. Celkový počet hlasů nikde zapsán není, je tedy třeba dokument projít a počet hlasů spočítat - do jedné stopy sice lze zapsat, aby hrálo více tónů najednou, ale je snazší, když každý hlas bude zapsán v jiné stopě. Údaj o rychlosti se v MusicXML dokumentu nachází ve stejném formátu jako v SMF souboru (počet tiků na jednu čtvrtkovou notu), není tedy třeba ho nijak upravovat.

Po hlavičce následuje zápis jednotlivých stop, kde každá stopa odpovídá jednomu hlasu. Protože různé hlasy jsou hrány různými nástroji, zařadím na začátek každé stopy událost *Program Change*. V MusicXML souboru se nachází informace o jednotlivých nástrojích - jejich id, používané v zápisu jednotlivých not i jejich MIDI program. Tyto informace však nejsou propojeny s čísly jednotlivých hlasů, takže musím sestavit převodní tabulku mezi hlasy a id jednotlivých nástrojů (resp. jejich MIDI programy).

Do první stopy také zařadím událost *Set Tempo*, která udává dobu trvání čtvrtkové noty v mikrosekundách. Hodnota v mikrosekundách s souboru MusicXML zapsána není, je tam však údaj o počtu čtvrtkových not v jedné minutě, který se dá na požadovanou hodnotu přepočítat.

Tímto jsou nastavené veškeré potřebné parametry stopy a na řadu tedy přichází zápis not v jednotlivých stopách. K tomu využiji událostí *Note On* a *Note Off*. Obě tyto události mají dva parametry - kód tónu a rychlost začátku přehrávání noty (resp. konce přehrávání). Tato rychlost v souboru MusicXML uvedena není. Do SMF souboru se noty zapisují pomocí číselného kódu, který je přiřazen každé notě v rozsahu od tónů, které lidské ucho vnímá jen jako hluky až po šestičárkované G. V souboru MusicXML je nota zapsána pomocí názvu tónu a oktávy v mezinárodním zápisu (např. D 5 - dvoučárkované d). Pokud má být tón zvýšený či snižený, vyskytuje se v MusicXML souboru parametr, který udává o kolik půltónů je tón posunutý.

Před každou zprávou události je tzv. deltačas, udávající, kdy má začít daná událost, a to jako počet tiků od poslední události. V MusicXML souboru je zapsaná doba trvání noty či pomlky, takže je jen potřeba tyto hodnotypřiřadit správným událostem. Přehrávání noty může začít hned po skončení předchozí události (není-li v zápisu pomlka), ukončení přehrávání pak po čase daném délkou přehrávané noty. Každá stopa musí být zakončena událostí *End of Track*.

## 4 Implementace

MusicXML soubor si po načtení zkonvertuji do struktury (funkcí `xml2struct`), aby se dalo k jednotlivým objektům snadněji přistupovat. Nejprve si ve struktuře najdu data, která se nemění (např. údaje o tempu nebo programy a id jednotlivých nástrojů). Projdu strukturu, abych zjistila celkový počet hlasů (tedy počet stop) a sestavila převodní tabulku mezi hlasem, id nástroje a MIDI programem nástroje. Ukázalo se, že pokud skladba obsahuje více nástrojových částí, tak v každé části jsou hlasy číslovány od jedničky, takže bude lepší nejprve zjistit počet hlasů a až při procházení celého MusicXML souboru a zapisování dat o jednotlivých notách připravovat převodní tabulku. Po projití jedné části zapíši do proměnné, ve které mám obsah budoucího SMF souboru, všechny hlasy (stopy), které se v dané části vyskytly. Zápis do SMF souboru provedu najednou, po projití celého MusicXML dokumentu. V MusicXML dokumentu je hudební skladba zapsaná po nástrojových sekcích (částech) a dále po taktech a po jednotlivých notách v taktech. Všechny hlasy vyskytující se v dané části jsou tedy zapsány „přes sebe“. Při procházení tedy musím informace o jednotlivých notách rozdělovat podle hlasu do jednotlivých polí, aby při zápisu jedné stopy stačilo projít pole jednoho hlasu a bez složitého hledání dalšího výskytu hlasu a přeskakování v indexování ji přepsat do MIDI událostí.

