

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.Szóke Igor, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Martin Adamec

20. května 2015

Poděkování

Chtěl bych poděkovat vedoucímu mé bakalářské práce Ing. Szóke Igor, 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

1	Úvod	3
2	Popis problému	4
2.1	Struktura zpracovávaného XML souboru	4
2.2	Existující řešení	5
3	Formát ePub	7
3.1	Vlastnosti ePub	7
3.2	Struktura ePub	8
3.3	ePub a práce se zvukem	8
3.4	Aplikace pro čtení formátu ePub	9
3.4.1	Přehled aplikací	10
4	Návrh řešení	11
4.1	Použité technologie	11
4.1.1	Programovací jazyk Python3	11
4.1.2	Šablonový systém jinja2	12
4.1.3	Git	12
4.1.4	Nástroje pro úpravu zvukové stopy	12
4.2	Návrh aplikace	13
5	Implementace	14
5.1	Struktura aplikace	14
5.2	Zpracování vstupního souboru	15
5.3	Práce se zvukovou stopou	15
5.4	Naplnění struktury ePub	16
5.4.1	Třída EpubBook	16
5.4.2	ePub a možnost překrývání objektů	17
5.5	Validace výsledné aplikace	18
5.6	Spuštění aplikace	18
6	Testování	20
7	Závěr	22
7.1	Zhodnocení výsledku práce	22
7.2	Návrh na následný posun aplikace	23
A	Obsah CD	25

B	Minimální schéma XML souboru	26
C	Struktura ePub	27
D	Instalace	28

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 našim 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 funkcionalitu. 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 i sama pamatuje, kde čtenář skončil. Toto jsou základní výhody elektronických knih a také asi důvody, proč se stávají v dnešní době tak populární.

V námi vytvářené aplikaci se budeme zabírat formátem *ePub*¹, který je jedním z nejčastěji používaným formátem pro elektronické knihy. Tento formát nám díky verzi 3, která byla uveřejněna v roce 2011, 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, tak to má za následek odlišnou interpretaci dat (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 rozšířený 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. O postupu jak jsem této konverze dosáhl a jaké problémy bylo třeba řešit, se Vám budu snažit přiblížit v následujících kapitolách.

¹<http://idpf.org/epub/301>

Kapitola 2

Popis problému

Aktuálním trendem se stává vytváření záznamů přednášek a jejich následné poskytnutí omezené či celé veřejnosti. Jeden ze serverů, který nabízí tyto přednášky ke shlédnutí a stažení v nejrůznějších podobách je webový portál *SuperLectures.com*. Fakulta informačních technologií VUT, která je zadavatelem této práce, tento portál také využívá, ať už pro záznam z přednášek vyučovaných předmětů nebo záznamy nejrůznějších konferencí a externích přednášek.

Pro tuto práci mi byly poskytnuty data přednášky, uložené v XML¹ souboru (viz 2.1), který má přesně danou strukturu. Z tohoto souboru můžeme získat audio stopu přednášky, promítané snímky a další data, které jsme použili při vytváření naší aplikace.

I přes rozmanitost poskytnutých formátů ke stažení, tak je absence formátu, který by byl ideální pro přenosná zařízení, zejména s ohledem na velikost dat. Video záznam, které má velikost v řádu stovek megabajtů, není vhodné pro tento účel. Jako zajímavá alternativa se naskytlo využití formátu elektronické knihy, konkrétně se jedná o formát **ePub** (více o tomto formátu v kapitole 3). Tento formát je ideálním pro spojení přednášených snímků a audio záznamu dané přednášky. Jelikož se výsledný soubor bude skládat pouze z komprimovaných obrázků a audio souboru, tak se splní požadavek na malou velikost výsledného souboru.

Zmíněný formát nabízí od roku 2011, kdy publikovala svou třetí verzi, synchronizaci zvoleného elementu (obrázek, text, ...) s audio stopou. Zvolená audio stopa, se v momentě zobrazení elementu spustí od nastaveného času. Díky této funkcionalitě můžeme vytvářet e-knihy, které čtenářovi předčítají zobrazený text a tento text je i současně zvýrazněn.

Chování ePub formátu je specifický pro každou aplikaci, což ze subjektivního pohledu toto vnímám jako největší slabinu tohoto formátu. A především ne všechny aplikace jsou schopny přehrát i audio stopu, ale o tom již v sekci 3, která se věnuje formátu ePub podrobněji.

