



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

System pro elektronické ctečky knih

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: 1802T007 – Informační technologie

Autor práce: **Bc. Karel Šír**

Vedoucí práce: doc. Ing. Josef Chaloupka Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

System for e-Books

Master thesis

Study programme: N2612 – Electrical Engineering and Informatics

Study branch: 1802T007 – Information Technology

Author: **Bc. Karel Šír**

Supervisor: doc. Ing. Josef Chaloupka Ph.D.



ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Karel Šír**

Osobní číslo: **M16000181**

Studijní program: **N2612 Elektrotechnika a informatika**

Studijní obor: **Informační technologie**

Název tématu: **System pro elektronické čtečky knih**

Zadávací katedra: **Ústav informačních technologií a elektroniky**

Z á s a d y p r o v y p r a c o v á n í :

1. Seznamte se se specifikací používaných formátů pro elektronické čtečky knih.
2. Vytvořte (v C# WPF) aplikaci umožňující převod mezi různými typy formátů do elektronických čteček (Amazon Kindle atd.).
3. Součástí programu by mělo být i rozhraní pro vygenerování příslušného formátu z vlastního textu.
4. Navržený systém by měl obsahovat i inteligentní rozhraní pro archivaci elektronických knih.

Rozsah grafických prací: Dle potřeby dokumentace
Rozsah pracovní zprávy: cca 40-50 stran
Forma zpracování diplomové práce: tištěná/elektronická
Seznam odborné literatury:

- [1] Petzold, Ch.: Mistrovství ve Windows Presentation Foundation, COMPUTER PRESS, ISBN 9788025121412, 2008
- [2] Mareš, A.: 1001 tipů a triků pro C# 2010, COMPUTER PRESS, ISBN 9788025132500, 2011
- [3] Comparison of e-book formats:
https://en.wikipedia.org/wiki/Comparison_of_e-book_formats

Vedoucí diplomové práce: doc. Ing. Josef Chaloupka, Ph.D.
Ústav informačních technologií a elektroniky
Konzultant diplomové práce: Ing. Karel Paleček, Ph.D.
Ústav informačních technologií a elektroniky
Datum zadání diplomové práce: 19. října 2017
Termín odevzdání diplomové práce: 14. května 2018

prof. Ing. Zdeněk Plíva, Ph.D.
děkan



prof. Ing. Ondřej Novák, CSc.
vedoucí ústavu

V Liberci dne 19. října 2017

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.


Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 14.5.2018

Podpis: 

Abstrakt

Tato diplomová práce popisuje tvorbu aplikace pro převod formátů do elektronických čteček knih. Nejdříve popisuje podporované formáty knih, poté následuje představení existujících aplikací na tvorbu a správu elektronických knih, poté následuje popis aplikace pro převod knih naprogramované v jazyce C# s použitím grafického rozhraní Windows Presentation Foundation a uživatelského rozhraní Ribbon, je představeno rozhraní pro archivaci knih v elektronické knihovně a rozhraní pro editaci knih, které jsou součástí aplikace a práce s podporovanými formáty. Závěrem je aplikace porovnána s dalšími aplikacemi podobné funkčnosti.

Klíčová slova

elektronické knihy, C#, Visual Studio, WPF, EPUB, MOBI, PDB, PDF, HTML

Abstract

This master thesis describes the creation of an application for electronic book reader format converting. Firstly, this thesis describes ebook formats supported by the application, then it describes the process of creating the application for format converting using the programming language C#, graphical interface Windows Presentation Foundation and user interface Ribbon and it describes the parts of the application, an electronic book library used for book archivation, the book editor, and the process of reading and creating supported ebooks, in detail. Lastly, the application is compared to other applications with similar functionalities.

Key words

ebooks, C#, Visual Studio, WPF, EPUB, MOBI, PDB, PDF HTML

Poděkování

Rád bych poděkoval mému vedoucímu doc. Ing. Josefu Chaloupkovi Ph.D. za jeho podněty udávající další směřování a vedení při práci.

Obsah

Seznam zkratk	11
1 Úvod	12
2 Podporované formáty	13
2.1 EPUB	13
2.2 MOBI	16
2.2.1 Hlavičky	17
2.3 PDB	18
2.4 PDF	19
2.5 HTML	20
3 Existující aplikace na správu elektronických knih	22
3.1 Calibre	22
3.2 Adobe Digital Editions	24
3.3 Alfa eBook Manager	26
3.4 Reedsy Book Editor	28
3.5 Vellum	29
4 Aplikace pro převod formátů	31
4.1 Ovládací prvky aplikace	32
4.2 Přidávání souborů knih do knihovny	34
4.3 Elektronická knihovna	35
4.3.1 Recenze knih	37
4.4 Editor knih	38
4.4.1 Práce s kapitolami	40
4.5 Vícejazyčná podpora	41
4.6 Náповěda	42
5 Popis práce s formáty	43
5.1 HTML	43
5.1.1 Čtení	43
5.1.2 Vytváření	43
5.2 EPUB	44

5.2.1	Čtení	44
5.2.2	Vytváření	44
5.3	MOBI	45
5.3.1	Čtení	45
5.3.2	Vytváření	46
5.4	PDB	47
5.4.1	Čtení	47
5.5	PDF	47
5.5.1	Čtení	47
5.5.2	Vytváření	48
6	Porovnání s ostatními aplikacemi pro správu elektronických knih	49
7	Závěr	50
A	Obsah přiloženého DVD	54

Seznam obrázků

2.1	Struktura formátu EPUB	14
2.2	Ukázka vykreslování HTML formátu	21
3.1	Obrazovka programu Calibre	22
3.2	Porovnávání souborů v aplikaci Calibre	23
3.3	Obrazovka programu Adobe Digital Editions	24
3.4	Obrazovka programu Alfa eBook Manager	26
3.5	Editace metadat v programu Alfa eBook Manager	27
3.6	Obrazovka programu Reedsy Book Editor	28
3.7	Obrazovka programu Vellum	29
4.1	Kódová mapa aplikace	31
4.2	Hlavní okno aplikace	32
4.3	Menu aplikace	33
4.4	Okno pro zobrazení a editaci metadat	33
4.5	Okno s knihovnou elektronických knih	36
4.6	Porovnání oken „Otevřít v“ v systémech Windows XP, 7, 10	37
4.7	Okno editoru	39
4.8	Okno s tvorbou kapitol	40
4.9	Ukázka jazykových souborů	41
4.10	Ukázka nápovědy	42

Seznam zkratek

C#	C Sharp, objektový programovací jazyk
WPF	Windows Presentation Foundation, grafické rozhraní
EPUB	Electronic Publication, formát pro elektronické knihy
MOBI	Mobipocket, formát pro elektronické knihy
PDB	PalmDoc, formát pro elektronické knihy
PDF	Portable Document Format, souborový formát pro ukládání dokumentů
HTML	HyperText Markup Language, značkovací jazyk pro tvorbu webových stránek
.NET	.NET Framework, platforma pro tvorbu aplikací ve Windows
API	Application Programming Interface, rozhraní pro programování aplikací
GDI	Graphics Device Interface , rozhraní Windows pro reprezentaci grafických objektů
LINQ	Language Integrated Query, jazyk pro přístup k datům
XML	eXtensible Markup Language, rozšiřitelný značkovací jazyk
XAML	eXtensible Application Markup Language, značkovací jazyk využívaný k popisu grafického rozhraní
CSV	comma-separated values, souborový formát určený pro výměnu tabulkových dat

1 Úvod

Na trhu se vyskytuje spousta rozličných zařízení od rozličných výrobců, které spolu navzájem nemusí být kompatibilní. Ačkoliv situace se poslední dobou zlepšuje a převažují standardizované formáty, pořád existuje spousta starších knih, které na novějších čtečkách nemusí jít přečíst. Navíc i dnes existují výrobci čteček (ke kterým patří například Amazon a jeho Kindle, který tvoří nezanedbatelnou část trhu), jejichž zařízení nepodporují jiné formáty než své proprietární. V tu chvíli má uživatel dvě možnosti – buďto svou sbírku knih vyhodit nebo najít nějaký způsob, jak ji zkonvertovat do moderních formátů.

Existují různé způsoby, jak toho docílit. Problém je v jejich roztříštěnosti – mohou podporovat export nebo import v jediném formátu, případně mohou fungovat jen pro určitý typ čteček. Ne všechny také mohou být uživatelsky zcela přívětivé – mohou mít podobu skriptů, které je potřeba zkompileovat, mohou od uživatelů vyžadovat instalaci externích knihoven, nemusí mít dostupnou dokumentaci či manuál a dále.

Po zhodnocení dnešní době nejrozšířenějších formátů, které by taková aplikace měla podporovat bylo cílem vytvořit aplikaci v prostředí .NET v grafickém rozhraní Windows Presentation Foundation s uživatelským rozhraním Ribbon, která se těmito problémy vyvaruje a bude používat univerzální postup na čtení a vytváření elektronických knih.

2 Podporované formáty

Aplikace podporuje čtyři z nejpoužívanějších formátů elektronických čteček knih – EPUB, MOBI, PDB, PDF a formát HTML, sloužící jako základní formát pro vytváření knih v ostatních formátech. Tyto formáty byly vybrány pro jejich rozšířenost a univerzálnost, jelikož jejich kombinací jsou pokryty téměř všechny existující typy elektronických čteček.

2.1 EPUB

Formát EPUB (někdy stylizováno jako *ePub*, zkráceně *electronic publication* – elektronická publikace) je standardizovaný formát definovaný asociací International Digital Publishing Forum. Jedná se o souborový formát sloužící k distribuci digitálních publikací a dokumentů, založený na standardech W3C. Definuje způsob distribuce, zabalení a enkódování strukturovaného webového obsahu, mezi který patří XHTML soubory, CSS styly, SVG grafika, obrázky v dalších formátech a jiné zdroje, ve formě jediného souboru. Jeho výhodou je, že tvůrcům a vydavatelům umožňuje jednoduše vytvářet a distribuovat digitální knihy a další publikace a zákazníkům umožňuje jednoduše používat jimi koupené knihy, bez ohledu na použitý software či hardware.

Oficiálním standardem IDPF se stal v září 2007, kdy nahradil formát Open eBook (také OPF). Poslední verzí EPUB je verze 3.1, která byla standardem přijata 5. ledna 2017 a přinesla do formátu novou funkcionalitu, ke které patří nové textové formáty, možnost využívat vzdálené zdroje a rozšířená podpora HTML a CSS. EPUB je nejrozšířenější neproprietární formát elektronických knih založený na formátu XML a je podporovaný největším množstvím elektronických čteček [6].

