

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## PŘEDNÁŠKY NA CESTY - KONVERZE ZÁZNAMU PŘEDNÁŠKY DO E-BOOK FORMÁTU

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN ADAMEC

BRNO 2015



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## **PŘEDNÁŠKY NA CESTY - KONVERZE ZÁZNAMU PŘEDNÁŠKY DO E-BOOK FORMÁTU**

LECTURES WITH YOU - LECTURE RECORDINGS CONVERSION INTO E-BOOK FORMAT

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MARTIN ADAMEC**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. SZÓKE IGOR, Ph.D.**

BRNO 2015

## **Abstrakt**

Tato práce se zabývá převodem záznamu přednášky, uloženém v XML souboru, do formátu EPUB verze 3.0. Výsledný soubor obsahuje přednášené snímky, které jsou doplněny zvukovým výkladem přednášejícího. Řešení je implementováno v jazyce Python.

## **Abstract**

This thesis deals with the transfer record lectures, stored in XML files, to EPUB format (version 3.0). The resulting file includes the reciting images with audio commentary of lecturer. The solution is implemented in Python.

## **Klíčová slova**

ePub, Python, e-kniha

## **Keywords**

ePub, Python, e-book

## **Citace**

Martin Adamec: Přednášky na cesty - konverze záznamu přednášky do e-book formátu, bakalářská práce, Brno, FIT VUT v Brně, 2015

# Přednášky na cesty - konverze záznamu přednášky do e-book formátu

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Igora Szóke, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Martin Adamec  
11. července 2015

## Poděkování

Chtěl bych poděkovat vedoucímu své bakalářské práce Ing. Igoru Szóke, Ph.D. za pomoc a cenné rady při konzultacích.

© Martin Adamec, 2015.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Popis problému</b>	<b>4</b>
2.1	Existující řešení . . . . .	5
2.2	Aplikace pro čtení formátu ePub . . . . .	6
2.2.1	Přehled aplikací . . . . .	7
<b>3</b>	<b>Formát ePub</b>	<b>8</b>
3.1	Vlastnosti ePub . . . . .	8
3.2	Struktura ePub . . . . .	9
3.3	ePub a práce se zvukem . . . . .	9
<b>4</b>	<b>Návrh řešení</b>	<b>10</b>
4.1	Použité technologie . . . . .	10
4.1.1	Programovací jazyk Python3 . . . . .	10
4.1.2	Šablonový systém jinja2 . . . . .	11
4.1.3	Git . . . . .	11
4.1.4	Nástroje pro úpravu zvukové stopy . . . . .	11
4.2	Návrh aplikace . . . . .	12
4.2.1	Struktura aplikace . . . . .	12
<b>5</b>	<b>Implementace</b>	<b>14</b>
5.1	Zpracování vstupního souboru . . . . .	14
5.1.1	Struktura zpracovávaného XML souboru . . . . .	14
5.2	Práce se zvukovou stopou . . . . .	15
5.3	Naplnění struktury ePub . . . . .	16
5.3.1	Třída EpubBook . . . . .	16
5.3.2	ePub a možnost překrývání objektů . . . . .	17
5.4	Validace výsledné aplikace . . . . .	18
5.5	Spuštění aplikace . . . . .	18
<b>6</b>	<b>Testování</b>	<b>20</b>
6.1	Testování úprav zvukové stopy . . . . .	20
6.2	Testování optimální rychlosti . . . . .	21
<b>7</b>	<b>Závěr</b>	<b>22</b>
7.1	Zhodnocení výsledku práce . . . . .	22
7.2	Návrh na následný posun aplikace . . . . .	23

<b>A</b>	<b>Obsah CD</b>	<b>25</b>
<b>B</b>	<b>Sekvenční diagram aplikace</b>	<b>26</b>
<b>C</b>	<b>Optimální schéma XML souboru</b>	<b>27</b>
<b>D</b>	<b>Struktura ePub</b>	<b>28</b>
<b>E</b>	<b>Instalace</b>	<b>29</b>

# Kapitola 1

## Úvod

V současné době stále více využíváme mobilní zařízení, ať už jsou to mobilní telefony, tablety nebo také čtečky elektronických knih. Stoupající výkonnost těchto zařízení nám umožňuje rozšiřovat paletu aplikací a služeb, které se stávají naším každodenním společníkem.

Jeden z nových trendů je čtení digitalizovaných knih ve svém přenosném zařízení. Pro tento účel byly vyvinuty nejen aplikace určené do chytrých telefonů, ale také vznikla nová zařízení - *čtečky e-knih*, které jsou určeny primárně pro čtení e-knih a většinou nemají jinou funkci. Tyto čtečky obvykle disponují uhlopříčkou čtecí plochy okolo 6 palců (cca 15 cm). Jsou tedy o něco větší než chytré telefony, což přináší lepší komfort při čtení.

Jak aplikace, tak čtečky nám umožňují tvořit virtuální poličky, což nám umožní roztřídit si knihy dle svých kritérií. V e-knihách si můžeme udělat záložky či poznámky. Většina aplikací si pamatuje, kde čtenář skončil. Toto jsou základní výhody elektronických knih a také asi důvody, proč se stávají v dnešní době tak populární.

Ve vytvářené aplikaci se budu zabírat formátem *ePub*<sup>1</sup>, který je jedním z nejčastěji používaných formátů pro elektronické knihy. Tento formát nám díky verzi 3, přináší rozšířené možnosti pro vytváření e-knih. Jeden z největších přínosů je podpora audia, neboli propojení elementu v textu s audio stopou, která se přehraje v momentě, kdy si zobrazíme (popřípadě po kliknutí) daný element. Tímto lze dosáhnout i předčítání textu, kde právě předčítaný text je zvýrazněn.

Nevýhodou aplikací pro čtení e-knih je neurčení pevného standardu, jak prezentovat data. Jak si později ukážeme, má to za následek odlišnou interpretaci dat (viz kapitola 3) v jednotlivých aplikacích.

Cílem této práce bylo tedy také naleznout ideální řešení napříč všemi platformami a aplikacemi, aby byla výsledná e-kniha správně interpretována v co největší množině aplikací.

Tento formát nám tedy umožňuje rozšířit možnosti zpětného prohlížení přednášek, které FIT VUT v Brně poskytuje studentům. Námi vytvářená aplikace konvertuje data (audio záznam přednášky, přednášené snímky) do formátu pro elektronické knihy, konkrétně ePub. Postupu jak jsem této konverze dosáhl a jaké problémy bylo třeba řešit, se Vám budu snažit přiblížit v následujících kapitolách.

---

<sup>1</sup><http://idpf.org/epub/301>

## Kapitola 2

# Popis problému

Aktuálním trendem se stává vytváření záznamů přednášek a jejich následné poskytnutí omezené skupině lidí (studentům, zaměstnancům, účastníkům kurzu, atd.) či celé veřejnosti. Jeden ze serverů, který nabízí přednášky ke zhlédnutí a stažení v nejrůznějších podobách, je webový portál *SuperLectures.com*. Na tomto portálu můžeme najít veřejné přednášky a také některé přednášky z FIT VUT Brno.

Cílem práce je demonstrovat převod přednášek do formátu ePub pro nově vznikající portál *prednasky.com*, který vyvíjí FIT VUT Brno. Pro tuto práci mi byly poskytnuty data přednášky uložené v XML<sup>1</sup> souboru (struktura viz 5.1.1), který má přesně danou strukturu. Z tohoto souboru lze získat audio-stopu přednášky, promítané snímky a další data, která se použili při vytváření výsledné e-knihy, pomocí této aplikace.

I přes rozmanitost poskytnutých formátů, ve kterých je možné si přednášku stáhnout, není poskytnut formát, který by byl vhodný pro přenosná zařízení, zejména s ohledem na velikost dat. Je možné získat video-záznam, který má však velikost v řádu stovek megabajtů a díky tomu není vhodný pro přenosná zařízení, která nedisponují tak velkým úložným prostorem jako osobní počítače. Jako zajímavá alternativa se naskytlo využití formátu elektronické knihy, konkrétně se jedná o formát **ePub**. Tento formát je ideálním pro spojení přednášených snímků a audio záznamu dané přednášky. Jedná se o souborový formát ZIP s přesně danou strukturou pro *ePub*. Výsledný soubor se bude skládat pouze z komprimovaných obrázků a audio souboru, tím se docílí menší velikosti výsledného souboru, než je u videozáznamu.