2.1 Struktura zpracovávaného XML souboru

Jak již bylo řečeno, program získává informace pro svůj chod z XML souboru. Strukturu tohoto souboru nalezneme v příloze B. Tato struktura byla určena dle referenčního vstupního souboru k této práci, kterou byla přednáška z předmětu 2. přednáška z předmětu Zpracování řečových signálů², která je uveřejněna na SuperLectures.com. Význam jednotlivých

¹eXtensible Markup Language

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

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
- **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 w3c³
- **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 nás však u této aplikace nezajímají 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**.

2.2 Existující řešení

Volně dostupnou aplikaci, ať už konzolová či klasická, která by řešila stejnou problematiku, tedy vytvoření e-knihy (formátu ePub) se zvukovou stopou jsem nenašel. Dle mého, je hlavní příčina to, že pokud nějaká firma/instituce potřebuje takovouto konverzi, tak má, stejně jako v této práci, přesně danou vstupní strukturu a výsledná e-kniha má specifický

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

účel. Jako příklad můžeme uvést firmu **Sinkronigo**⁴, která se zabývá vytvářením e-knih, kde zobrazený text je předčítán, avšak jejich aplikace není volně dostupná.

Pokud by si běžný uživatel chtěl vytvořit e-knihu, tak to většinou spočívá v převodu souboru PDF či doc, která obsahuje pouze text a obrázek. Nástroje pro tuto základní konverzi existuje nesčetně mnoho a některé jsou i online jako je například konvertor na webu Online-Convert.com⁵. Díky univerzálnosti těchto nástrojů nebývá výsledek příliš kvalitní a je vhodný pouze pro osobní použití.

Také pro LibreOffice Writer⁶ je dostupný plugin Writer2ePub⁷, 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 zmíníme ještě detailněji, a nebo aplikace **Sigil**⁸, která má obdobnou funkcionalitu.

Nástroj Calibre

Calibre⁹ 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 GitHub¹⁰. Jádro aplikace je napsáno v jazyku Python a stalo se mým studijním materiálem při tvoření aplikace.

Díky této aplikaci si můžeme 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 nemá aktuálně zřejmě konkurenci a to především díky tomu, že si můžete zde otevřít zdrojové soubory e-knihy z knihovny a upravovat text, měnit obrázky a spoustu dalšího. Lze tedy předěláním již existující e-knihy, vytvořit si svou vlastní e-knihu. Bohužel nenabízí možnost vytvořit si vlastní e-knihu tím, 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. Běžný uživatel se pravděpodobně nikdy nesetká s potřebou udělat si složitější vlastní e-knihu.

⁴<http://www.sinkronigo.com/>

⁵<http://ebook.online-convert.com/convert-to-epub>

⁶<http://www.libreoffice.org/discover/writer/>

⁷<http://extensions.services.openoffice.org/en/project/Writer2ePub>

⁸<http://sigil-ebook.com/>

⁹<http://calibre-ebook.com/>

¹⁰<https://github.com/kovidgoyal/calibre>

Kapitola 3

Formát ePub

Formální definice formátu ePub¹:

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 pracujeme 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 většiny známých platforem.

Díky podpoře formátu HTML5 můžeme v e-knize mít audio či video soubory, které lze synchronizovat s textem, obrázek atd., což je stěžejní pro naši 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 nabízí odkázat na jiný multimediální soubor v případě, že daný interpret (čtečka/aplikace) nepodporuje onen typ souboru. To znamená, že pokud bychom měli 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.org² a s neoficiálními poznatky na internetových fórech.

¹<http://cs.wikipedia.org/wiki/Epub>

²<http://idpf.org/epub/30>

3.2 Struktura ePub

Jak již bylo řečeno ve formální definici, tak ePub soubor je pouze ZIP soubor s pevně danou strukturou - tuto základní strukturu a také kompletní strukturu, která je použita v naší aplikaci můžeme najít v příloze C. 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