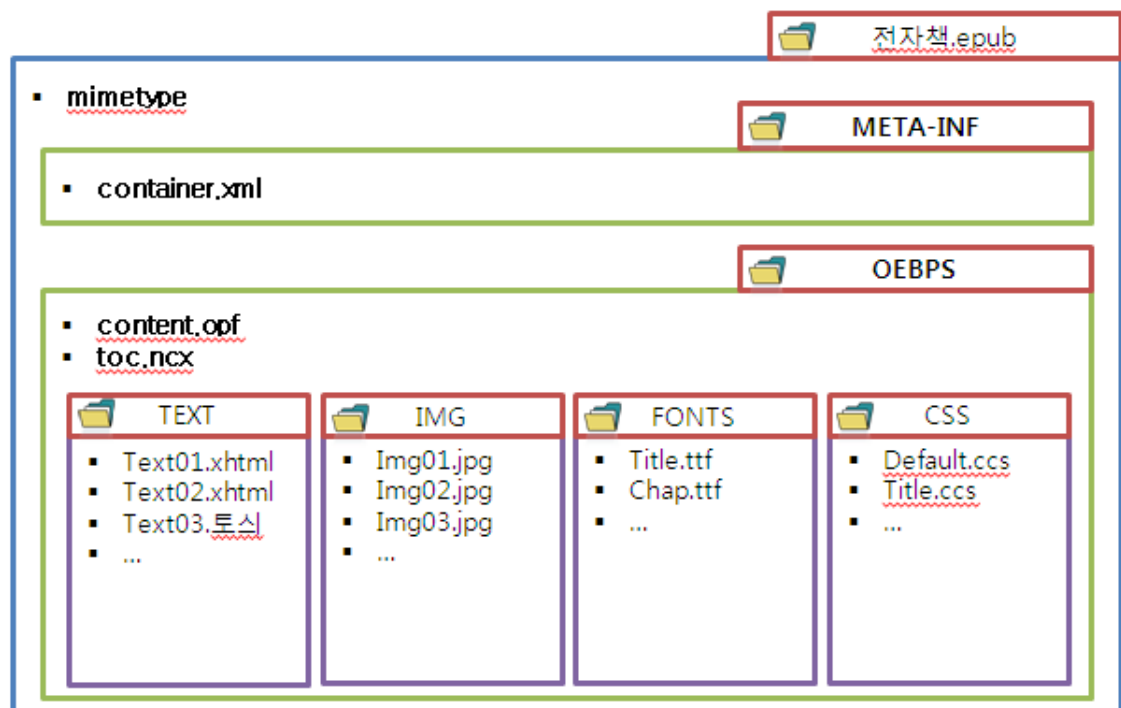
Každá EPUB publikace obsahuje minimálně jedno tzv. *Rendition* (vykreslení obsahu), každé reprezentované svým *Package Document* (dokumentací balení). Package Document specifikuje všechny zdroje, které jsou potřeba pro vykreslení obsahu a také definuje pořadí, ve kterém mají být jednotlivé soubory čteny a doplňuje vykreslení o metadata. *Package Document* má na rozdíl od typické webové stránky několik výhod – webová stránka kupříkladu odkazuje na použité zdroje (jako obrázky) přímo v textu, což je sice jednoduché a flexibilní, ale komplikuje to další práci s nimi. Navíc neexistuje žádný standartizovaný způsob, jak spojit několik webových stránek do jednoho lineárního celku, což je zajištěno právě elementem *spine* formátu

EPUB, který poskytuje externí vazbu pomáhající při navigaci kolekcí dokumentů. Package Document rovněž definuje standartizovaný způsob, jak svázat metadata s kolekcí webových stránek.

Každá EPUB publikace musí obsahovat minimálně následující soubory [obrázek 2.1]:

- Soubor `mimetype`
- Soubor `container.xml` ve složce META-INF
- Soubor s příponou `.opf` (obvykle ve složce OEBPS) obsahující seznam ostatních souborů v archivu a metadata
- Soubor `toc.ncx` ve složce OEBPS

Soubor `mimetype` je nekomprimovaný ASCII textový soubor, který obsahuje pouze řádek s textem `application/epub+zip`, sloužící jako identifikátor formátu souboru. V archivu vždy musí být na prvním místě a v kořenovém adresáři.



Obrázek 2.1: Struktura formátu EPUB

Soubor `container.xml` slouží jako odkaz na kořenový soubor obsahující seznam všech ostatních souborů.

Soubor s příponou `.opf` odkazovaný souborem `container.xml` je XML soubor, který je rozdělený na tři části. V první části zvané *metadata* jsou uvedeny informace jako název knihy, jméno autora, vydavatel, jazyk, datum vydání nebo unikátní identifikátor (jako například ISBN kód). Těchto metadata tagů existuje nespočet, ale povinné jsou pouze čtyři: název knihy, jméno autora (pro zobrazení v ovládacích rozhraních elektronických čteček), identifikátor a jazyk knihy. Druhou částí je *manifest*, který v sobě má seznam všech souborů v EPUB (včetně obrázků) a přiřazuje jim id, které bude v další části použito pro seřazení jednotlivých HTML souborů podle pořadí. Třetí částí je *spine*, řadí jednotlivé HTML soubory do správného pořadí pro účely zobrazení v elektronické čtečce. Ve *spine* nemusí být uvedeny všechny HTML soubory, ale pouze ty, které tvoří obsah knihy.

Soubor `toc.ncx` tvoří podklady pro seznam kapitol, který je generován přímo čtečkou generuje a výsledná implementace se liší čtečku od čtečky. Obsahuje tagy `navPoint` ve formátu, který je uveden na následujícím odstavci. Každý `navPoint` musí obsahovat unikátní id, pořadí, ve kterém bude jeho soubor zobrazen, popis a zdroj. V souboru `toc.ncx` jsou tak duplikovány informace uvedené v souboru `.opf`, ale i tak je pro zobrazení seznamu kapitol vyžadován.

Kromě těchto vyžadovaných souborů EPUB archiv zpravidla obsahuje HTML/XHTML soubory se samotným obsahem knihy (které mohou či nemusí být rozděleny do kapitol, pro rychlost načítání se preferuje rozdělovat) a obrázky ve formátech GIF, JPG, PNG a SVG.

2.2 MOBI

Proprietární formát používaný čtečkami Amazon Kindle. Původně se jednalo o rozšíření PalmDOC (PDB) formátu (spravovaného společností MobiPocket, kterou koupil Amazon) o podporu HTML tagů. Tento formát MOBI souborů je dodnes používaný, ale časem byl formát rozšířený o podporu nového proprietárního kompresního algoritmu, zvaného *Huff/cdic*. Ostatní soubory používají kompresi *LZ77*, kterou čte i program, který je předmětem práce. MOBI soubory se mohou vyskytovat i s příponou AZW, které se liší tím, že jsou zakódované DRM metodou, jsou tedy z principu legálně nečitelné [7].

MOBI formát může pojmout komplexní obsah, podporuje indexaci, pokročilé navigační prvky a vysokou míru komprese. Pokud pro to existuje podpora v uživatelských zařízeních, mohou si uživatelé přidávat k MOBI publikacím vlastní stránky, kresby, anotace, záložky a poznámky. MOBI publikace mohou také obsahovat vlastní slovníky.

MOBI formát je určen především pro použití na menších obrazovkách, jelikož vložené obrázky mají omezenou velikost (64 kilobajtů) a mohou být pouze ve formátu GIF, což je dělá nevhodnými pro větší tablety. Mezi další limitace formátu patří:

- Bloky textu nemůžou mít vpravo větší než výchozí vzdálenost od okraje.
- Text nemůže obtékat obrázky vyšší než jeden řádek textu.
- Velikost obrázků se nemění se změnou velikosti fontu.
- Velké množství měr nepodporuje nastavení velikosti v jednotce em (česky čtverčík).
- Jednotlivé části textu není možno zobrazovat v neproporcionálním fontu.
- Zobrazení tabulek se liší skrz různé čtečky liší, hlavně u tabulek větších než jedna obrazovka.
- Není možno vkládat tabulky do tabulek.

Jelikož formát MOBI nemá oficiální dokumentaci, není funkce všech bajtů ve všech hlavičkách zcela zřejmá a proto některé údaje ze zdrojů, ze kterých bylo čerpáno, nemusí být přesné, na extrahování textu ze souboru ale postačují. Je ale jisté, že MOBI soubory s kompresí *LZ77* mají pevně danou strukturu [7]:

- Název knihy a část jména autora (co se vejde do 31 bajtů)
- PalmDoc hlavička, obsahující údaje o nekomprimované délce knihy, počtu bloku textů a kódování

- MOBI hlavička
- EXTH hlavička, obsahující metadata
- Komprimovaný text
- Komprimované obrázky

Mezi hlavičkami jsou mezery různých délek tvořené nulovými bajty. Není známo, proč jsou mezery dlouhé tak, jak jsou, jelikož mezi různými knihami nejsou konzistentní a není jasné, podle čeho se jejich délka určuje.

2.2.1 Hlavičky

Nejdůležitější hlavička obsažená v MOBI souboru je PalmDoc hlavička, která poskytuje informace potřebné k přečtení souboru. Specifika jsou popsány v následující tabulce.

offset	počet bajtů	obsah	popis
0	2	komprese	1 == nulová komprese, 2 = PalmDOC komprese, 17480 = HUFF/CDIC komprese
2	2	nepoužito	vždy nula
4	4	délka textu	nekompirovaná délka textu knihy
8	2	počet záznamů	počet PDB záznamů, ze kterých se kniha skládá
10	2	délka záznamu	maximální délka jednoho PDB záznamu (vždy 4096)
12	4	poloha kurzoru	pozice čtenáře v podobě offsetu nekomprimované délky

Tabulka 2.1: Hlavička PalmDoc

EXTH hlavička obsahuje metadata, kterými mohou být například údaje o autorovi, vydavateli či datu vydání. Tyto metadata jsou uchovávány v EXTH záznamech, obsažených v EXTH hlavičce.

počet bajtů	obsah	popis
4	identifikátor	písmena E X T H
4	délka	délka EXTH hlavičky (včetně identifikátoru, bez výplně)
4	počet záznamů	počet EXTH záznamů, zbytek EXTH hlavičky sestává z EXTH záznamů
	začátek EXTH záznamu	
4	typ záznamu	číslo identifikující typ EXTH záznamu
4	délka záznamu	= L, do délky se počítají i bajty s údaji o typu a délce
L-8	data záznamu	
p	výplň	prázdné bajty doplňující délku EXTH hlavičky na násobek čtyř

Tabulka 2.2: Hlavička MOBI

2.3 PDB

PDB (také PalmDOC) patří mezi starší formáty pro elektronické knihy. Původně vznikl jako formát pro komprimaci textových dokumentů na kapesních počítačích značky Palm od firmy Aportis, ale časem se rozšířil i na ostatní zařízení a stal se velmi populárním formátem pro šíření elektronických knih. Z dnešního pohledu je poněkud primitivní a chybí mu podpora jistých důležitých funkcí, ke kterým patří například kompletní absence jakéhokoliv formátování, ukládání metadat nebo absence podpory DRM. I přes to se i dnes dají najít zařízení, která PDB knihy podporují, především proto, že uživatelé dodnes mohou mít archivy knih v PDB z minulosti. Nástupcem formátu PDB se stal již zmíněný formát MOBI, který rozšířil PDB o velké množství funkcionality [8].