Zmíněný formát od své verze 3.0, která byla uveřejněna v roce 2011, umožňuje synchronizaci zvoleného elementu (obrázek, text, ...) s audio-stopou. Zvolená audio-stopa, se v momentě zobrazení elementu spustí od nastaveného času. Díky této funkcionalitě lze například vytvářet e-knihy, které čtenáři předčítají zobrazený text a tento text je i současně zvýrazněn. Více o formátu ePub lze najít v kapitole 3.

Interpretace ePub formátu je často odlišný dle konkrétní aplikace, což ze subjektivního pohledu vnímám jako největší slabinu tohoto formátu. Ne všechny aplikace jsou však schopny přehrát i audio stopu, ale o tom bude více napsáno v podkapitole 2.2.1, která se věnuje formátu ePub podrobněji.

---

<sup>1</sup>eXtensible Markup Language



## 2.1 Existující řešení

Volně dostupnou aplikaci, ať už konzolovou či klasickou, která by řešila stejnou problematiku, tedy vytvoření e-knihy (formátu ePub) se zvukovou stopou, jsem nenašel. Dle mého názoru je hlavní příčina to, že pokud nějaká firma či instituce potřebuje takovouto konverzi má přesně danou vstupní strukturu, stejně jako v této práci, a výsledná e-kniha má specifický účel. Jako příklad můžeme uvést firmu **Sinkronigo**<sup>2</sup>, která se zabývá vytvářením e-knih, kde zobrazený text je předčítán, však jejich aplikace není volně dostupná. Existuje více firem, které nabízí konverzi dat do e-knihy, ale vždy se jedná o soukromý nástroj, který neposkytují veřejně.

Pokud by si běžný uživatel chtěl vytvořit e-knihu, tak si většinou vystačí s převodem souboru PDF či doc, který obsahuje pouze text a obrázky. Nástrojů pro tuto základní konverzi existuje nesčetně mnoho a některé jsou i online, jako je například konvertor na webu Online-Convert.com<sup>3</sup>. Díky univerzálnosti těchto nástrojů nebývá výsledek příliš kvalitní a je vhodný pouze pro osobní použití.

Také pro LibreOffice Writer<sup>4</sup> je dostupný plugin Writer2ePub<sup>5</sup>, který umožňuje text dokumentu uložit ve formátu ePub. S použitím tohoto pluginu můžeme dosáhnout o něco lepšího výstupu než z online řešení.

Pokud uživatel vyžaduje pokročilejší práci s formátem ePub, může využít editory pro tento formát, které umožňují měnit meta data, editovat obal a další drobné zásahy do obsahu. Jeden z takových nástrojů je aplikace **Calibre**, o které se zmíním ještě detailněji, a nebo aplikace **Sigil**<sup>6</sup>, která má obdobnou funkcionalitu.

### Nástroj Calibre

Calibre<sup>7</sup> je volně dostupná aplikace, která nabízí celou škálu možností, jak si upravit e-knihu. Předností aplikace je její multiplatformnost a volně dostupné zdrojové soubory na GitHub<sup>8</sup>. Jádro aplikace je napsáno v jazyce Python a stalo se mým studijním materiálem při tvoření aplikace.

Díky této aplikaci si lze vytvořit domácí knihovnu e-knih, kterou si můžeme hromadně vyexportovat pro své mobilní zařízení. Dokáže zpracovat celou řadu formátů (PDF, doc, mobi, lit, ...) a převádět mezi těmito formáty. Aplikace nabízí i vyhledávání knih z několika zdrojů a jejich stažení.

Tento nástroj aktuálně nemá nejspíše konkurenci především díky tomu, že si v této aplikaci můžete otevřít e-knihy a upravovat text, měnit obrázky a spoustu dalšího. Předěláním již existující e-knihy si lze vytvořit svou vlastní. Bohužel nenabízí možnost vytvořit si vlastní e-knihu tak, že by uživatel dostal základní šablonu, kde by si již sám přidával data, jak potřebuje.

### Shrnutí

Obecná aplikace, která slouží pro vytváření (nikoliv editaci) e-knihy pravděpodobně neexistuje. A také není výrazný zájem, aby takový nástroj byl volně publikovaný, jelikož běžný

---

<sup>2</sup><http://www.sinkronigo.com/>

<sup>3</sup><http://ebook.online-convert.com/convert-to-epub>

<sup>4</sup><http://www.libreoffice.org/discover/writer/>

<sup>5</sup><http://extensions.services.openoffice.org/en/project/Writer2ePub>

<sup>6</sup><http://sigil-ebook.com/>

<sup>7</sup><http://calibre-ebook.com/>

<sup>8</sup><https://github.com/kovidgoyal/calibre>

uživatel se pravděpodobně nikdy nesetká s potřebou udělat si složitější vlastní e-knihu a vystačí si s volně dostupnými konvertory.

## 2.2 Aplikace pro čtení formátu ePub

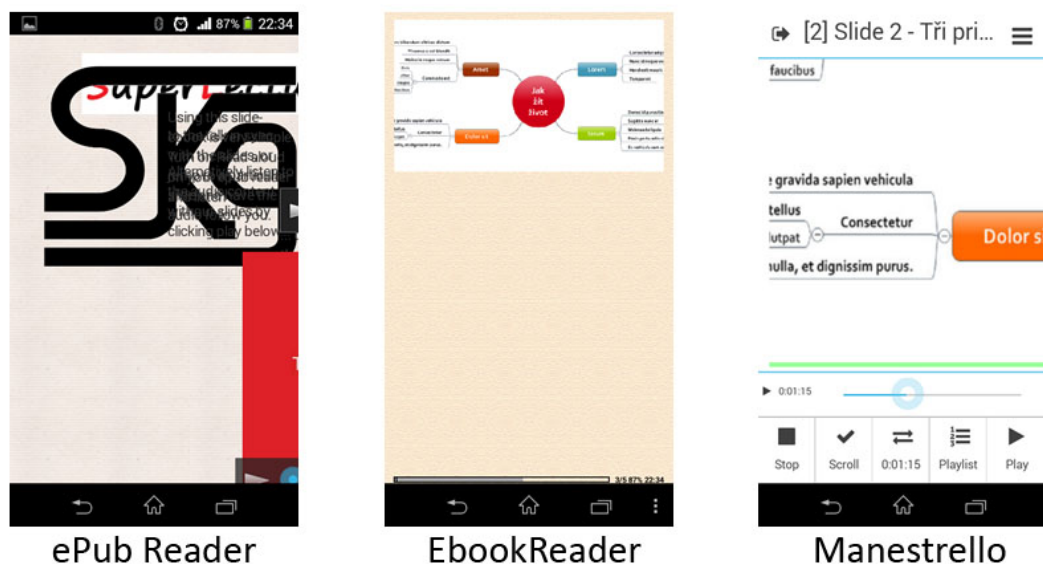
Téměř všechny moderní platformy (Android, iOS, Windows, různé Linux distribuce, Kindle, ...) dokáží pracovat s formátem ePub. Některé to zvládají již v základní instalaci, jiné za pomoci aplikace tvůrce operačního systému (OS) nebo třetí strany. Vytvářený program se zaměřuje především na operační systém Android, ale snaha je i o co největší podporu aplikací pod operačním systémem iOS nebo na zařízeních Kindle.

Aplikací je nepřehledné množství a vybrání té optimální není snadné. Hlavní problém se ukrývá v tom, že není pevně daný standard, který by určil, jak mají aplikace interpretovat ePub formát, takže můžeme najít mírné nuance, ale i velmi odlišnou interpretaci. V tomto se skýtal největší problém při vytváření aplikace - bylo třeba najít optimální XHTML strukturu jednotlivé stránky a CSS vlastnosti, aby se e-kniha chovala obdobně v co největším počtu aplikací.

Je nutno také rozlišovat aplikace, které jsou zastaralé a podporují pouze starší strukturu formátu ePub (verze 2).

### Odlíšné interpretace aplikací

Příklad odlišné interpretace e-knihy<sup>9</sup>, která mi byla zadána jako referenční, můžete najít na následujícím obrázku (2.1). E-kniha obsahuje také zvukovou stopu, tedy je to výstup přesně takový, jaký je požadován z mé aplikace. Všechny tři aplikace jsou určené pro operační systém Android.



Obrázek 2.1: Odlíšná interpretace e-knihy

Lze si všimnout, že vzorová e-kniha není vytvořena správně, jelikož její interpretace je velmi odlišná. A zároveň si lze povšimnout, že ne všechny aplikace podporují audio,

<sup>9</sup>Bar Camp Brno 2014 - Tři principy veřejného mluvení - záložka "Stáhnout/Sdílet" položka EPUB

a to i přesto, že v popisu aplikace je uvedeno, že to podporují.