Jak jsme zmínili dříve, tak ePub verze 3 podporuje HTML5 značku `audio`³. 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 od shora dolů a aplikace zvolí první zdroj, který je schopna přehrát. Pokud aplikace neumí pracovat ani s jedním ze zdrojů, tak zobrazí informaci, kterou můžeme uvést 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>
```

³Více o značce `audio` nalezneme zde: http://www.w3schools.com/html/html5_audio.asp

3.4 Aplikace pro čtení formátu ePub

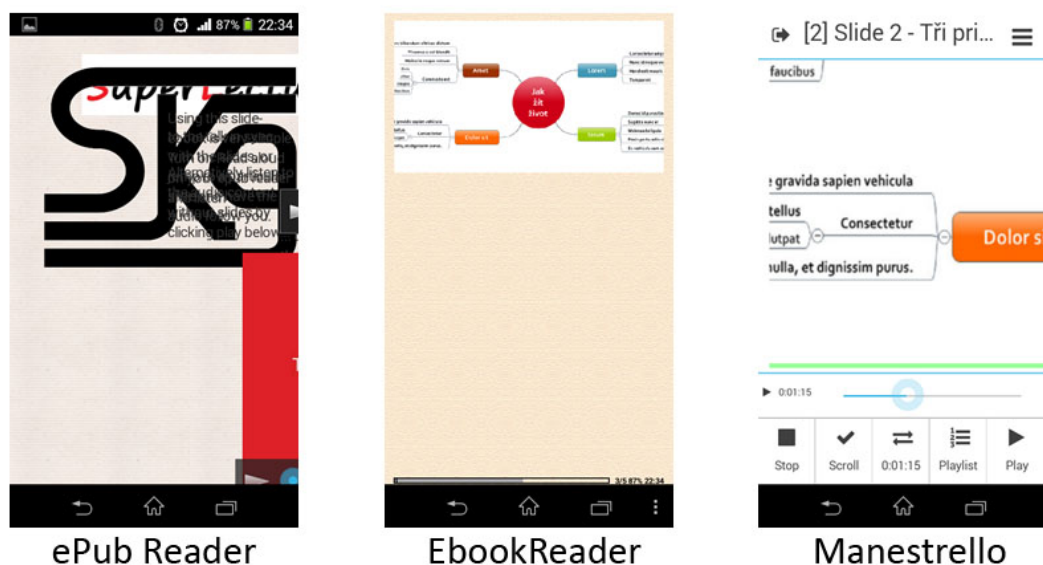
Téměř všechny moderní platformy (Android, iOS, Windows, různé Linux distribuce, Kindle ...) dokáží pracovat s formátem ePub, některé to zvládají již v základní instalaci, jiné za pomoci aplikace tvůrce operačního systému (OS) nebo třetí strany. Náš program se zaměřuje především na operační systém Android,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

Aplikací je nepřehledné množství a vybráním toho optimálního 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 co nejoptimálnější 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).

Odlišné interpretace aplikací

Příklad odlišné interpretace e-knihy⁴, který mi byla zadána jako referenční, můžete najít na následujícím obrázku (3.1). 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 3.1: Odlišná interpretace e-knihy

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

⁴Bar Camp Brno 2014 - Tři principy veřejného mluvení - záložka "Stáhnout/Sdílet" položka EPUB

3.4.1 Přehled aplikací

Níže uvedená tabulka ukazuje příklady aplikací a jejich podporu funkcí.

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

Tabulka 3.1: Srovnání aplikací pro čtení ePub

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

^a[Knihy Google](#) na Google Play

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

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

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

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

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

Kapitola 4

Návrh řešení

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

4.1 Použité technologie

Před započítí 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, tak technologie byla zvolena dle subjektivního uvážení, i s ohledem na dosavadní zkušenosti s daným nástrojem.

4.1.1 Programovací jazyk Python3

Jako první si bylo třeba zvolit programovací jazyk. S přihlédnutím faktu, že výsledný program je konzolový, tak jako nejlepší varianta se naskytl jazyk Python3, s kterým jsem měl již zkušenosti. Tento jazyk je také již v základu každé linuxové distribuce. Učební literaturou se stala kniha *Python3 Výukový kurz*^[1] a také oficiální dokumentace¹.

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

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

¹<https://docs.python.org/3/>

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

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

⁴<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 nabízeli následující možnosti:

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

S přihlédnutím na přehlednost a snadnou úpravu jsem zvolil šablonový systém, konkrétně se jedná o knihovnu `jinja2`⁶. Tato knihovna byla vybrána na základě referencí a mého subjektivního rozhodnutí. `jinja2` je moderní knihovna, která vychází z Django⁷ 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.

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

Volně dostupných verzovacích systémů je více, avšak nejznámějšími jsou GitHub⁸ a BitBucket⁹. Zvolil jsem BitBucket, jelikož s tímto nástrojem mám dlouholetou zkušenost.

4.1.4 Nástroje pro úpravu zvukové stopy

Dle zadání této práce bylo třeba i 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`¹⁰, 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 *bitrate* 16bit/s. Do této podoby převedeme MP3 soubor, který nám byl dán jako vstupní pro naši aplikaci, za pomoci konzolového audio editoru SoX.

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

⁶<http://jinja.pocoo.org/>

⁷<https://www.djangoproject.com/>

⁸<https://github.com/>

⁹<https://bitbucket.org/>

¹⁰<http://speech.fit.vutbr.cz/cs/software/phoneme-recognizer-based-long-temporal-context>