PalmDOC soubory jsou vychozí textový formát pro zařízení Palm. Jedná se o prosté ASCII soubory, značící konec odstavců znakem pro konec řádku. Pro ušetření místa v RAM RLE algoritmus zmenší výsledný soubor o 40%. Použitá kompresní metoda není vyvážená a dekomprese textu je mnohem rychlejší, než jeho komprese. Proto je vhodnější PalmDOC soubory vytvářet na počítačích a synchronizovat je se čtečkou.

PalmDOC soubory přizpůsobují zobrazení textu displeji, na kterém jsou zobrazeny a nepodporují mnoho možností pro formátování. Některé čtečky PalmDOC souborů obsahují podporu pro záložky, které mohou fungovat jako seznam kapitol a jsou ukládány na konec souboru.

Na rozdíl od MOBI formátu PDB formát obsahuje pouze PalmDoc hlavičku (obsahující údaje o nekomprimované délce knihy, počtu bloku textů a kódování) a text.

2.4 PDF

Portable Document Format (PDF) je souborový formát určený na spolehlivou prezentaci a výměnu dokumentů, nezávisle na použitém softwaru, hardwaru či operačním systému. Původně byl navržen firmou Adobe a později se stal otevřeným standardem udržovaným organizací International Organization for Standardization (ISO). PDF dokumenty mohou dnes obsahovat širokou škálu obsahu kromě textu a grafických prvků, například interaktivní prvky jako anotace, formuláře, vrstvy, multimediální soubory jako video a trojrozměrné objekty s použitím formátů U3D nebo PRC. Specifikace PDF také poskytuje podporu pro šifrování, digitální podpisy, příkládání souborů a metadata [9].

PDF soubory kombinují tři technologie [10]:

- Skript v odnoži jazyku PostScript, generující dokument.
- Systém pro vkládání/nahrazení fontů, zajišťující správné zobrazení dokumentů.
- Strukturovaný úložný systém, který tyto prvky a další obsah svazuje do jednoho souboru a podle potřeby aplikuje kompresi dat.

PostScript je interpretovaný programovací jazyk, určený ke grafickému popisu tisknutelných dokumentů, jehož výstupem je obraz. Má podporu grafických prvků a standardních prvků programovacích jazyků, jako jsou cykly. PDF formát je z velké části založen na PostScriptu, je ale zjednodušen a zaměřuje se především na příkazy určené pro zobrazení grafických prvků.

PDF soubor je 7-bitový ASCII soubor, ale obsahuje i prvky binárních typů. Začíná hlavičkou obsahující údaje o použité verzi formátu, jako například PDF-1.7. PDF formát je poddruhem COS ("Carousel" Object Structure) formátu. Stromová struktura COS sestává z primárně z osmi typů objektů:

- Booleanové hodnoty
- Čísla

- Řetězce, oddělené závorkami ((...)), mohou obsahovat 8-bitové znaky
- Jména, začínající lomítkem (/)
- Pole, oddělené hranatými závorkami ([...])
- Slovníky, kolekce dvojic objektů klíč/hodnota, oddělené dvojitými špičatými závorkami («...»)
- Datové proudy, které mohou být komprimované a binární
- Objekt null

Kód může také obsahovat komentáře, začínající znakem procenta (%). Komentáře mohou obsahovat 8-bitové znaky.

PDF popisuje stránku pomocí systému kartézských souřadnic, nezávislém na zvoleném prohlížeči či zařízení. Popis stránky PDF může používat matice pro rotaci, zmenšování/zvětšování a zkosení grafických prvků. Jedním z klíčovým konceptů PDF je grafický stav, což je kolekce grafických parametrů, které mohou být upravovány, ukládány, a obnovovány popisem stránky. PDF má (od verze 1.6) 24 proměnných grafického stavu.

PDF soubory mohou obsahovat dva typy metadat. Prvním je Document Information Dictionary, sada objektů klíč/hodnota, kterými mohou být informace o autorovi, názvu dokumentu nebo data změny a vytvoření dokumentu. Tyto objekty jsou ukládány do nepovinného zápatí souboru. Tato metoda byla počínaje PDF 2.0 zrušena [11].

PDF verze 1.4 přidalo podporu datových proudů metadat, které používají Extensible Metadata Platform (XMP) pro přidání metadat odpovídající XML standardům, používaných v ostatních datových formátech. To dovoluje přidat metadata k jakémukoliv datovému proudu obsaženém v dokumentu (jako například k vloženým obrázkům) i k celému dokumentu.

2.5 HTML

HTML je zkratka pro jazyk HyperText Markup Language (hypertextový značkovací jazyk), který je v dnešní době nejrozšířenějším jazykem pro tvorbu webových stránek [obrázek 2.2]. Slovo hypertextový znamená, že HTML dokument může obsahovat odkazy, pomocí kterých jsou jednotlivé stránky propojené. K vytváření a formátování dokumentů používá HTML takzvané *tagy*, které společně s textem v nich vnořených tvoří prvky HTML. Vzhled a funkčnost jako třeba výška, šířka nebo barva prvků jsou ovlivněny skrz *atributy* tagů, které slouží k nastavení těchto vlastností [4].

HTML stránky mohou obsahovat vložené skripty, napsané v skriptovacích jazycích jako JavaScript, které ovlivňují chování a obsah stránek. Mohou také obsahovat CSS styly, které definují vzhled a rozložení obsahu.

Značkový jazyk HTML se skládá z několika klíčových komponent, včetně již zmíněných tagů a jejich atributů, datových typů a odkazů. HTML tagy se většinou vyskytují v párech (jako například `<h1>` a `</h1>`), ačkoliv některé reprezentují prázdné elementy a jsou tak osamocené, jako třeba ``. První z tagů v páru se nazývá otevírací tag a druhý se nazývá uzavírací.

```
<!DOCTYPE html>
<html>
<!--
Created 16-10-2014
-->
<head>
<title>Sample</title>
</head>
<body>
<p>Sample text</p>
</body>
</html>
```

The HTML code above produces the following below

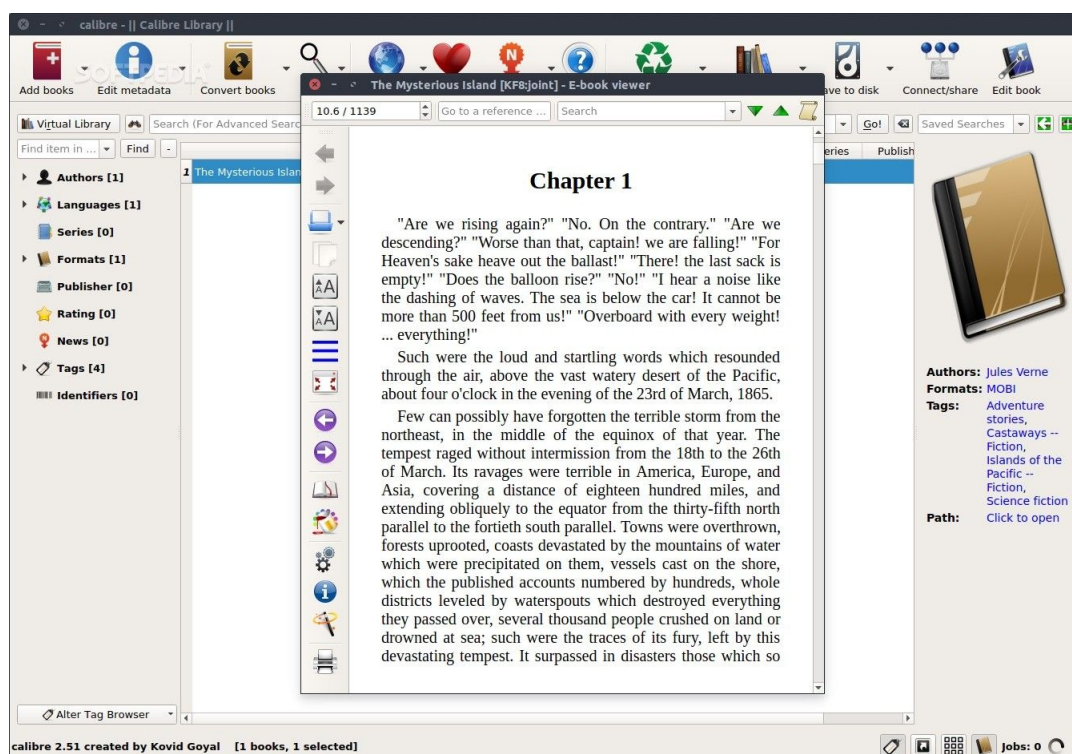


Obrázek 2.2: Ukázka vykreslování HTML formátu

Hlavička HTML dokumentu obsahuje informace, které nejsou viditelné v okně prohlížeče a patří mezi ně například titulek stránky, odkaz na CSS styly, ikony stránky a další metadata (jako jsou autor stránky a klíčová slova popisující dokument).