Do srovnání jsem chtěl zahrnout i aplikaci přímo od tvůrce OS Android - Google, ale tato aplikace vyhodnotila tuto e-knihu jako nevalidní.

### 2.2.1 Přehled aplikací

Níže uvedená tabulka ukazuje příklady aplikací a jejich podporu funkcí. Uvedené aplikace jsou ukázkové pro porovnání jejich funkcionality, aplikací existuje daleko více, ale tím by vznikly redundantní hodnoty v tabulce.

Název aplikace	OS	Audio podpora	Audio funguje
Knihy Google <sup>a</sup>	Android	Ano	Ne*
Manestrello <sup>b</sup>	Android	Ano	Ano
ePub Reader <sup>c</sup>	Android	Ne	-
Gitden Reader <sup>d</sup>	Android, iOS	Ano	Ano
Calibre <sup>e</sup>	Win, Linux, OS X	Ne	-
iBooks <sup>f</sup>	iOS, OS X	Ano	Ano

Tabulka 2.1: Srovnání aplikací pro čtení ePub

\* Knihy Google deklarují podporu audia, nepodařilo se mi však najít způsob, jak jej v této aplikaci spustit. Přisuzuji to tomu, že audio se spouští kliknutím na element, který je spojen s audio souborem, ale v této aplikaci se po kliknutí zobrazí obsah knihy.

---

<sup>a</sup>Knihy Google na Google Play

<sup>b</sup><http://www.readbeyond.it/menestrello/>

<sup>c</sup><http://www.graphilos.com/epub/Default.aspx>

<sup>d</sup><http://gitden.com/>

<sup>e</sup><http://calibre-ebook.com/>

<sup>f</sup><https://www.apple.com/ibooks/>

## Kapitola 3

# Formát ePub

Formální definice formátu ePub [3]:

EPUB (electronic publication) je svobodný software vytvořený pro e-booky dle standardu organizace International Digital Publishing Forum (IDPF). Přípona souboru je .epub, ve skutečnosti se jedná o přejmenovaný soubor .zip se specifickou adresářovou a souborovou strukturou. Zip byl zvolen jako kontejner pro svou širokou podporu a známou specifikaci. ePub je navržen k distribuci a výměně jako digitální publikace a dokumenty. Definuje způsob prezentace obálek, strukturovaný a schématicky rozšířený webový obsah - včetně HTML5, CSS, SVG, obrázky a další zdroje pro distribuci ve formátu tzv. "single-souboru". Obsah je zapsán pomocí XML a HTML5 tagů. Jako formát pro elektronickou publikaci splňuje tzv. reflowable, to znamená, že čtenář si může přizpůsobit rozvržení textu pro konkrétní čtecí zařízení. ePub se stal neoficiálním českým formátem pro elektronickou knihu.

### 3.1 Vlastnosti ePub

V naší aplikaci pracuji s ePub verze 3, která byla uveřejněna v roce 2011 a přináší podporu HTML5 značek, CSS3 vlastností a také podporu javascriptové knihovny jQuery. Od této verze se začaly e-knihy tohoto formátu daleko více rozšiřovat jako univerzální formát napříč většinou známých platforem.

Díky podpoře formátu HTML5 lze mít v e-knize audio či video soubory, které lze synchronizovat s textem, obrázkem atd., což je stěžejní pro mou aplikaci. Pokud je audio synchronizováno s textem, tak většina aplikací označuje právě čtený text.

Další významná novinka v této verzi je, že při načítání externích multimediálních souborů lze použít tzv. **fallback**. Tato funkce umožňuje odkázat na jiný multimediální soubor v případě, že daný interpret (čtečka/aplikace) nepodporuje onen typ souboru. To znamená, že pokud máme hudební soubor ve formátu M4A a MP3 a daná aplikace by formát M4A neuměla přehrát, tak nastavíme **fallback**, který načte soubor ve formátu MP3 (samozřejmě za předpokladu, že tento formát je schopna zpracovat).

K této problematice neexistuje žádná oficiální literatura, a tak je třeba si vystačit s oficiální dokumentací, kterou můžeme najít na webové stránce IDPF.org<sup>1</sup> a s neoficiálními poznatky na internetových fórech.

---

<sup>1</sup><http://idpf.org/epub/30>

## 3.2 Struktura ePub

Jak již bylo řečeno ve formální definici, ePub soubor je pouze ZIP soubor s pevně danou strukturou - tuto základní strukturu a také kompletní strukturu, která je použita v naší aplikaci lze najít v příloze D. Nyní si vysvětlíme význam jednotlivých souborů:

- `mimetype` - soubor, který informuje o tom, že se jedná o ePub knihu, obsahuje pouze příslušný MIME typ (`application/epub+zip`)
- `container.xml` - je to jediný soubor ve složce `META-INF`, který obsahuje pouze informaci, kde najdeme hlavní soubor, který obsahuje meta data a další informace o souboru. V našem případě je to `package.opf`
- `toc.xhtml` - obsah knihy
- `package.opf` - xHTML soubor, který obsahuje veškeré informace o e-knize: její ID, datum vytvoření, jazyk publikace, autory, definice všech stránek, obrázků, audio stop a jiných souborů.
- `text` - složka, která obsahuje jednotlivé stránky e-knihy
- `image` - složka, která obsahuje obrázky použité v e-knize
- `audio` - složka, která obsahuje audio soubor
- `stylesheet.css` - soubor s CSS vlastnostmi pro stránky e-knihy

## 3.3 ePub a práce se zvukem

Již jsem uvedl, že ePub verze 3 podporuje HTML5 značku `audio`<sup>2</sup>. Většina aplikací, které umí přehrávat audio, podporuje tyto základní formáty: MP3, WAV, M4A, MP4. Některé aplikace mají rozšířenou škálu podporovaných audio formátů, takže lze využít možnosti vložit do značky `audio` více zdrojů, které se procházejí shora dolů a aplikace zvolí první zdroj, který je schopna přehrát. Pokud aplikace neumí pracovat ani s jedním ze zdrojů, zobrazí informaci, která se uvádí před uzavřením značky `audio` - viz následující příklad použití:

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>
```

---

<sup>2</sup>Více o značce `audio` lze najít zde: [http://www.w3schools.com/html/html5\\_audio.asp](http://www.w3schools.com/html/html5_audio.asp)

# Kapitola 4

## Návrh řešení

V následující kapitole bude čtenář seznámen s návrhem vytvářené aplikace a s technologiemi, které byly pro tvorbu aplikace zvoleny.

### 4.1 Použité technologie

Před započítím realizace bylo třeba si stanovit nástroje, které budou využity pro tvorbu naší aplikace. Technologie byly vybrány po analýze problematiky s ohledem na co nejefektivnější vývoj a snadné použití programu. V případě, kdy se skýtalo více možností, byla technologie zvolena dle subjektivního uvážení s ohledem na dosavadní zkušenosti s daným nástrojem.

#### 4.1.1 Programovací jazyk Python3

Jako první si bylo třeba zvolit programovací jazyk. S přihlédnutím k faktu, že výsledný program je konzolový, jako nejlepší varianta se naskytl jazyk Python3, se kterým jsem měl již zkušenosti. Tento jazyk je také v základu každé linuxové distribuce. Učební literaturou se stala kniha *Python3 Výukový kurz* [2] a oficiální dokumentace<sup>1</sup>.

Tento jazyk byl také zvolen z důvodu, že již v základu obsahuje velké množství knihoven, které lze snadno rozšiřovat knihovnami třetích stran. Pro naši aplikaci byly využity následující knihovny třetích stran:

- `jinja2` - knihovna pro šablonování, více v podsekcí [4.1.2](#)
- `termcolor`<sup>2</sup> - pomocná knihovna pro vytisknutí výstupu na standardní výstup s použitím barev
- `zipfile`<sup>3</sup> - vytvoření archivu ZIP, respektive ePub
- `shutil`<sup>4</sup> - funkce pro práci se soubory (rekurzivní mazání, kopírování, přesouvání, ...)

---

<sup>1</sup><https://docs.python.org/3/>

<sup>2</sup><https://pypi.python.org/pypi/termcolor>

<sup>3</sup><https://docs.python.org/3/library/zipfile.html>

<sup>4</sup><https://docs.python.org/3/library/shutil.html>

### 4.1.2 Šablonový systém jinja2

V rámci tvorby výstupní struktury ePub souboru, bylo třeba dynamicky vytvářet soubory (xHTML soubory, `package.opf`, `mimetype`, atd. viz popis struktury ePub v sekci 3.2). Jako řešení se nabízely následující možnosti:

- (a) Vytvářet soubory za pomoci standardních knihoven, jako je například knihovna `xml.etree.ElementTree`<sup>5</sup>
- (b) Zvolit vhodný systém pro šablonování

S přihlédnutím na přehlednost a snadnou úpravu jsem zvolil šablonový systém, konkrétně se jedná o knihovnu `jinja2`<sup>6</sup>. Tato knihovna byla vybrána na základě referencí a mého subjektivního rozhodnutí. `jinja2` je moderní knihovna, která vychází z Django<sup>7</sup> syntaxe.

### 4.1.3 Git

Efektivní týmový vývoj aplikací vyžaduje použití verzovacího systému, který slouží k uchování historie prováděných změn ve zdrojovém kódu a jejich šíření mezi ostatní členy týmu. Avšak verzovat svou práci je vhodné i v případě, kdy projekt vyvíjí pouze jeden vývojář. Distribuovaný systém správy verzí Git klade důraz především na jednoduchost a flexibilitu.

Důvodem využití tohoto nástroje bylo především kvůli revizi změn a vytváření záložní kopie celého projektu na vzdáleném serveru, ze kterého mohou být data kdykoliv obnovena. Jelikož jsem aplikaci vyvíjel na více zařízeních, usnadnil mi Git přenos změn stavu projektu.

Volně dostupných verzovacích systémů je více, nejznámějšími jsou `GitHub`<sup>8</sup> a `BitBucket`<sup>9</sup>. Zvolil jsem `BitBucket`, jelikož s tímto nástrojem mám dlouholetou zkušenost.

### 4.1.4 Nástroje pro úpravu zvukové stopy

Dle zadání této práce bylo třeba upravit audio stopu, neboli záznam hlasu přednášejícího. Tento soubor bylo zapotřebí optimalizovat pro použití v e-knize. Zejména se jednalo o vymazání tzv. *hluchých* míst, tedy dlouhé tiché prostoje. Také byla přidána možnost změny rychlosti záznamu.

### Detekce hluchých míst za pomoci rozpoznání fonémů

Pro detekci *hluchých* míst mi byla doporučena knihovna `PhnRec` [1], která byla vytvořena kolektivem autorů z fakulty informačních technologií na VUT v Brně. Výstup aplikace je seznam fonémů s časovým razítkem.

Knihovna vyžaduje specifický vstup: typ souboru formátu WAV, frekvenci 8000 Hz a přenosovou rychlost 16bit/s. Do této podoby převedeme MP3 soubor, který nám byl dán jako vstupní pro naši aplikaci, za pomoci konzolového audio editoru `SoX`.

<sup>5</sup><https://docs.python.org/3/library/xml.etree.elementtree.html>

<sup>6</sup><http://jinja.pocoo.org/>

<sup>7</sup><https://www.djangoproject.com/>

<sup>8</sup><https://github.com/>

<sup>9</sup><https://bitbucket.org/>

## Audio editor SoX

SoX<sup>10</sup> je volně šiřitelný editor audio souborů, napříč všemi platformami. Editor je psán v programovacím jazyku C a jedná se o konzolovou aplikaci.

V našem projektu je využit pro vyříznutí *hluchých* míst, převodu mezi formáty a změnu rychlosti záznamu zvuku.

## Knihovna ffmpeg

Výstupní MP3 soubor je třeba převést do formátu M4A, což nám provede linuxová knihovna `ffmpeg`<sup>11</sup>. Formát MP3 pro e-knihu formátu ePub není vhodný, jelikož při nižší přenosové rychlosti se audio nesprávně synchronizuje a to i přesto že audio-stopa má konstantní bit rate a nikoliv variabilní.

## 4.2 Návrh aplikace

Po stanovení technologií, které se budou využívat, bylo třeba stanovit koncept vytvářené aplikace. Plánování vývoje aplikace je nezbytná součást každého projektu, byť bývá často opomíjena méně zkušenými vývojáři. Dobře připravený návrh nám přináší časovou úsporu a tím i redukuje potřebné prostředky pro vývoj.

Grafické uživatelské rozhraní aplikace nebylo třeba vytvářet, jelikož aplikace se spouští z terminálu. Avšak uživatel by měl vědět, co se právě děje, a proto aplikace vypisuje na standardní výstup informace o aktuálním stavu konverze.

### Cílová skupina

Stěžejním bodem návrhu aplikace je fakt, že program bude využívat omezená skupina lidí. Jelikož se jedná převážně o pracovníky FIT VUT v Brně, lze předpokládat, že tato skupina bude mít pokročilé znalosti v užívání konzolových aplikací.

#### 4.2.1 Struktura aplikace

Aplikace je psána jako plugin (balíček) jazyka Python3, díky čemuž ji bude možno začlenit do libovolného projektu, kde bude přes stanovené rozhraní pracovat stále stejně. Zdrojové soubory bude třeba rozdělit do logických celků, především třída pro vytváření ePub souboru by měla být co nejméně závislá na zbytku projektu, aby se mohla potencionálně vložit i do jiných projektů. Veškeré potřebné zdrojové soubory jsou umístěny ve složce `XmlToEpub`. Jelikož bylo třeba využít některých knihoven/balíčků třetích stran, tak v kořenovém adresáři se vedle hlavního adresáře vytvářené aplikace nachází i adresář `Lib`, který obsahuje dané knihovny (viz podkapitola 4.1.1). Kompletní adresářovou strukturu lze najít na obrázku 4.1.

Nad aplikací je vytvořeno základní rozhraní, které je určeno následujícími třídami:

- **Base** - základní třída, která zpracovává argumenty a řídí celou konverzi, má také na starosti úklid po dokončení či přerušení programu
- **Core** - rodičovská třída většiny tříd, která byla vytvořena pro zpracování inicializačních argumentů tříd

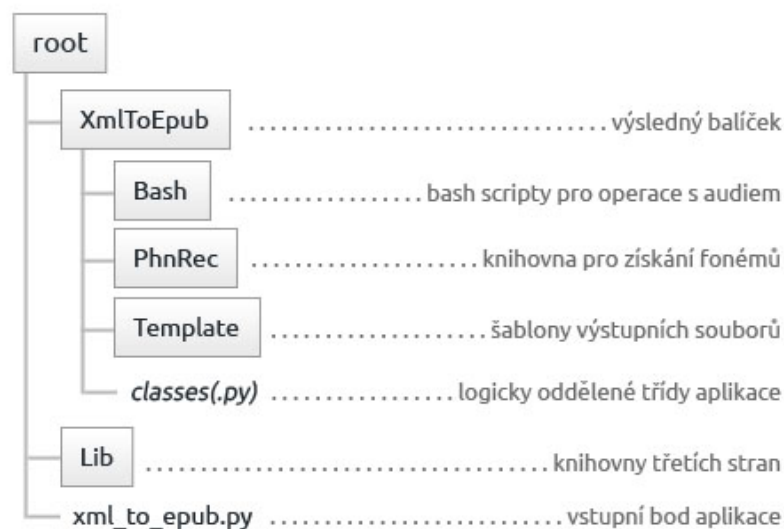
---

<sup>10</sup><http://sox.sourceforge.net/>

<sup>11</sup><https://www.ffmpeg.org/>



- `Config` - třída uchovávající konfiguraci programu
- `Function` - knihovna obecných funkcí, jako je stahování souborů, generování nahodných řetězců apod.
- `Exceptions.py` - soubor s deklarací pomocných výjimek pro zpřehlednění běhu aplikace



Obrázek 4.1: Struktura aplikace `XmlToEpub`

### Sekvenční diagram

Pro demonstraci komunikace objektů aplikace jsem vytvořil sekvenční diagram, který lze najít v příloze **B**. Pro přehlednost diagramu není zobrazena komunikace s objekty, které poskytují pomocné funkce, například stažení souborů apod.

# Kapitola 5

## Implementace

Po představení návrhu aplikace je na čase podrobně rozebrat samotnou implementaci. Tato kapitola informuje o způsobu vývoje finální aplikace a upozorňuje na důležité aspekty vývoje. Každá z podkapitol se zabývá několika konkrétními logicky souvisejícími celky.