Audio editor SoX

SoX¹¹ 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

Vstupní MP3 je třeba převést do formátu M4A, což nám provede linuxová knihovna `ffmpeg`¹². Formát MP3 pro e-knihu formátu ePub není vhodný, jelikož při nižší frekvenci se audio nesprávně synchronizuje. Toto jde vyřešit zvýšením frekvence, avšak to způsobí výrazný nárůst velikosti souboru.

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 bude vypisovat na `stdout`¹³ 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, tak můžeme předpokládat, že tato skupina bude mít pokročilé znalosti v užívání konzolových aplikací.

Struktura aplikace

Aplikace bude 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.

¹¹<http://sox.sourceforge.net/>

¹²<https://www.ffmpeg.org/>

¹³standardní výstup

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 Struktura aplikace

Aplikace je psána jako Python balíček, kde 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 můžete najít na obrázku 5.1.



Obrázek 5.1: Struktura aplikace `XmlToEpub`

Nad aplikací je vytvořeno základní rozhraní, které určuje několik vytvořených tříd. Jedná se o následující třídy:

- `Base` - základní třída, která zpracovává argumenty a řídí celou konverzi, také má na starosti úklid po dokončení či přerušení programu

- **Core** - rodičovská třída většiny tříd, která byla vytvořena pro zpracování inicializačních argumentů tříd
- **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

5.2 Zpracování vstupního souboru

Jak již bylo řečeno v předešlé kapitole (2), tak vstupem je url XML souboru, který v sobě nese data přednášky, která má být převedena do formátu ePub. Více informací o struktuře můžete najít v podkapitole 2.1. Tento soubor je pro zpracování třeba stáhnout a za pomoci standardní Python3 knihovny `ElementTree`¹ získáme požadované data.

Parsování XML souboru provádím ve třídě `XmlTools.XmlParser`, která nám 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. V naší aplikaci při přidání snímku jsou prohledány snímky, které byly přidány dříve. V případě shody URL snímku, se daný snímek znovu nestahuje, ale uloží si nalezený výskyt jako svého rodiče.

5.3 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 je prováděno v pomocné třídě `Audio.AudioTools`.

Princip detekce a odstranění hluchých míst

Před vytvořením algoritmu pro rozpoznání těchto úseků bylo třeba si stanovit co vlastně budeme považovat za *hluché místo*. To vedlo k vytvoření dvou konstant v třídě `AudioTools`:

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

V první řadě bylo nutné převést zvukový záznam na fonémy, což nám umožnila knihovna `PhnRec`, která je součástí výsledné aplikace. Více o této knihovně je uvedeno podsekcí 4.1.4. Naším cílem bylo najít v seznamu fonémů takové identifikátory fonému, které značí ticho nebo šum. Fonémy `pau` reprezentující ticho. Šum určují fonémy `int` a `spk`. Pokud jsme tedy našli sekvenci fonémů ticha či šumu, které přesahuje délku určenou konstantou `MIN_PAUSE`, tak přidáme časové razítko začátku a konce výřezu do pole, pro pozdější zpracování.

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

Samotné vyříznutí hluchých míst se provádí za pomoci knihovny `SoX` (viz 4.1.4). Princip spočívá ve vyříznutí částí, které zůstávají, které se získají za pomoci seznamu vyříznutých částí. Jednotlivé ponechané části se následně spojí opět pomocí knihovny `SoX` a tím vznikne výsledný zvukový soubor formátu MP3.

Změna rychlosti

Uživatel aplikace může ovlivnit rychlost zvukové stopy a to za pomoci přepínače `-s`, respektive `--speed` (více viz 5.6). 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`.

5.4 Naplnění struktury ePub

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

5.4.1 Třída `EpubBook`

Samostatná třída vytvářející strukturu ePub (více o tomto formátu můžeme nalézt v kapitole 3), která není závislá na zbytku projektu². Její samostatnost je cílená, jelikož se tímto stává snadno přenositelná pro použití v jiných projektech. Třída je, ale vytvořena pro naši konkrétní aplikaci, takže má své specifické metody a chování. Proto si nyní popíšeme více její 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

Naplnění daty