3 Existující aplikace na správu elektronických knih

3.1 Calibre



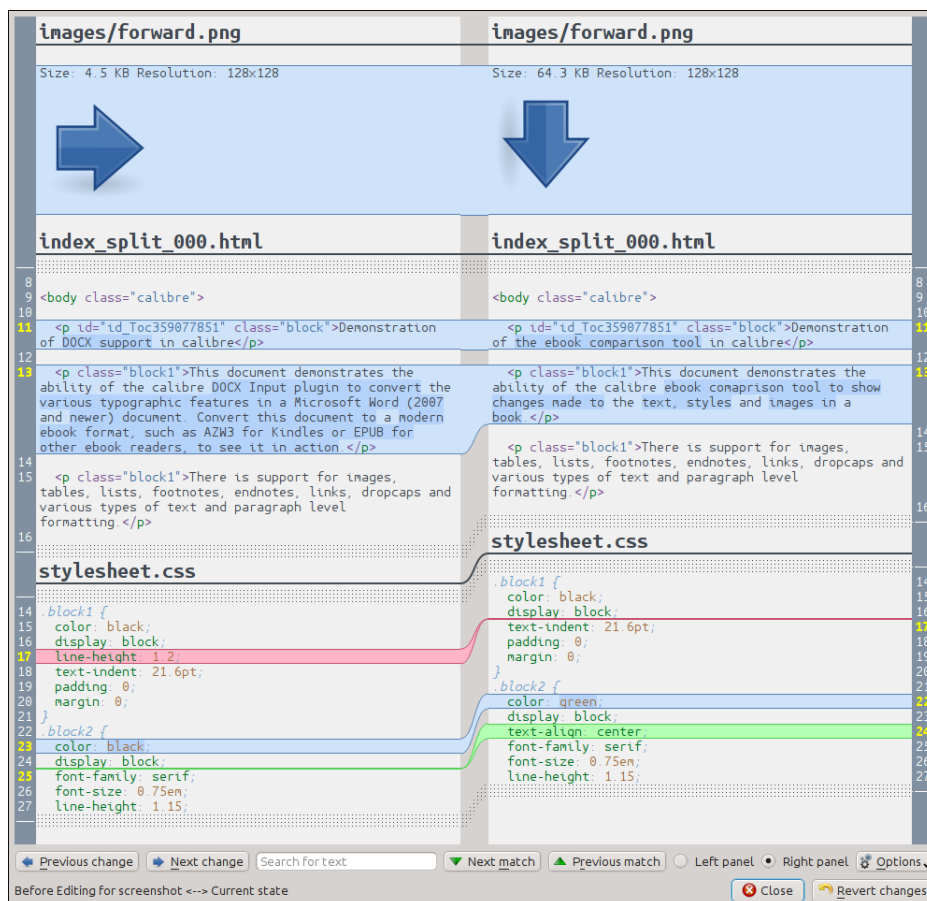
Obrázek 3.1: Obrazovka programu Calibre

Open-source aplikace Calibre [obrázek 3.1] pro Windows/Mac/Linux je jedna z nejpoužívanějších aplikací na správu elektronických knih díky svému velkému množství funkcí. S jeho pomocí je možné knihy ukládat do virtuální knihovny a tu spravovat, prohlížet svou knihovnu podle obalů, titulů, tagů, autorů nebo vydavatelů, editovat knihy, stahovat metadata knih (jako název, autora, vydavatele, ISBN číslo, obal a popis) z internetových zdrojů, vyhledávat a třídit knihy

pomocí filtrů, přesunovat elektronické knihy do dalších zařízení, podporuje všechny zásadní formáty elektronických knih a stahovat zprávy pomocí RSS kanálů [12].

Elektronické knihy mohou být do knihovny Calibre importovány jednak manuálně či pomocí synchronizace s cloudovým úložištěm, zařízením (elektronickou čtečkou, mobilním telefonem nebo tabletem). Do knihovny je možné přidávat i internetové stránky, které jsou pomocí takzvaných recipes (receptů), krátkých programů napsaných v odnoži jazyka Python, staženy z internetu a konvertovány do podoby elektronické knihy. Tyto elektronické knihy mohou být pomocí USB kabelů, integrovaného mailového serveru (který je využíván například pro export na čtečky značky Amazon Kindle) či bezdrátově nahrány do všech podporovaných zařízení [13].

Calibre rovněž nabízí integrovaný editor elektronických knih, který může upravovat knihy v EPUB a Kindle formátech. Editor funguje na bázi HTML a obsahuje aktivní preview, které se automaticky aktualizuje po uživatelských změnách. Calibre také poskytuje různé automatické nástroje sloužící k čištění a opravě elektronických knih. Calibre obsahuje podporu pro uživatelské rozšíření.



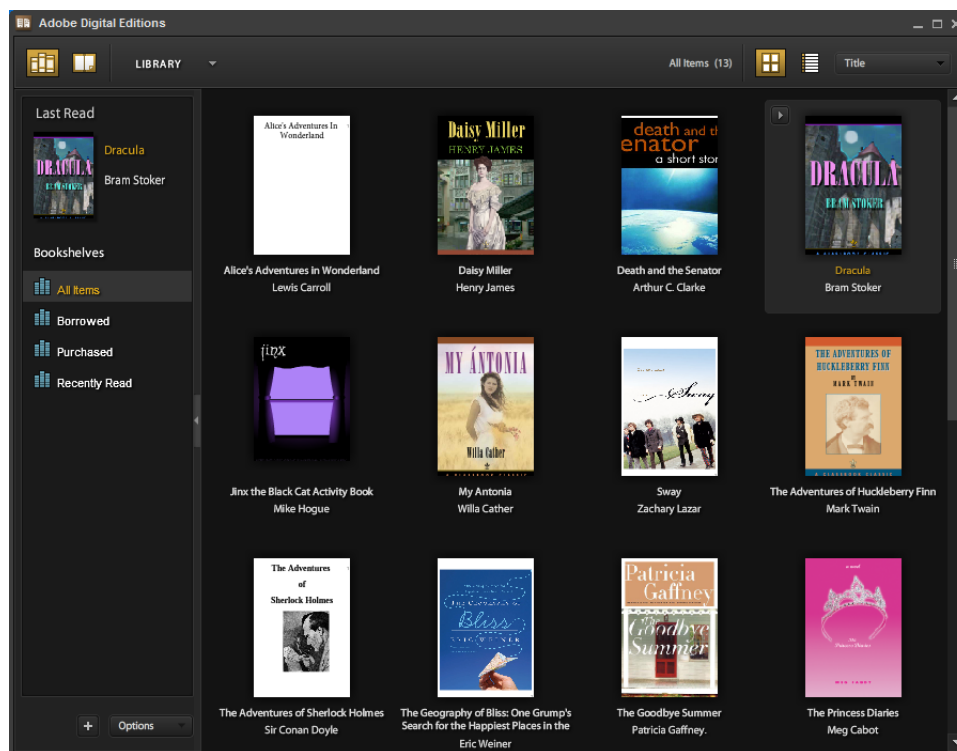
Obrázek 3.2: Porovnávání souborů v aplikaci Calibre

V programu je taktéž obsažena možnost vytvořit webový server a číst knihy

uložené v Calibre knihovně na dalších zařízeních přímo ve webovém prohlížeči, bez nutnosti instalovat jakékoliv další aplikace. Čtená kniha je stažena ze serveru a uložena do offline cache, je tedy možné pokračovat ve čtení i v případě, že zařízení je mimo internetovou síť. Jsou podporovány dva druhy serverů, lokální, který umožňuje čtení knih na zařízeních připojených k lokální síti, a veřejný, který dovoluje číst knihy na jakémkoliv zařízení připojeném k internetu. Pro funkčnost veřejného serveru musí být hostující počítač správně nastaven.

Unikátní funkcionalitou Calibre je nástroj na porovnávání elektronických knih [obrázek 3.2], který lze použít pro porovnání změn po editaci či konverzi knihy. Podporuje porovnávání knih v EPUB a AZW3 formátech. Pro rychlou orientaci ve změnách jsou v okně nástroje zobrazeny pouze relevantní řádky s okolním kontextem.

3.2 Adobe Digital Editions



Obrázek 3.3: Obrazovka programu Adobe Digital Editions

Adobe Digital Editions [obrázek 3.3] pro Windows/Mac, taktéž open-source, od firmy Adobe je rovněž určená k vytváření a organizaci knihovny elektronických knih a dalších elektronických publikací ve formátu EPUB. Na rozdíl od Calibre obsahuje

i obchod, ve kterém lze kupovat oficiální elektronické knihy a také možnost půjčovat si knihy z veřejných knihoven [14].

Stejně jako v Calibre, i s pomocí Adobe Digital Editions je možné vytvářet virtuální knihovnu elektronických knih, kterou lze organizovat podle obalů, názvů, tagů, autorů a vydavatelů knih. Knihy lze řadit a filtrovat pomocí vyhledávání, ale i pomocí vytváření specifických virtuálních políček dle libosti uživatele. Pro čtení Adobe Digital Editions nabízí režim celé obrazovky, vysokokontrastní mód pro čtení, změnu velikosti textu, uživatelské rozhraní, které je možné ovládat klávesnicí i myší možnost vkládat do knih záložky a komentáře. Adobe Digital Editions také umí knihy připravit pro tisk. Uživatelé operačních systémů Windows mohou knihy z Adobe Digital Edition kromě čtení na počítači také nahrát do podporovaných čteček.

Nástroj pro čtení knih obsahuje řadu rozšíření EPUB standardu, jako například podporu pro mapování stránek knihy podle fyzických výtisků. Odkaz na soubor s mapou stránek musí být obsažen v souboru *content.opf* a pokud není, je programem Adobe Digital Editions číslování stránek vygenerováno automaticky, rozdělením do stránek po 1024 znacích.

Mezi další podporovaná rozšíření EPUB formátu patří například rozšíření uspořádání stránek, kolektivně nazvány šablona XML stránky (*XML page template*, XPGT). Patří mezi ně vzorové stránky, pomocí kterých je možno na stránky přidat záhlaví, zápatí, postranní lišty a zobrazit text ve vícesloupcovém rozložení a dynamický výběr vzorových stránek a dynamické stylování podle prostředí (kterým mohou být rozměry plochy s textem, výchozí velikost fontu, rozlišení použitého zařízení a další). Tyto šablony jsou uloženy ve speciálním souboru se styly, na který je z XHTML souboru odkázáno jako na kterýkoliv jiný soubor kaskádových stylů.

Pro mobilní zařízení existuje speciální verze programu Adobe Digital Editions, zvaná Adobe Reader Mobile SDK, podporována více než 50 mobilními zařízeními a čtečkami od firmy Sony a dalších. Adobe Reader Mobile SDK ovšem nepodporuje všechny vlastnosti klasické verze programu pro PC/MAC, jako například zobrazení SWF souborů umístěných v knihách. Predominantně je tato aplikace používána čtečkami, podporující DRM od distributora Overdrive, použitým pro výpůjčování knih z veřejných knihoven.

3.3 Alfa eBook Manager

Aplikace Alfa eBook Manager [obrázek 3.4] pro Windows je další z aplikací určených k vytváření knihovny elektronických knih. Jedním z jeho unikátních prvků je možnost evidovat i papírové knihy, které si uživatel musí do virtuální knihovny ručně vložit [15].