## 5 Testování

V průběhu testování se ukázalo, že program sice vytvoří soubor, který je rozpoznán jako MIDI, ale nedá se přehrát. Párování hlasu a nástroje, který daný hlas hraje, či spíše zpětné dohledávání se kvůli neplatnému indexování ukázalo být těžším, než se zprvu zdálo, a proto je v souboru `xml2midi.m` napevno nastaveno, že všechny hlasy hraje klavír. Tento program byl odlaďován na přiloženém souboru `Telemann.musicxml`, u všech ostatních souborech, na kterých jsem program testovala se nepovedlo soubor nahrát – zdá se, že pro ně špatně sebehla funkce `removeDOCTYPE`, protože se objevovala chybová hláška, jejímuž objevování se měla tato funkce zamezit.

## 6 Návod k použití

Pro použití je třeba mít nainstalovaný program GNU Octave[12]. Pro správné fungování je třeba nahrát knihovny pro parsování XML dokumentů. V příkazovém okně GNU Octave je třeba spustit následující příkazy:

```
>>javaaddpath("cesta_k_souboru_xml-apis.jar");  
>>javaaddpath("cesta_k_souboru_xercesImpl.jar");  
>>pkg load io;
```

Zároveň by soubor xml2struct.m, soubor remove\_DOCTYPE.m a zdrojový \*.musicxml soubor měly být ve stejné složce, jako soubor xml2midi.m. Pro spuštění samotného programu je třeba spustit tento příkaz:

```
>>xml2midi("absolutní_cesta_ke_zdrojovému_souboru_bez_přípony");
```

Výsledný soubor \*.mid bude uložen do složky, ve které se nachází zdrojový soubor.

Ve složce xmlesamples je soubor Telemann.musicxml, pro který program sebehne bez chyb. Tento soubor je vzorový soubor MusicXML z oficiálních stránek[3].

## 7 Závěr

Vytvořený program vytvoří soubor \*.mid, ten však při spuštění nic nepřehraje, takže nelze ověřit, jestli všechny funkce, které jsem se snažila implementovat fungují, nebo ne, popřípadě které části fungují. Program samotný sebehne bez chybových hlášek a průběžná data vypadají dle očekávání, v důsledku čehož se mi zatím nepodařilo najít problém, kvůli němuž soubor nic nepřehrává. Vzhledem k šířce MusicXML specifikace a rozmanitosti jednotlivých MusicXML souborů je pravděpodobné, že ne všechny budou tímto programem zpracovatelné. V souboru xml2midw.m je kód doplněn o implementaci změny nástroje, který hraje daný hlas, nicméně tento program bez chyb nesebehne.

## Literatura

- [1] [en.wikipedia.org/wiki/XML](http://en.wikipedia.org/wiki/XML)
- [2] [cs.wikipedia.org/wiki/Extensible\\_Markup\\_Language](http://cs.wikipedia.org/wiki/Extensible_Markup_Language)
- [3] [musicxml.com](http://musicxml.com)
- [4] [usermanuals.musicxml.com/MusicXML/Content/Contents.htm](http://usermanuals.musicxml.com/MusicXML/Content/Contents.htm)
- [5] [w3.org/DOM/](http://w3.org/DOM/)
- [6] [xerces.apache.org/index.html](http://xerces.apache.org/index.html)
- [7] [digitalsoundandmusic.com/chapters/ch6/](http://digitalsoundandmusic.com/chapters/ch6/)
- [8] [root.cz/clanky/rozhrani-midi-na-osobnich-pocitacich/](http://root.cz/clanky/rozhrani-midi-na-osobnich-pocitacich/)
- [9] [root.cz/clanky/general-midi-a-format-souboru-smf/#k03](http://root.cz/clanky/general-midi-a-format-souboru-smf/#k03)
- [10] [cs.wikipedia.org/wiki/SMF](http://cs.wikipedia.org/wiki/SMF)
- [11] [csie.ntu.edu.tw/~r92092/ref/midi/](http://csie.ntu.edu.tw/~r92092/ref/midi/)
- [12] [gnu.org/software/octave/](http://gnu.org/software/octave/)
- [13] [wiki.octave.org/GNU\\_Octave\\_Wiki](http://wiki.octave.org/GNU_Octave_Wiki)
- [14] [octave.org/doc/](http://octave.org/doc/)
- [15] [mathworks.com/matlabcentral/fileexchange/58700-xml2struct-with-bug-fix-and-added-features](http://mathworks.com/matlabcentral/fileexchange/58700-xml2struct-with-bug-fix-and-added-features)