### 5.1 Zpracování vstupního souboru

Jak jsem uvedl v dřívější kapitole (2), vstupem je url XML souboru, který v sobě nese data přednášky, která má být převedena do formátu ePub. Více informací o struktuře uvádím v podsekcí 5.1.1. Tento soubor je pro zpracování třeba stáhnout a za pomoci standardní Python3 knihovny `ElementTree`<sup>1</sup> získat požadovaná data.

Parsování XML souboru je prováděno ve třídě `XmlTools.XmlParser`, která pro každou přednášku v souboru vytvoří a naplní daty instanci třídy `Lecture`.

#### Opakující se snímek

Při zpracovávání se aplikace musí správně zachovat v případě, že načítaný snímek byl již použit v dřívější části přednášky. Ve vytvářené aplikaci při přidání snímku jsou prohledány snímky, které byly přidány dříve. V případě shody URL snímků, se daný snímek znovu nestahuje, ale uloží si nalezený snímek jako svého rodiče.

#### 5.1.1 Struktura zpracovávaného XML souboru

Jak již bylo zmíněno, program získává informace pro svůj chod z XML souboru. Strukturu tohoto souboru je možné nalézt v příloze C. Tato struktura byla určena dle referenčního vstupního souboru k této práci, kterou byla *2. přednáška z předmětu Zpracování řečových signálů*<sup>2</sup>, která je uveřejněna na portálu SuperLectures.com. Význam jednotlivých XML elementů je následující:

- `data` - kořenový uzel
- `lecture` - kořenový uzel jedné přednášky povinně obsahuje atributy `id`: identifikátor dané přednášky a `name`, který skýtá název přednášky, dále volitelně atribut `url` - URL adresa přednášky na SuperLecture.com, která je uveřejněna na poslední straně e-knihy

---

<sup>1</sup><https://docs.python.org/2/library/xml.etree.elementtree.html>

<sup>2</sup><http://www.superlectures.com/fit-zre/api/getLecture?id=1>

- **authors** - seznam autorů dané přednášky, autoři jsou v pod-elementu **person**, kde tento element má povinný atribut **name**, který obsahuje jméno autora
- **speakers** - seznam přednášejících dané přednášky, přednášející jsou v subelementu **person**, kde tento element má povinný atribut **name**, který obsahuje jméno přednášejícího
- **datetime\_recorded** - datum a čas, kdy tato přednáška proběhla, obsah elementu je ve formátu **YYYY-MM-DD hh:mm:ss**, dle standardu w3c<sup>3</sup>
- **thumbnail\_large** - URL cesta k průvodnímu obrázku (cover)
- **audio** - rodičovský uzel pro zvukové části záznamu přednášky, které je možno stáhnout, tento uzel má povinný parameter **duration**, který udává celkovou délku audio stopy přednášky
- **audio\_mp3** - tag pro audio stopu formátu MP3 s kterou budeme dále pracovat, URL souboru je uveden v atributu **src**
- **slides** - rodičovský uzel pro snímky, které jsou prezentovány, argument **number\_of\_slides** nám udává celkový počet přednášených snímků
- **slide** - tento element reprezentuje jednotlivé snímky, kde atribut **start** nám sděluje, v jaké sekundě audio/video stopy se daný snímek prezentuje, z atributu **src\_large** získáme URL adresu daného snímku pro stažení

V zadaném referenčním vstupu najdeme i několik dalších elementů s informacemi o dané přednášce, ty jsou však pro tuto aplikaci nepotřebné a proto byly vynechány.

### Nutné části XML souboru

Pro každou přednášku, kterou chceme zpracovat je nezbytně nutné, aby obsahovala elementy **audio\_mp3**, **datetime\_recorded** a seznam snímků v elementu **slides**.

## 5.2 Práce se zvukovou stopou

Jedním z hlavních úkolů této práce bylo zpracování a optimalizace zvukového záznamu přednášky pro použití v e-knize. Záznam obsahoval mnoho tzv. *hluchých míst*, které je třeba vyříznout. Veškeré úpravy zvukové stopy jsou prováděny v pomocné třídě **Audio.AudioTools**.

### Princip detekce a odstranění hluchých míst

V první řadě bylo nutné převést zvukový záznam na fonémy, dle těchto fonémů bude program dále zpracovávat audio stopu. Tento převod nám zprostředkovala knihovna **PhnRec**, která je součástí výsledné aplikace. Více o této knihovně je uvedeno v podsekcí **4.1.4**. Jako další krok si bylo třeba stanovit co vlastně budeme považovat za *hluché* místo - jsou to fonémy, které značí ticho nebo šum. Fonémy **pau** reprezentující ticho a šum určují fonémy **int** a **spk**.

Dále byly vytvořeny dvě konstanty v třídě **AudioTools**:

<sup>3</sup><http://www.w3.org/TR/NOTE-datetime-970915>

- `MIN_PAUSE` - hodnota v sekundách, která určuje minimální délku *hluchého* úseku
- `CUT_BORAD` - velikost okraje v sekundách, jedná se o hranici, která bude ponechána před a za vyřiznutou částí, čímž se snažíme eliminovat přehnaný zásah do přednesu

V případě nalezení sekvence fonémů ticha či šumu, která přesahuje délku určenou konstantou `MIN_PAUSE`, přidáme časové razítko začátku a konce výřezu do pole, pro pozdější zpracování.

Samotné vyřiznutí hluchých míst se provádí prostřednictvím knihovny `SoX` (viz 4.1.4). Princip spočívá ve vyřiznutí částí, které následně spojíme dohromady opět pomocí knihovny `SoX`, tím vznikne výsledný zvukový soubor formátu MP3. Jednotlivá část je určena koncem jednoho hluchého místa a začátkem druhého.

## Změna rychlosti

Uživatel aplikace může ovlivnit rychlost zvukové stopy a to za pomoci přepínače `-s`, respektive `--speed` (více viz podkapitola 5.5, kde jsou uvedeny přepínače pro ovládní vytvořené aplikace). Tento vstupní argument přijímá desetinné číslo, kde hodnota 1.0 znamená běžnou rychlost a například hodnota 1.2 udává, že audio bude zrychleno o 20%.

K úpravě rychlosti je opět využita knihovna `SoX` a její argument `tempo`, která v kombinaci s argumentem `gain` změní rychlost audio-stopy, bez nepříjemného zkreslení zvuku, které vznikne použitím argumentu `speed`.

Abychom zamezili nežádoucímu zrychlení/zpomalení audio stopy, kdy by už audio stopa nebyla srozumitelná, stanovil jsem interval akceptovaných hodnot: `<0.5, 3.0>`.

## 5.3 Naplnění struktury ePub

Již jsme si ukázali, jak zpracujeme vstupní soubor a také úpravu zvukové stopy záznamu přednášky. Nyní se dostáváme do poslední části tvorby e-knihy, a to naplnění ePub struktury a zabalení do archivu.

### 5.3.1 Třída `EpubBook`

Jedná se o samostatnou třídu vytvářející ePub strukturu (více o tomto formátu můžeme nalézt v kapitole 3), která není závislá na zbytku projektu<sup>4</sup>. Její samostatnost je cílená, jelikož se tímto stává snadno přenositelná pro použití v jiných projektech, a to pouze s úpravou některých metod. Pro naši konkrétní aplikaci bylo zapotřebí mít specifické metody a chování, proto si nyní popíšeme více rozhraní.

### Vstupní bod

Pro vytvoření instance se využívá klasického způsobu, který náleží jazyku Python. Má tři povinné vstupní parametry, kterými jsou:

- `rootDir` - cesta ke složce, kam si může třída ukládat dočasné soubory
- `name` - název e-knihy
- `date` - datum, kdy byla kniha vytvořena

<sup>4</sup>Vazbu má pouze na šablonový systém, který se ovládá pomocí třídy `Template`

## Naplnění daty

Data se plní přes navržené metody třídy, čímž se pouze nastaví meta informace e-knihy a její budoucí obsah. Až zavoláním metody `finish` způsobíme vytvoření ePub struktury složek, přesunutí/zkopírování obrázků a audio stopy, vytvoření potřebných souborů a samotného obsahu. Metoda `finish` přijímá jeden argument a to `filename`, který reprezentuje název výsledné e-knihy.

## Vytvoření výstupních souborů

Jednotlivé soubory, jako je například `package.opf`, `tox.xhtml` či samotné stránky knihy, se vytvářejí za pomoci šablonové knihovny `jinja2` (viz 4.1.2). Pro sjednocené vytváření souborů byla vytvořena třída `Template`, která poskytuje metody pro snadnou práci s daným šablonovým systémem.

### 5.3.2 ePub a možnost překrývání objektů

Úvodem je zapotřebí říct, že každý snímek přednášky je jako samostatná stránka e-knihy. Tato stránka má název `slide-X.xhtml`, kde `X` reprezentuje číslo snímku, které je určeno pořadím ve vstupním XML souboru.

Pro synchronizaci snímku přednášky se zvukovým záznamem přednášky bylo třeba využít metodu překrývání multimediálních objektů - **EPUB Media Overlays**<sup>5</sup>. Princip spočívá v tom, že pro každou stránku knihy, která má být spojena s audio obsahem, je třeba vytvořit tzv. `smil` element. Tento element je vytvořen vždy jako samostatný soubor, kde název je `slide-X.smil`, kde `X` reprezentuje číslo snímku/stránky, s kterým je tento element spjat. Neúplná ukázka takového souboru je uvedena na níže (5.1):