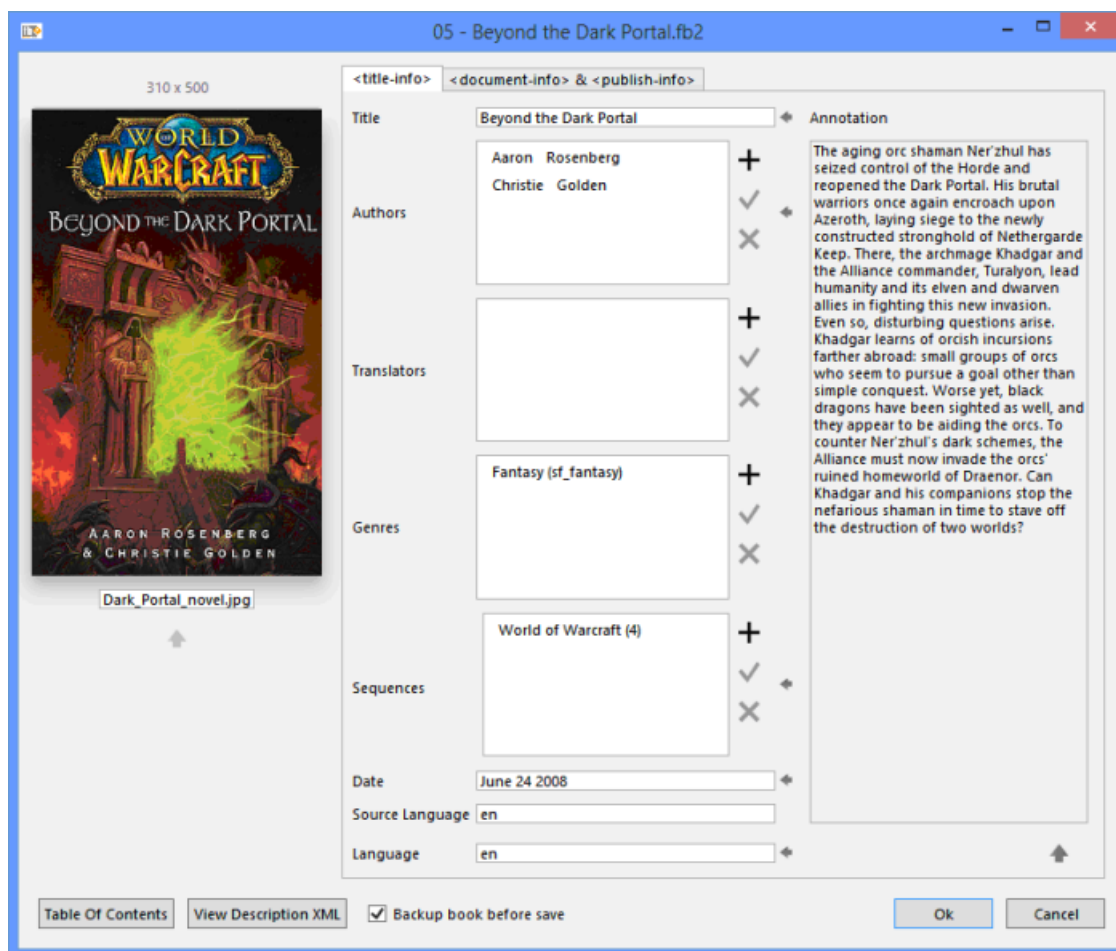


Obrázek 3.4: Obrazovka programu Alfa eBook Manager

Kromě funkcionality pro údržbu knihovny elektronických knih je v programu Alfa Ebooks Manager obsažen i vestavěný nástroj pro čtení knih, podporující jednak textové knihy ve formátech PDF, EPUB, MOBI, FB2, CHM, HTML, TXT, ale i obrázkové archivy CBR a CBZ, určené především pro čtení komiksů. Obsahuje také audio přehrávač pro formáty MP3 a M4B pro poslech audioknih.

Data o knihách obsažených v knihovně jsou uloženy v databázovém souboru typu SDF, který lze otevřít a upravovat pomocí dalších databázových programů, jako je například CompactView. Tyto soubory lze zálohovat a přenášet mezi počítači. Každá kniha ve virtuální knihovně je reprezentována svou kartou, která obsahuje odkazy na knihy uložené na pevném disku počítače, metadata týkající se knihy a odkazy na obálku knihy. V kartě knihy jsou také uloženy náhledy obálek v podobě binárního řetězce, ne jako separátní soubory na disku.

Metadata příslušná knihám [obrázek 3.5] je možno stahovat z různých knižních databází a elektronických obchodů, ke kterým patří Amazon, Google Books nebo



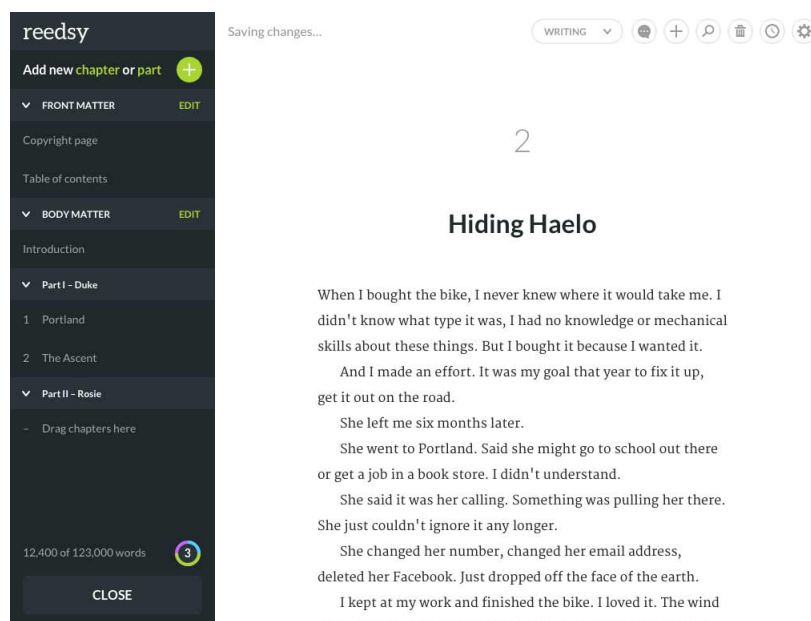
Obrázek 3.5: Editace metadat v programu Alfa eBook Manager

Barnes & Noble. Data z nich lze stahovat separátně či jejich kombinací. Pro každou z těchto stránek existuje separátní plugin, který se stará o parsování dat a pokud daná stránka změní strukturu, přestane plugin fungovat a je ho třeba aktualizovat.

Program také dovoluje importovat svou kolekci knih z jiných databází, kterými jsou Calibre, Bookcat, Google Library a strukturované CSV soubory a exportovat ji do formátů CSV, HTML a XML.

Na rozdíl od konkurence je Alfa eBook Manager placený software, dostupný ve třech verzích – základní verze nabízí pouze omezené možnosti pro uspořádávání knihovny a čtení knih, jako je základní vyhledávání a filtrování, přidávání tagů ke knihám a parsování metadat z internetových databází. Druhá, pokročilá verze nabízí navíc možnost editovat metadata knih ručně, konvertovat elektronické knihy do jiných formátů a vytvářet složitější adresářové struktury v rámci virtuální knihovny. Třetí a nejdražší je webová verze, která dovoluje uživatelům hostovat na svých počítačích webový server a číst tak své knihy z jiných zařízení [16].

3.4 Reedsy Book Editor



Obrázek 3.6: Obrazovka programu Reedsy Book Editor

Jednoduchá webová aplikace, zaměřená na snadné vytváření validních elektronických knih ve formátu EPUB [obrázek 3.6]. Editor funguje na principu WYSIWYG, jeho uživatelé tedy nemusí pro vytváření ovládat HTML jazyk. Editor dovoluje vytvářet kapitoly a podkapitoly a obsahuje několik šablon na automatické formátování textu, ale dovoluje uživateli i formátovat text ručně pomocí klasických stylů jako jsou tučný text, citace, seznamy s odrážkami a další. Editor dovoluje i vkládání obrázků a poznámek pod čarou. Části knihy jako autorská tiráž a seznam kapitol jsou generovány automaticky. Výsledné knihy mohou být vygenerovány ve formátech EPUB (pro publikaci v podobě elektronické knihy) či PDF (pro tisk) a jsou připraveny pro komerční vydání [17].

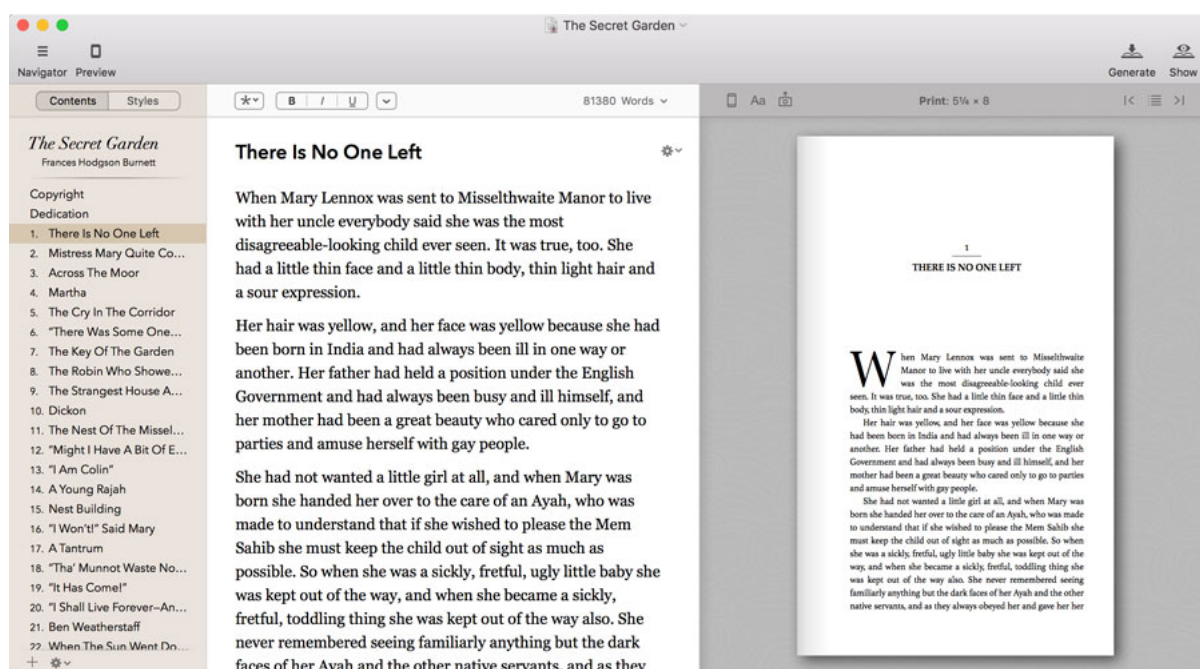
Reedsy Book Editor počítá s tím, že na tvorbě jedné knihy může pracovat více lidí, čemuž napomáhá implementací sledování změn a systému poznámek, pomocí kterých se spisovatelé mohou jednoduše domluvit se svými editory a naopak. Pro nezávislé tvůrce nabízí i možnost najmout si profesionální editory skrz portál Reedsy Marketplace.