Data se plní přes navržené metody třídy, čímž se pouze nastaví vlastnosti e-knihy a jeho budoucí obsah. Až zvolením 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.

²Vazbu má pouze na šablonový systém, který se ovládá pomocí třídy `Template`

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 za pomoci byla vytvořena třída `Template`, která poskytuje metody pro snadnou práci s daným šablonovým systémem.

5.4.2 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**³. 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 uveden na obrázku níže (5.2):

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

Obrázek 5.2: Neúplný vzor `smil` souboru

Za pomoci tohoto souboru se vytvoří objekt nad stránkou `slidex-{x}.html`, který je spjat s elementem nesoucí ID `p000000`. Díky tomuto objektu se při aktivaci objektu s 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

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 úseku⁴, 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.

³<http://www.idpf.org/epub/30/spec/epub30-mediaoverlays.html>

⁴Podoba meta tagu: `<meta refines="#slide-x.smil"property="media:duration">HH:MM:SS.MS</meta>`

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 (5.3):

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

Obrázek 5.3: Provázání stránky se smil objektem

5.5 Validace výsledné aplikace

Po vytvoření e-knihy, bylo třeba ověřit, zda je vytvořena správně. K tomuto účelu slouží oficiální online validátor na webu IDPF.org⁵. 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`⁶ nainstalovat⁷ do své linuxové distribuce. Instalací získáme pohodlnější validaci a odpadá nutnost soubor nahrávat na server třetí strany.

5.6 Spuštění aplikace

Jak jsme již dříve zmínili, tak aplikace je konzolová a určená pro operační systém linux. Pro svůj běh vyžaduje přítomnost několika nástrojů, informace o které nástroje se jedná a popis jejich získání naleznete v příloze D.

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 popíšeme 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
- `-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 **1.19** (= 119%), tato hodnota byla získána na základě uživatelského testování viz kapitola 6

⁵<http://validator.idpf.org/>

⁶<https://github.com/IDPF/epubcheck>

⁷Instalace příkazem: `apt-get install epubcheck`

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, který se nachází v kořenové složce zdrojových souborů, se jmenuje `xml_to_epub.py`. Tento soubor načítá vytvořenou knihovnu, nastavuje časové pásmo pro správné vygenerování času vzniku e-knihy. Dále vytváří základní instanci aplikace (třídy `Base`) a nastavuje `callback` při ukončení scriptu uživatelem, který odstraní dočasně vytvořené soubory.

Jelikož námi vytvořený balíček obsahuje řadu výjimek, kde nejhlavnější je `XmlToEpub.FatacErrorException`, kterou aplikace vyhodí v případě kritické chyby, je třeba tuto výjimku odchyťovat blokem `try ... except`⁸.

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`

⁸<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.

Testování úprav zvukové stopy

Úspěšnost úprav audio souboru byl otestován nad 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 příliš ovlivnila níže uvedenou tabulku.

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

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 provedené algoritmu se celkový čas audia zkrátí o **12%**. Rozdíl ve velikosti výsledného audia je u formátu MP3 velmi uspokojivý, kde se zmenšilo průměrně o **42%**. Bohužel formát MP3 není pro e-knihu vhodný, jelikož výstupní soubor má nízkou frekvenci a nastává tak problém se synchronizací, kdy audio se spustí v odlišný čas než má. Řešení problému je převést soubor do formátu M4A, který tyto problémy nemá.

S výsledky testů jsem spokojen a to i přes zvýšení velikosti audio souboru, způsobené převodem do formátu M4A, ale i přesto dosahujeme průměrného zmenšení o **11%**.

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 10 osob, které dostali několik verzí úryvku e-knih, které se lišily rychlostí audio výstupu. Na srovnání reakcí se podíváme v níže uvedené tabulce 6.2.

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

Rychlost	1	2	3	4	5	6	7	8	9	10	Průměr
1.0x	10	10	10	10	10	10	10	10	10	10	10.0