```
<smil xmlns="http://www.w3.org/ns/SMIL" ...>
  <body>
    <seq epub:textref="slide-{x}.xhtml" epub:type="bodymatter chapter"
    ...>
      <par id="p000000">
        <text src="slide-{x}.xhtml#f000000"/>
        <audio clipBegin="{begin}" clipEnd="{end}" ... />
      </par>
    </seq>
  </body>
</smil>
```

Obrázek 5.1: Neúplný vzor `smil` souboru

Za pomoci tohoto souboru se vytvoří objekt nad stránkou `slidex-{x}.xhtml`, který je spjat s elementem nesoucí atribut `id p000000`. Díky tomuto objektu se při aktivaci objektu s atributem `id p000000` spustí i daný zvukový úsek - viz následující vysvětlení proměnných `smil` souboru:

- `x`: udává ID, které doplňuje název stránky, s kterou je tento `smil` element spjat
- `begin` a `end` určují počáteční a koncový čas části zvukového záznamu, který náleží danému snímku

<sup>5</sup><http://www.idpf.org/epub/30/spec/epub30-mediaoverlays.html>

## Provázání stránky se smil objektem

Nově vytvořený objekt, respektive soubor, je třeba provázat s danou stránkou, a také jeho existenci a vlastnosti uvést do hlavního souboru e-knihy `package.opf`. Jednak je třeba uvést do metadat informaci o délce zvukového úseku<sup>6</sup>, dále přidat tento soubor do manifestu a do záznamu stránky v manifestu uvést nový atribut `media-overlay`, který obsahuje ID daného smil souboru v manifestu.

Poslední krok je úprava samotné stránky e-knihy, kde je třeba přidat obalující element snímku (`figure`) s ID `p000000`, element `figure` je třeba obalit xHTML značkou `section` s atributem `epub:type` s příslušnou hodnotou, viz následující část zdrojového kódu stránky (obrázek 5.2):