V rámci jednoduchosti ale program nepodporuje mnoho uživatelských nastavení, což může představovat problém pro zkušenější a náročnější uživatele. Dostupné šablony včetně fontů byly navrženy profesionálními sazeči na základě uznávaných standardů, nemohou ale zákonitě pokrýt potřeby všech autorů.

Aplikaci také není možno používat na zařízeních nepřipojených k internetu, je-

likoř je pouze webov a pro její pouřzování musí bt uřivatel pihlšen ke svmu Reedsy ctu. Nepodporuje tak export do jinch formt neř je EPUB, dle slov autor aplikace je ale vsledn EPUB soubor jednoduře zkonvertovateln na formt MOBI pomocí nstroj firmy Amazon.

3.5 Vellum



Obrzek 3.7: Obrazovka programu Vellum

Aplikace pro operan sstm Mac, slouřící na vytvření elektronickch knih, odpovdjícím specifikcím obchod, a knih pipravench k tisku [obrazek 3.7]. Je zamřen pedevřím na snadnost pouřzování a vslednou vizuln podobu knih. Podporuje import textu z dokument Word a obsahuje jednoduch interface pro pravu knih [18].

Pi importu existujcho dokumentu se Vellum snař obsah dokumentu inteligentn parsovat. Pokud dokument obsahuje seznam kapitol, vsledn rozdlení knihy by mu mel co nejvce odpovdat, a pokud neobsahuje, orientuje se program podle formtovacch znaek, jako jsou konce strnek a odstavc. Pokud napařsovan dokument neodpovd pedstav uřivatele, mře jednoduře vytvřt, mazat a pejmenovvat vlastn kapitoly.

Pro tvorbu tiskutelnch knih Vellum obsahuje spoustu funkc, kter knihu automaticky piprav pro tisk. Patr k nim naprklad vber vhodn velikch okraj

a fontů pro zadané rozměry stránky, automatické očíslování stránek po obálce, přidávání konfigurovatelných záhlaví a zápatí na určité stránky, automatická tvorba seznamu kapitol s vygenerovanými čísly stránek nebo automatická oprava vdov a balancování řádků.

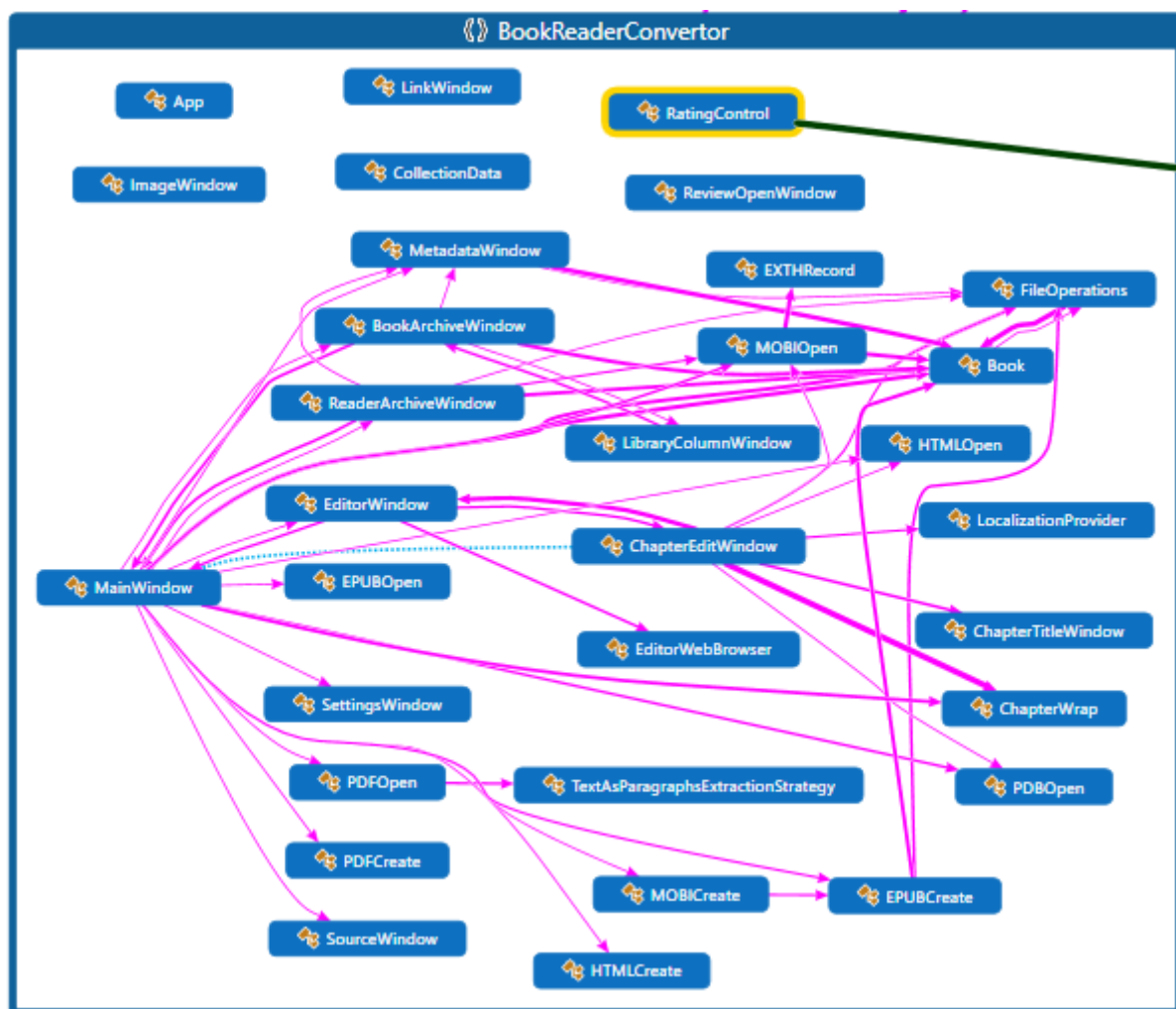
Vellum disponuje předdefinovanými styly, kterými lze upravit vzhled vytvořené knihy. Styly lze aplikovat na celou knihu či jen na jednotlivé prvky, jako jsou například nadpisy. Kromě předdefinovaných stylů si může uživatel vytvořit i vlastní.

Ve výchozím nastavení Vellum exportuje knihy připravené pro vydání na platformách firem Apple, Kobo a Amazon. Nepodporuje ale nativní export do formátu MOBI pro čtečky Amazon Kindle, ke kterému musí uživatel doinstalovat utilitu KindleGen. Pokud chce uživatel vydat knihu i na jiných platformách než třech zmíněných, je dostupný i export generického souboru EPUB.

Jednorázový export elektronické knihy (ve všech dostupných formátech) vyjde uživatele na 29,99 dolarů. Je dostupná i neomezená licence, která na tvorbu elektronických knih stojí 199,99 dolarů a na tvorbu elektronických knih a papírových knih stojí 249,99 dolarů.

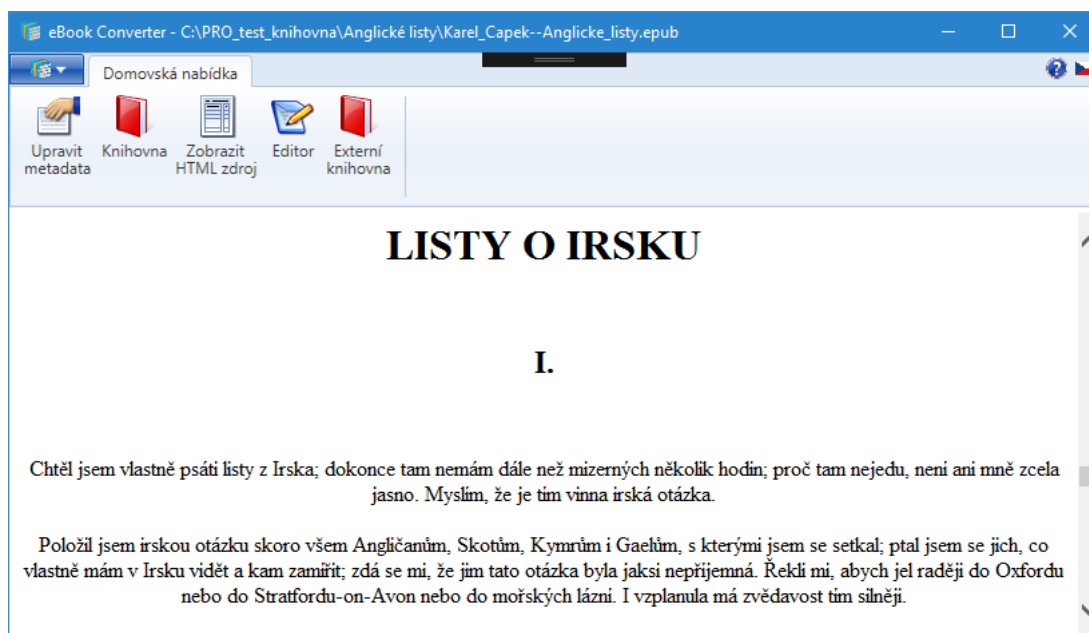
4 Aplikace pro převod formátů

V rámci diplomové práce byla vytvořena aplikace na převod elektronických knih mezi jednotlivými formáty a na jejich uchovávání ve virtuální knihovně [obrázek 4.1]. Aplikace je naprogramovaná v prostředí Visual Studio 2013 v jazyce C# s použitím Windows Presentation Foundation a ovládacích prvků uživatelského rozhraní Ribbon.