1.1x	10	10	10	10	10	10	9	10	9	10	9.8
1.2x	9	10	10	9	10	9	10	10	9	10	9.6
1.3x	9	10	9	9	10	10	9	10	8	9	9.3
1.4x	7	7	8	7	8	8	8	8	9	8	7.8
1.5x	5	6	7	7	7	6	7	6	5	7	6.3
1.6x	3	4	6	4	5	4	5	4	4	6	4.5
1.7x	1	3	4	4	3	5	4	3	2	4	3.3
1.8x	0	1	1	2	1	3	2	1	0	2	1.3
Preference	1.2x	1.3x	1.2x	1.1x	1.3x	1.2x	1.2x	1.2x	1.1x	1.2x	1.19

Tabulka 6.2: Test optimální rychlosti zvukové stopy

Dle výsledků je zřejmé, že testovaný občas nerozezná malou nuanci mezi rychlostmi lišícími se pouze o 10%. Nicméně můžeme vyvodit výsledek, že optimální rychlost pro poslech je 1.19x, tedy **119%** 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 XHTML struktury a vhodných 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 nastali další problémy, které bylo třeba optimálně vyřešit. Jeden z nich byl navázání zvuku na daný snímek, což se podařilo vyřešit, díky možnosti překrývání médií (viz podkapitola 5.4.2). 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

S konečným výsledkem mé práce jsem spokojen. Naučil jsem se důkladněji jazyk python, který skýtá větší možnosti, než jsem kdy tušil. Dostalo se mi taky velkého množství znalostí o formátu ePub, který je sice známý, ale jen málo vývojářů s ním skutečně pracuje. Dle mého osobního názoru má tento formát budoucnost, ale pouze za předpokladu, že se nastaví standard pro čtecí aplikace a vytvoří se více volně šiřitelného softwaru. Co shledávám jako nedostatek ePub je absence 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 a nestane se to pouze *"projektem do šuplíku"*, což byl také můj cíl při výběru tématu bakalářské práce. 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, který má dle testů průměrnou velikost 30.8 MB, což je daleko více přijatelné než kompletní video záznamu, který má velikost v řádu stovek MB, do mobilního zařízení.

V původním plánu jsem měl i otestování projektu na zařízeních Kindle, avšak toto

zařízení nevlastním a nepodařilo se mi vypůjčit si toto zařízení. Důvod byl také v tom, že ne všechny 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. Ale jelikož 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 vznikl 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. Jednalo se o chybějící uvedení koncového bodu ve zvukovém záznamu. Detailněji si tuto chybu popíšeme za chvíli a popíšeme taky 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`, která udává bod ve zvukovém záznamu, kdy začíná přednášející mluvit o daném snímku. Některé přednášky¹ 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 vůbec souviset s tímto snímkem a můžete si jen domýšlet, co se na přednášce dělo.

Řešení se mi naskytují hned dvě:

- (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 doplnit to jako další snímky prezentace

Naneštěstí obojí řešení s sebou nese další možná úskalí. Pravděpodobně nelze vymyslet řešení, které by díky řešení aplikaci mírně neomezovalo, ale bylo by dobré se nad tímto problémem zamyslet.

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

Literatura

- [1] Summerfield, M.: *Python3 Výukový kurz*. Computer Press, a.s., 2011, iSBN 978-80-251-2737-7.

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

Příloha B

Minimální 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 C

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.mp4
 - image/*
 - text/*
 - package.opf
 - toc.xhtml
 - stylesheet.css

Příloha D

Instalace

Pro spuštění aplikace je třeba mít programovací jazyk `Python3`, ve verzi 3.4.0 či novější. Jelikož program využívá i jiné knihovny, tak je třeba zajistit jejich podporu. Všechny uvedené příkazy jsou platné pro OS Ubuntu 14.04¹, na kterém jsem aplikaci vyvíjel. Více informací o nástrojích jsme si řekli v podkapitole 4.1.4 a zde je již nebudeme podrobněji popisovat. Nyní si uvedeme seznam potřebných komponent a návod na jejich instalaci:

- `PhnRec` - rozpoznávač fonémů potřebuje pro svůj běh knihovnu `libcblas.so`, kterou můžeme 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 není² ve standardním repositáři a tak jeho instalace je složitější, je ji nutno provést v následujících třech krocích, kde 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 5.6.

¹<http://releases.ubuntu.com/14.04/>

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