```
<section epub:type="bodymatter chapter">
  <figure id="f000000"></figure>
</section>
```

Obrázek 5.2: Provázání stránky se smil objektem

## 5.4 Validace výsledné aplikace

Po vytvoření e-knihy, bylo třeba ověřit, zda je vytvořena správně. K tomuto účelu slouží oficiální online validátor na webu IDPF.org<sup>7</sup>. Tento nástroj má však nevýhodu, že má stanoven maximální velikost souboru na 10MB.

Je možné využít jiné online nástroje, které pracují se stejnou knihovnou, nebo si samotnou knihovnu `epubcheck`<sup>8</sup> nainstalovat<sup>9</sup> do své linuxové distribuce. Instalací získáme pohodlnější validaci a odpadá nutnost soubor nahrávat na server třetí strany.

## 5.5 Spuštění aplikace

Jak jsem již dříve uvedl, aplikace je konzolová a určená pro operační systém Linux. Pro svůj běh vyžaduje přítomnost několika nástrojů, informaci o které nástroje se jedná a popis jejich získání naleznete v příloze E.

### Ovlivnění chování aplikace přepínači

Přepínače jsou standardním prostředkem pro ovlivňování stylu spuštění souboru, také se za pomoci přepínačů nastavují vstupní či výstupní soubory. Náš program dokáže zpracovat několik málo přepínačů, avšak toto množství je pro náš program zcela dostačující. Každý zkrácený přepínač, vyjma přepínače `-a`, má i svou dlouhou alternativu. Akceptované přepínače popíši v následujícím seznamu:

- `-h, --help`: při výskytu tohoto přepínače, je na standardní výstup vytisknuta nápověda k programu
- `-i, --input`: **povinný** argument, který udává URL cestu vstupního XML souboru

<sup>6</sup>Podoba meta tagu: `<meta refines="#slide-x.smil"property="media:duration">HH:MM:SS.MS</meta>`

<sup>7</sup><http://validator.idpf.org/>

<sup>8</sup><https://github.com/IDPF/epubcheck>

<sup>9</sup>Instalace příkazem: `apt-get install epubcheck`

- `-o, --output`: **nepovinný** argument, který udává cestu a název výstupního souboru. Pokud není uveden, je soubor pojmenovaný názvem přednášky.
- `-a`: **nepovinný** přepínač, díky kterému vznikne e-kniha bez audio souboru
- `-s, --speed`: **nepovinný** argument určující rychlost audia, kde hodnota **1.0** značí skutečnou rychlost audia, výchozí hodnotou je **1.4** (= 140%), tato hodnota byla získána na základě uživatelského testování viz kapitola 6. Atribut může nabývat hodnot v rozmezí `<0.5, 3.0>`.

## Ukázkové spuštění aplikace

Pro dotvoření představy o fungování aplikace je třeba uvést příklady, jak lze danou aplikaci spustit. Pro spuštění je třeba základní soubor aplikace se jménem `xml_to_epub.py`, který se nachází v kořenové složce zdrojových souborů. Tento soubor načítá vytvořenou knihovnu, nastavuje časové pásmo pro správné vygenerování času vzniku e-knihy. Dále vytváří základní instanci aplikaci (třídy `Base`) a nastavuje `callback`, který odstraní dočasné soubory, při ukončení skriptu uživatelem.

Jelikož námi vytvořený balíček obsahuje řadu výjimek, kde podstatná je `XmlToEpub.FatacErrorException`, která je aplikací vyvolána v případě kritické chyby, je třeba tuto výjimku odchyťovat blokem `try ... except`<sup>10</sup>.

Nyní si již představíme příklady spuštění, za proměnnou `URL` si můžeme dosadit referenční vstup, který se nachází na webovém portálu **SuperLectures.com** - `http://www.superlectures.com/fit-zre/api/getLecture?id=1`

- Vytisknutí nápovědy: `./xml_to_epub.py --help`
- Minimální spuštění: `./xml_to_epub.py -i URL`
- Ovlivnění rychlosti audio stopy: `./xml_to_epub.py -i URL -s 1.1`
- E-kniha bez audio stopy: `./xml_to_epub.py -i URL -a`
- Ovlivnění názvu výstupního souboru: `./xml_to_epub.py -i URL -o text.epub`

---

<sup>10</sup><https://docs.python.org/3.4/tutorial/errors.html>

## Kapitola 6

# Testování

Pro kvalitní zhodnocení odvedené práce, bylo třeba zanalyzovat úspěšnost vyvinutých algoritmů, a to zejména co se týče úprav zvukového záznamu přednášky.

### 6.1 Testování úprav zvukové stopy

Úspěšnost úprav audio souboru byla otestována osmi vzorovými soubory. Výsledky si můžeme prohlédnout v následující tabulce. Koeficient rychlosti zvukové stopy byl ponechán na hodnotě 1.0, jinými slovy rychlost nebyla změněna, jelikož by měla zřetelný vliv na níže uvedenou tabulku.

	1	2	3	4	5	6	7	8	Průměr
<b>Vstupní audio</b>									
Počet snímků	30	58	37	16	37	48	46	5	<b>34.6 ks</b>
Délka audia	109	107	109	49	101	105	94	101	<b>97 min</b>
Velikost (MP3)	37.4	36.8	37.5	17.0	34.7	36.3	32.3	34.9	<b>33.4 MB</b>
<b>Výstup</b>									
Vyřiznutých úseků	407	481	509	226	641	601	537	472	<b>484.3 ks</b>
Délka audia	99	99	100	45	82	91	76	86	<b>85 min</b>
Velikost (MP3)	22.5	22.6	22.9	10.2	18.7	20.9	17.4	19.7	<b>19.4 MB</b>
Velikost (M4A)	34.4	34.5	34.9	15.6	28.6	31.9	26.6	30.1	<b>29.6 MB</b>
Velikost e-knihy	35.8	36.7	36.3	15.8	30.1	33.2	28.5	30.0	<b>30.8 MB</b>

Tabulka 6.1: Výsledek testování úpravy zvukového záznamu přednesu

#### Vyhodnocení testu

Z výsledků testu je patrné, že u průměrného souboru se vyřízne přibližně **5,7** úseků za minutu. Dále lze vypočítat, že po provedení algoritmu se celkový čas audia zkrátil o **12%**. Rozdíl ve velikosti výsledného audia je u formátu MP3 velmi uspokojivý, jelikož se zmenšila průměrně o **42%**. Bohužel formát MP3 není pro e-knihu vhodný, jelikož výstupní soubor má nízkou přenosovou frekvenci a nastává tak problém se synchronizací, následkem toho se audio spustí v nesprávný čas. Řešením problému je převod souboru do formátu M4A, který tyto problémy nemá.



Výsledky testů jsou uspokojivé, a to i přes zvýšení velikosti výsledné audio-stopy způsobené převodem do formátu M4A, ale i přesto dosahujeme průměrného zmenšení velikosti zvukové stopy o **11%**.

## 6.2 Testování optimální rychlosti

Dalším předmětem testování bylo získat ideální hodnotu zrychlení zvukové stopy z pohledu uživatele. Důvod hledání defaultní hodnoty zrychlení byl ten, že vycházíme z předpokladu, že uživatel má zájem projít přednášku relativně rychle. Zároveň však bylo třeba zachovat přívětivou srozumitelnost. Pro tento test bylo využito 8 osob, které dostaly několik verzí úryvku e-knih, které se lišily rychlostí audio výstupu. Srovnání reakcí uvádím v tabulce 6.2.

Testovaný uživatel vyjadřoval na stupnici 0-10, jak mu přijde daná rychlost srozumitelná. Nakonec zvolil rychlost, která by mu vyhovovala nejvíce, a to s přihlédnutím na jeho osobní preference. Vzorky byly dány v různém pořadí, aby nebyl úsudek ovlivněn pravidelnou posloupností změn.

Rychlost	1	2	3	4	5	6	7	8	Průměr
1.0x	10	10	10	10	10	10	10	10	<b>10.0</b>
1.1x	10	10	10	10	10	10	9	10	<b>9.875</b>
1.2x	9	10	10	10	10	9	10	10	<b>9.75</b>
1.3x	9	10	9	9	10	10	10	10	<b>9.625</b>
1.4x	10	10	9	10	10	9	9	10	<b>9.625</b>
1.5x	8	7	8	8	8	6	8	9	<b>7.75</b>
1.6x	5	5	6	7	6	7	6	7	<b>6.125</b>
1.7x	3	2	5	4	5	3	5	4	<b>3.875</b>
1.8x	1	1	1	2	1	3	2	4	<b>1.875</b>
Preference	1.4x	1.4x	1.3x	1.4x	1.4x	1.3x	1.4x	1.5x	<b>1.3875</b>

Tabulka 6.2: Test průměrné optimální rychlosti zvukové stopy

### Vyhodnocení testu

Dle výsledků je zřejmé, že uživatel občas nerozezná malou nuanci mezi rychlostmi lišícími se pouze o 10%. Nicméně můžeme vyvodit výsledek, že průměrná optimální rychlost pro poslech je přibližně 1.4x, tedy **140%** původní rychlosti.

# Kapitola 7

## Závěr

Cílem této práce bylo vytvořit konzolovou aplikaci, která zprostředkuje konverzi dat reprezentující záznam přednášky do formátu ePub. Pro realizaci jsem si vybral programovací jazyk Python3, který se po důsledné analýze zdál jako nejvhodnější. Bylo třeba především navrhnout vhodný algoritmus pro úpravu audio záznamu přednášky, který bylo třeba optimalizovat. Optimalizace proběhla v podobě možnosti ovlivnit rychlost daného zvukového záznamu a také proběhla detekce a odstranění tzv. *hluchých míst*, neboli místa kde je delší sekvence ticha či šumu. Díky těmto úpravám se poslech e-knihy stává více přívětivým.

Problém při implementaci byl především v nalezení vhodné kombinace XHTML struktury a CSS vlastností, aby byla výsledná e-kniha použitelná v co největším počtu aplikací - především bylo třeba vyřešit přizpůsobení velikosti snímku velikosti a natočení displeje.

Jakmile se do ePub struktury přidal i audio soubor, nastaly další problémy, které bylo třeba vyřešit. Jeden z nich byl navázání zvuku na daný snímek, což se podařilo vyřešit díky možnosti překrývání médií (viz podkapitola 5.3.2). Dalším příkladem může být špatná synchronizace MP3 souboru s daným snímkem. Řešením byla konverze souboru do formátu M4A, kde se tento problém již neprojevil.

### 7.1 Zhodnocení výsledku práce

Dle mého osobního názoru má formát ePub budoucnost, ale pouze za předpokladu, že se nastaví standard pro čtecí aplikace a vytvoří se více volně šířitelného softwaru. Jako nedostatek ePub shledávám absenci odborné literatury s komplexními příklady. Jediným oficiálním studijním materiálem je dokumentace projektu. Vývojář je poté odkázán pouze na onu dokumentaci a neoficiální rady a domněnky jiných vývojářů.

Zároveň jsem rád, že jsem mohl vytvářet aplikaci, která má skutečný potenciál aktivního využití. Tímto nástrojem se významně rozšíří možnosti off-line prohlížení přednášek v mobilních zařízeních. Díky propojení zvukového záznamu s jednotlivými snímky jsme získali velikostně přijatelný soubor pro mobilní zařízení, který má dle testů průměrnou velikost 30.8 MB pro 85 minutovou přednášku bez zrychlení audio-stop, což je přijatelnější, než kompletní video záznamu, který má velikost v řádu stovek MB.

V původním plánu jsem měl i otestování projektu na zařízeních Kindle, avšak toto zařízení nevlastním a nepodařilo se mi ho ani vypůjčit. Důvod byl také ten, že ne všechna Kindle zařízení podporují zvukovou stopu. Takže toto utváří mírnou mezeru, která by se mohla zaplnit v dalším vývoji aplikace.

## 7.2 Návrh na následný posun aplikace

Aplikace dle mého názoru plní požadovanou funkcionalitu a tedy mě nenapadá žádné významnější rozšíření řešení. Pro usnadnění práce s aplikací se však nabízí následující vylepšení:

- Vytvoření GUI pro snadnější práci s aplikací.
- Detekce optimálního zrychlení pro každého řečníka zvlášť. Toho lze dosáhnout detekcí délek samohlásek a porovnání s optimální délkou. Toto rozšíření jsem však neimplementoval.

Vzhledem k tomu, že jsem projekt vyvíjel především pro operační systém Android, tak největší podpora aplikací je právě u této platformy. Ostatní platformy jsou také podporované, ale bylo by dobré provést podrobnější testování aplikací pro operační systémy iOS, Windows Phone a Kindle. Tato analýza by mohla zapříčinit drobné úpravy aplikace - například by mohl vzniknout přepínač, který by uzpůsobil výslednou aplikaci optimalizovanou pro některou platformu, která může vyžadovat mírné odlišnosti v provedení.

Při vývoji aplikace jsem odhalil pouze jednu závažnější chybu ve vstupních datech. Jednalo se o chybějící uvedení koncového bodu ve zvukovém záznamu. V následující podsekcí tuto chybu podrobněji popíši a navrhnou možná řešení.

### Absence uvedení konce snímku

Ve vstupním XML souboru je XML značka `slide`, která reprezentuje právě jeden snímek, respektive stránku, vytvářené e-knihy. Tato značka obsahuje atribut `start`, který udává bod ve zvukovém záznamu, kdy začíná přednášející mluvit o daném snímku. Některé přednášky<sup>1</sup> se skládají nejen ze snímků prezentace, ale také jsou doprovázeny interaktivní plochou, kde přednášející kreslí schémata, počítá příklady, či jinak doprovází výklad. Následkem toho v e-knize uslyšíte u snímku výklad, který nemusí souviset s tímto snímkem a můžete si jen domýšlet, co se na přednášce dělo.

Navrhuji jedno z následujících řešení:

- (a) Přidání atributu `end`, který určí konečný bod přednesu o daném snímku, aktuálně se za konečný bod považuje počáteční bod následujícího snímku.
- (b) Uchování výstupu interaktivní plochy a jejího doplnění jako dalšího snímku prezentace.

Naneštěstí obojí řešení s sebou nesou další možná úskalí. Vymyslet řešení, které by aplikaci nijak neomezovalo, se zdá velmi obtížné, ale bylo by dobré se nad tímto problémem zamyslet.

---

<sup>1</sup>Příklad takové přednášky nalezneme zde: <http://www.superlectures.com/fit-zre/zre-prednaska-9>

# Literatura

- [1] Schwarz, P.: *Phoneme recognition based on long temporal context*. Brno University of Technology, 2008.
- [2] Summerfield, M.: *Python3 Výukový kurz*. Computer Press, a.s., 2011, ISBN 978-80-251-2737-7.
- [3] WWW stránky: Definice formátu EPUB. <http://cs.wikipedia.org/wiki/Epub>.

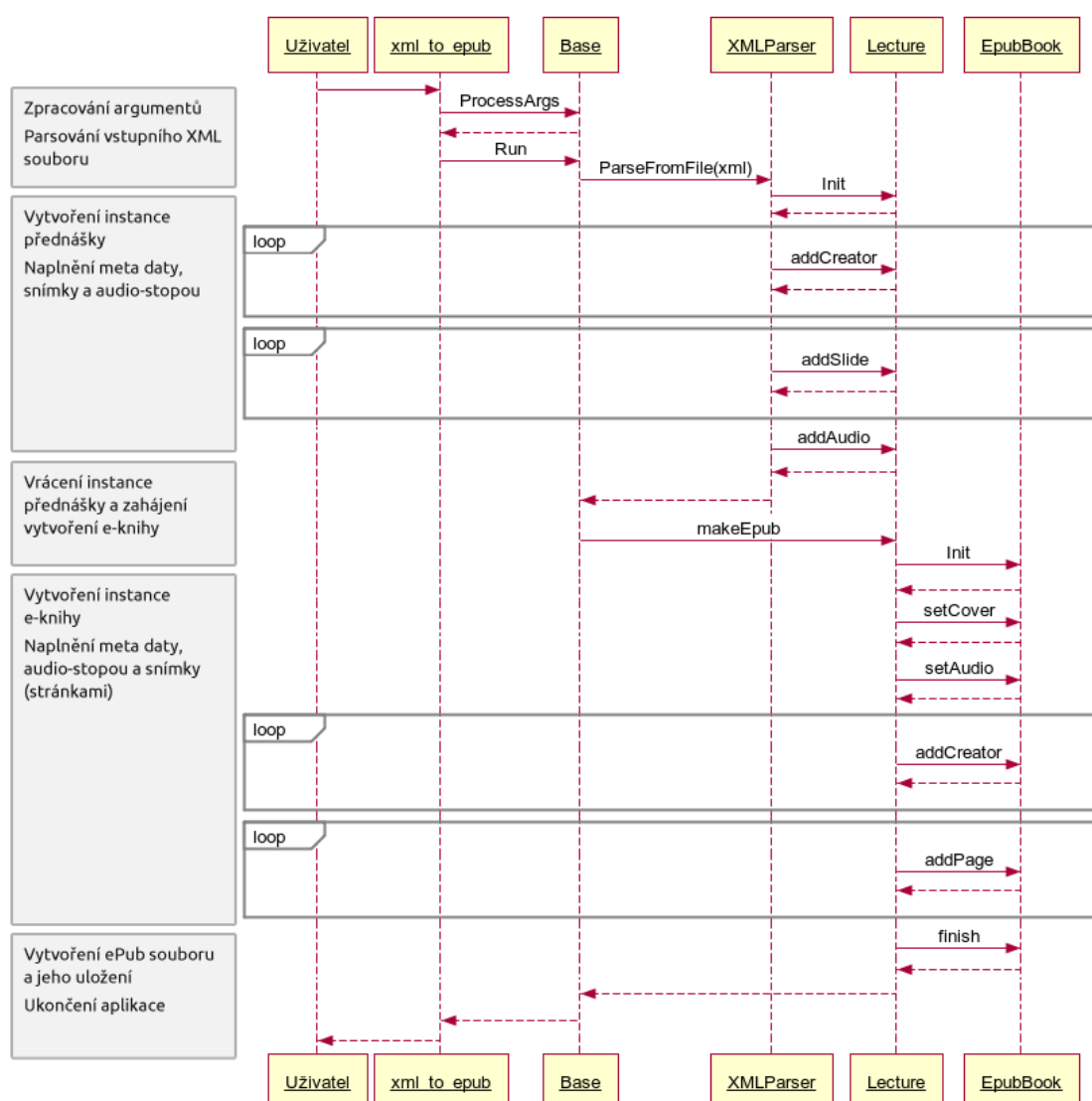
# Příloha A

## Obsah CD

- /src/ - složka se zdrojovými soubory
- /doc/ - zdrojové soubory písemné zprávy
- /xadame36-BP.pdf - elektronická verze písemné zprávy
- /poster.pdf - plakát k aplikaci
- /README.txt - pokyny k ovládání aplikace
- /video.mp4 - doprovodné video k aplikaci
- /zre\_2.epub - ukázkový výstup aplikace

## Příloha B

# Sekvenční diagram aplikace



Obrázek B.1: Sekvenční diagram aplikace

## Příloha C

# Optimální schéma XML souboru