Obrázek 4.1: Kódová mapa aplikace

4.1 Ovládací prvky aplikace

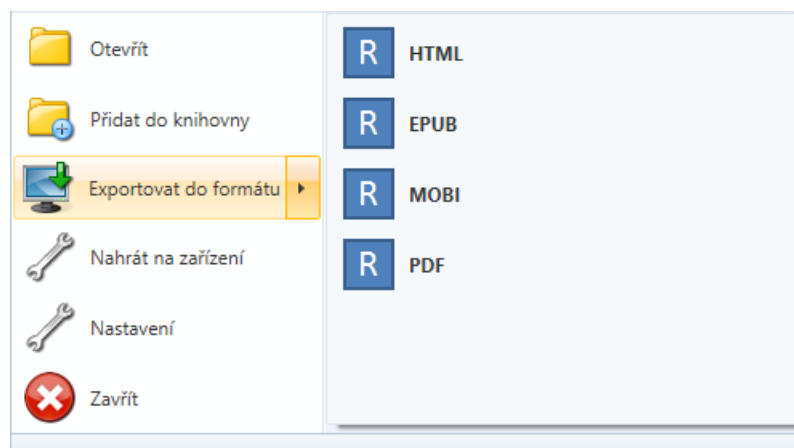


Obrázek 4.2: Hlavní okno aplikace

Hlavní okno aplikace `MainWindow` [obrázek 4.2] obsahuje dva hlavní ovládací prvky, uspořádané do mřížky *Grid*: první je panel nástrojů a hlavní nabídka *Ribbon* a druhým je *WebBrowser*, který slouží k zobrazení textu načtených knih. Po kliknutí na hlavní ovládací tlačítko v levém horním rohu se otevře hlavní ovládací nabídka *RibbonApplicationMenu*, obsahující elementy typu *RibbonApplicationMenuItem*, které slouží jako jednoduché položky menu s titulkem, ikonou a popisem funkce a elementy typu *RibbonApplicationSplitMenuItem*, které po přejetí myši otevřou oddělené menu s podpoložkami *RibbonApplicationMenu* [obrázek 4.3] [1].

První položkou v nabídce je tlačítko *Otevřít*, které kliknutím vyvolá metodu `OpenFile`. Ta uživateli otevře dialogové okno *OpenFileDialog*, ve kterém si vybere požadovanou knihu a podle její přípony (rozlišuje EPUB archivy, MOBI soubory, PDB soubory a HTML soubory s příponami `.html`, `.htm`, `.xhtml` a `.xhtml`) se zavolá metoda `Open` příslušné třídy (`EPUBOpen`, `MOBIOpen`, `PDBOpen` nebo `HTMLOpen`). Pro soubory typu MOBI a PDB se ještě volá metoda `FixMetaElement`, která kontroluje, jestli v nich nechybí element *meta* s atributem *charset* s informacemi o znakové sadě použité v souboru – pokud ano, tak ho přidá do hlavičky *head*. Získaný text se v okně zobrazí metodou `WebBrowser.NavigateToString`.

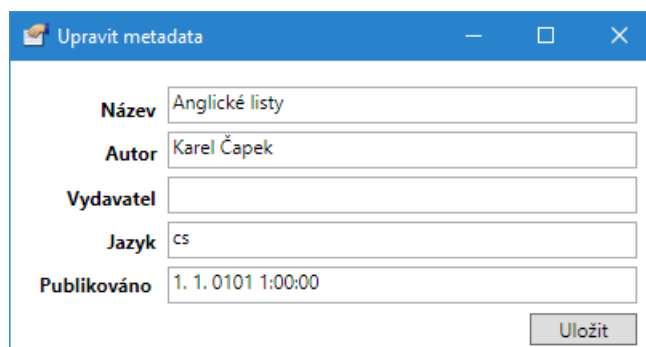
Druhá položka (*Přidat do knihovny*) je podrobněji popsána v následující sekci. Třetí položkou je *Exportovat do formátu*, element typu *RibbonApplicationSplitMenuItem*, ve kterém jsou zabalené tlačítka s názvy jednotlivých formátů. Po kliknutí



Obrázek 4.3: Menu aplikace

na jedno z nich se použije metoda `ExportFile`, která nejdříve otevře dialogové okno `SaveFileDialog`, kde si uživatel vybere, kam chce konvertovaný soubor uložit a pak se podle názvu elementu `RibbonApplicationMenuItem`, který metodu vyvolal, vyvolá příslušná metoda třídy (`EPUBCreate`, `MOBICreate`, `PDBCreate` nebo `HTMLCreate`), pomocí kterých je soubor uložen do zvolené cesty.

Poslední položkou v nabídce `RibbonApplicationMenu` je možnost `Nastavení`, která otevře nové okno `SettingsWindow` s předvolbami programu. V současné verzi obsahuje pouze jedinou možnost, volbu umístění složky obsahující knihovnu elektronických knih.



Obrázek 4.4: Okno pro zobrazení a editaci metadat

Na ovládacím prvku `Ribbon`, přítomném v hlavním okně aplikace jsou k dispozici další tři funkce, týkající se práce s vybranou elektronickou knihou. První je `Upravit metadata`, která, pokud pracujeme s knihou uloženou v knihovně, otevře separátní okno `MetadataWindow` [obrázek 4.4], ve kterém lze upravit existující informace o knize, uložené v souboru `metadata.opf` a následně přeparované do třídy `Book` ve

složce s knihou a přidat chybějící. To může sloužit pro opravu chybně definovaných metadat vydavatelem knihy nebo pro zlepšení uživatelského filtrování v knihovně.

Druhá možnost *Knihovna* slouží k otevření okna `BookArchiveWindow`. Knihovna je komplikovanější součástí programu a proto jí je věnována samostatná sekce. Třetí možnost *Zobrazit HTML zdroj* otevře okno `SourceWindow`, které, jak už titulek tlačítka napovídá, zobrazí otevřenou knihu ve formě HTML zdroje (bez možnosti editace).

4.2 Přidávání souborů knih do knihovny

Druhou položkou v nabídce *RibbonApplicationMenu* je tlačítko *Přidat do knihovny*, které pro knihu vytvoří složku v elektronické knihovně. Metoda `AddToLibrary` také otevře dialogové okno *OpenFileDialog*, ve které si uživatel vybere soubor elektronické knihy. Pokud jde o soubor ve formátu EPUB, název složky se určuje podle titulu knihy, který získáme skrz parametr `Title` otevřené knihy metodou `EpubReader.OpenBook`. Ten je regulárním výrazem zbaven znaků, které jsou v názvu souborů či složek nepovolené a vytvoříme v naší knihovně složku s vyčištěným názvem. Do té je zkopírován vybraný soubor elektronické knihy a extrahován z EPUB archivu soubor s metadaty tak, že je vytvořena instance třídy *ZipFile* a cyklem `for-each` ze seznamu *ZipEntry* nalezen soubor s příponou *.opf*, který je extrahován metodou `Extract` do nově vytvořené složky a přejmenován na *metadata.opf*. Tím jsou připraveny potřebné informace pro zobrazení v elektronické knihovně.

PDB soubory neuchovávají jiná metadata než název knihy, který je vždy uložen na začátku souboru v prvních 31 bajtech. Ten je možno přečíst metodou `ReadTitle` se vstupním parametrem `fileName`, reprezentujícím cestu k souboru, která přečte prvních 31 bajtů vybraného souboru, odstraní nulové bajty na konci a zbytek převede do textového řetězce. Ten je stejně jako u ostatních formátů zbaven nepovolených znaků a vytvořena složka s tímto názvem, do které je zkopírován vybraný soubor. Jelikož nejsou známa jiná metadata než název díla, metoda `CreateMetadataFile` vyplní ostatní vyžadované položky jako *unknown* a ostatní nechá prázdné.

Jelikož HTML soubory nemusí uchovávat ani název knihy v titulku souboru, je nejlepší složku s HTML knihou nazvat podle názvu HTML souboru. Co se týče metadat, je postup stejný jako u PDB souborů – všechny údaje kromě názvu knihy jsou neznámé.

Pokud jde o soubor ve formátu MOBI, je z něho nejdříve nutno získat název knihy stejnou metodou jako u souborů PDB, `ReadTitle` a ostatní metadata z hlavičky EXTH metodou `GetEXTHRecords`. Metoda nejdříve najde začátek hlavičky EXTH pomocí metody `FindStartOfEXTH`. 8 bajtů od začátku hlavičky je ve čtyřech bajtech uložený počet EXTH záznamů, aby algoritmus věděl, kdy přestat číst. EXTH záznamy mají následující formát [7]:

- 4 bajty – typ záznamu
- 4 bajty – délka záznamu $N + 8$
- N bajtů – záznam

Záznamy jsou uloženy do listu třídy `EXTHRecord`, který je předán metodě `ParseRecords`. Metoda najde v záznamech ty, které jsou relevantní pro účely aplikace – autora knihy (typ 100), vydavatelství (101), datum vydání (106) a jazyk (524) a vytvoří z nich instanci třídy `Book`. Nyní je možno vytvořit příslušnou složku, do které je zkopírován vybraný soubor. Získaná metadata jsou předána metodě `CreateMetadata` v parametru `book` třídy `Book`, která vytvoří XML soubor s metadaty a uloží ho do souboru `metadata.opf`.

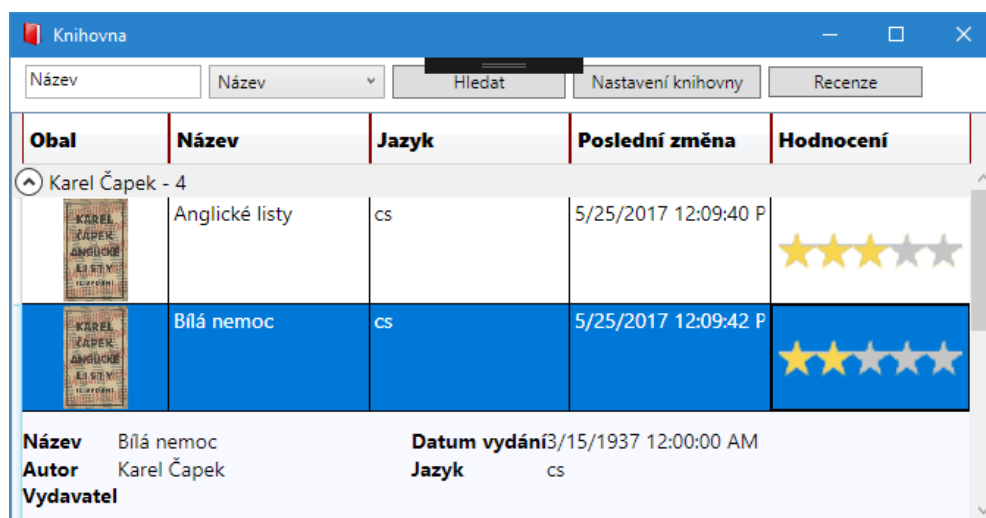
4.3 Elektronická knihovna

Okno `BookArchiveWindow` [obrázek 4.5] používá k zobrazení dostupných knih komponentu `DataGrid`. Seznam knih získává pomocí metody `GetBooks`, která projde všechny složky ve zvoleném umístění knihovny, najde soubory s metadaty a vytvoří z nich instance třídy `Book` s následujícími parametry:

- cesta ke složce s knihou
- název knihy
- autor knihy
- vydavatelství knihy
- jazyk knihy
- datum vydání knihy
- datum úpravy knihy
- velikost knihy v MB

Třída `Book` implementuje rozhraní `INotifyPropertyChanged`, které při změně parametru dovoluje oznámit klientům, že ke změně došlo a tak okamžitě zobrazit aktuální data. Toho je využito při editaci metadat, aby se kvůli úpravě jednoho řádku nemusela znovu načítat celá složka s knihami. Získané knihy jsou uloženy do seznamu typu `ObservableCollection`, který podporuje upozorňování při změně a smazání položky a obnově celého listu. Z tohoto seznamu je vytvořena kolekce typu `ListCollectionView`, která podporuje seskupování položek podle specifikovaného deskriptoru, kterým je v tomto případě jméno autora pro snadnější orientaci

v knihách. Tato kolekce je předána *DataGrid* jako *ItemsSource*, čímž je docíleno automatického zaplnění tabulky a aktualizaci při změně parametrů vložených položek kolekce [3].