```
<data>
  <lecture [id] [name] [url]>
    <authors>
      <person [name]>...</person>
      :
      :
      <person [name]>...</person>
    </authors>
    <speakers>
      <person [name]>...</person>
      :
      :
      <person [name]>...</person>
    </speakers>
    <datetime_recorded>[text]</recorded>
    <thumbnail_large [src] />
    <audio [duration]>
      <audio_mp3 [src] />
    </audio>
    <slides [number_of_slides]>
      <slide [start] [src_large] />
      :
      :
      <slide [start] [src_large] />
    </slides>
  </lecture>
  :
  :
  <lecture [id] [name] [url]>...</lecture>
</data>
```

## Příloha D

# Struktura ePub

### Obecná struktura

- mimetype
- META-INF/
  - container.xml
- OPS/
  - package.opf
  - toc.xhtml
  - ...ostatni soubory (audio, stránky, obrázky)

### Struktura naší aplikace

- mimetype
- META-INF/
  - container.xml
- OPS/
  - audio/audio.m4a
  - image/\*
  - text/\*
  - package.opf
  - toc.xhtml
  - stylesheet.css



# Příloha E

## Instalace

Pro spuštění aplikace je třeba mít programovací jazyk `Python3` ve verzi 3.4.0, popřípadě novější. Jelikož program využívá i jiné knihovny, je třeba zajistit jejich podporu. Všechny uvedené příkazy jsou platné pro OS Ubuntu 14.04<sup>1</sup>, na kterém jsem aplikaci vyvíjel. Více informací o nástrojích uvádím v podkapitole 4.1.4 a zde je již nebudu podrobněji popisovat. Níže uvádím seznam potřebných komponent a návod na jejich instalaci:

- `PhnRec` - rozpoznávač fonémů potřebuje pro svůj běh knihovnu `libcbblas.so`, kterou je možno nainstalovat za pomoci příkazu: `sudo apt-get install libatlas-base-dev`
- `SoX` - editor zvukových souborů, instalaci provedeme příkazem: `sudo apt-get install sox`
- `ffmpeg` - potřebný pro převod audio souboru z formátu MP3 do M4A. Nástroj bohužel není ve standardním repositáři<sup>2</sup> a tak je jeho instalace složitější. Je ji nutno provést v následujících třech krocích, kdy přidáme nový repositář, dále aktualizujeme všechny stávající repositáře a poté nainstalujeme nástroj `ffmpeg`:

```
sudo add-apt-repositor ppa:mc3man/trusty-media
sudo apt-get update
sudo apt-get install ffmpeg gstreamer0.10-ffmpeg
```

Spuštění aplikace již probíhá jako standardní python aplikace. Ovlivnit chování programu můžete za pomoci přepínačů. Více informací o spuštění aplikace a popis přepínačů najdete v podsekcí 5.5.

---

<sup>1</sup><http://releases.ubuntu.com/14.04/>

<sup>2</sup>Pro Ubuntu 15.04 by již měl být opět v standardním repositáři viz. <http://www.linuxexpres.cz/novinky/ffmpeg-se-vraci-do-ubuntu>