Obrázek 4.5: Okno s knihovnou elektronických knih

Aby se u definovaných kategorií mohl zobrazovat popis, je třeba definovat *ContainerStyle* (styl obalu) a v něm *ControlTemplate* (šablona ovládacího prvku) s typem *GroupItem*. Ten obsahuje ovládací prvky typu *Expander* sloužící k rozbalení/zabalení kategorie a *StackPanel*, který automaticky rovná vložené popisky typu *TextBox* za sebe. Popisky mají hodnoty *Name* (jméno) a *ItemCount* (počet položek), získané z parametrů své kategorie.

Knihovna obsahuje možnost hodnotit obsažené knihy známkou od jedné do pěti hvězd. Za tímto účelem byl vytvořen uživatelský ovládací prvek *RatingControl*, podporující uživatelské ovládání pomocí myši a automatické zobrazování v tabulce pomocí data binding. Toho je docíleno pomocí třídy *DefaultBindingProperty*, která označuje výchozí proměnnou použitou při data bindingu. Ovládací prvek je složen z tabulky *Grid*, která v sobě obsahuje dva panely *WrapPanel*, do kterých jsou vkládány hvězdy. Do prvního jsou vkládány hvězdy podle počtu udělených bodů a do druhého jsou ukládány hvězdy, do kterých body uděleny nebyly.

Tlačítko *Nastavení zobrazení* slouží k skrytí/zobrazení sloupců tabulky. To z *Properties.Settings* načte hodnoty symbolizující sloupce a uživatel si pomocí zatržitek může vybrat, které chce zobrazit. Po uložení se vyvolá metoda *RefreshColumnVisibility*, která zobrazení sloupců aktualizuje.

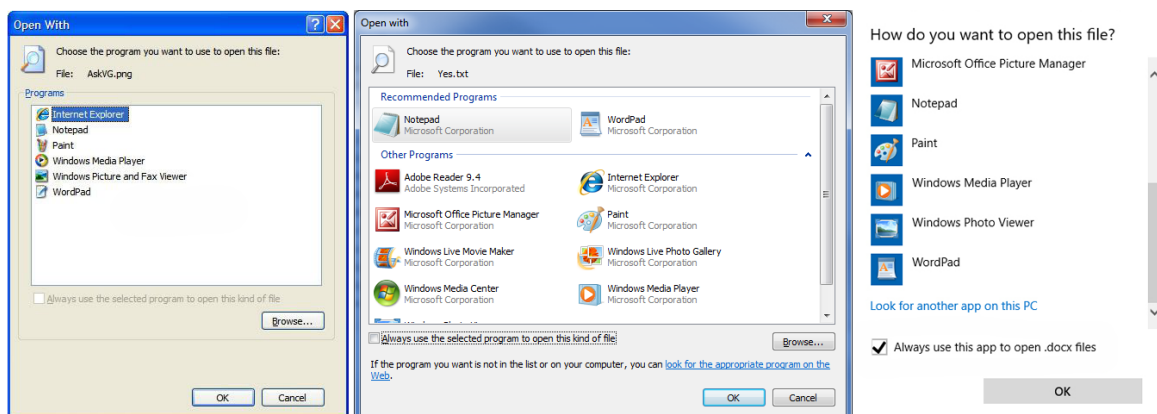
Po kliknutí na knihu se rozbálí záložka s metadaty. K tomu je použit *RowDetailsTemplate* (šablona pro detail řádku) obsahující *DockPanel* se čtyř sloupcovou mřížkou *Grid*, do které jsou metadata uspořádány. Tato metadata jsou získávána pomocí metody *FindSelectedRow*, která podle indexu řádku najde položku třídy

Book v jejich kolekci. Dvojitým kliknutím se kniha otevře v hlavním okně, kde si ji uživatel může prohlédnout a upravit metadata.[1]

V knihách můžeme i vyhledávat pomocí jednoduchého rozhraní navrchu okna, ve kterém si uživatel může vybrat parametr, podle kterého chce vyhledávat. K vyhledávání je využito toho, že kolekce *ListCollectionView* se dá filtrovat pomocí predikátových metod. Za tímto účelem je vytvořena predikátová metoda *BookFilter* vracící boolean *true* v případě, že zvolený parametr knihy v sobě obsahuje řetězec vložený uživatelem.

4.3.1 Recenze knih

Tlačítko *Recenze* slouží k otevření textového souboru, ve kterém si uživatel může uložit recenzi či poznámky k vybrané knize. Z důvodu přenositelnosti byl pro editaci recenze zvolena možnost otevření souboru v externím editoru, kterým může být například Poznámkový blok, Microsoft Word nebo LibreOffice Writer. Aby si ale uživatel mohl vybrat, ve kterém programu chce soubor editovat a neotvíral se automaticky ve výchozím programu, nastaveném v systému, bylo nutno vyřešit problém, jak programátorsky otevřít dialogové okno „Otevřít v“ [obrázek 4.6]. Způsob se liší podle uživatelského operačního systému



Obrázek 4.6: Porovnání oken „Otevřít v“ v systémech Windows XP, 7, 10

Ve Windows 7 a starších je možné toto dialogové okno vyvolat pomocí vstupního bodu shellu s názvem *SHOpenWithDialog*. Jmenný prostor Shell organizuje souborový systém a další objekty spravované shellem do jedné stromové struktury. Jednou z hlavních funkcí shellu je správa a udělování přístupu k objektům v zmíněné stromové struktuře. Pomocí vhodně nastavených parametrů tak lze shell využít k otevření požadovaného dialogového okna, včetně možnosti nastavit nový program jako výchozí [25].

Jelikož ve Windows 10 byl ale shellový dialog nahrazen novým dialogem rozhraní UWP, toto řešení není funkční. V UWP aplikacích je možno použít třídu *LauncherOptions*, nastavit její proměnnou *DisplayApplicationPicker* na *true* a otevřít soubor pomocí metody *Launcher.LaunchFileAsync*, tyto třídy a metody ale v klasických Win32 aplikacích nejsou dostupné. Nabízela se možnost přidat do projektu reference na žádané knihovny, ať už pomocí NuGet balíku *UwpDesktop* [19] či ručně – balík *UwpDesktop* ovšem v novějších instalacích Windows přestal fungovat, jelikož po každé aktualizaci mění požadované knihovny umístění na disku a balík nebyl aktualizován. Ruční přístup sice funguje, naskytl se ovšem problém, že Win32 aplikace po ručním přidání mohou využívat UWP API, ne ale všechny, jelikož část z nich vyžaduje programový manifest, který Win32 aplikacím schází [27].

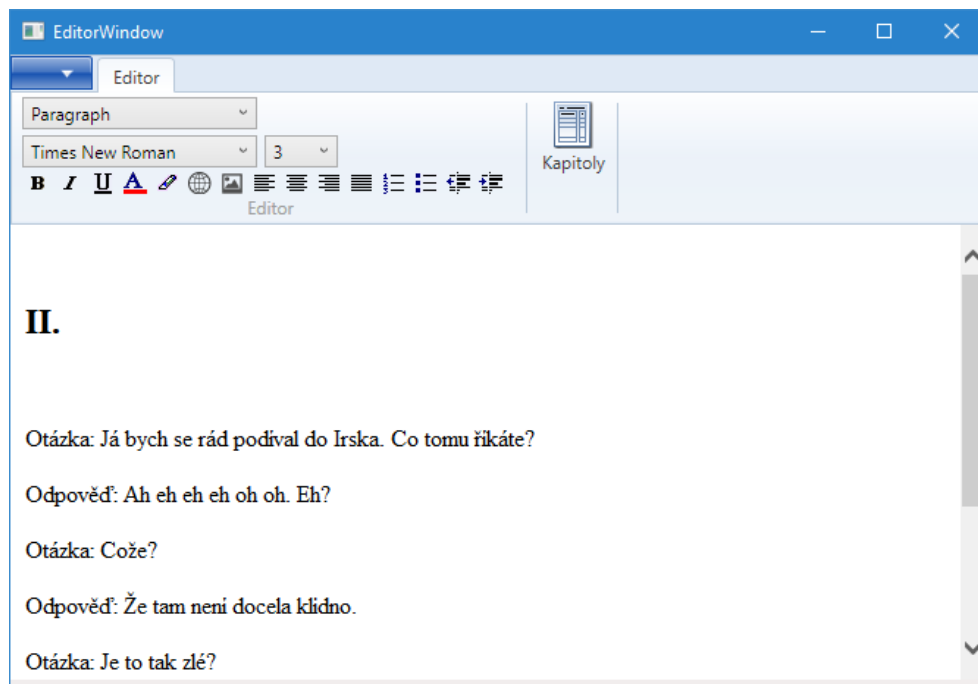
Bylo tedy nutné přistoupit k vlastnímu řešení a zprostředkovat uživateli omezenou selekci programů pomocí vlastního dialogového okna. Nejdříve je z registrů pomocí známých klíčů zjištěno, zda jsou nainstalovány hledané editory (Microsoft Word a LibreOffice Writer) a pokud ano, je vrácena jejich cesta a zpřístupní se v dialogovém okně jako jedna z možností. Pokud není nainstalován ani jeden z editorů, je k otevření použit vždy dostupný Poznámkový blok.

4.4 Editor knih

Program obsahuje editor knih [obrázek 4.7], ve kterém si uživatelé mohou upravovat obsah svých knih. Uživatelské rozhraní editoru je podobné standartům známým z editorů, jako je Microsoft Word. Editor podporuje většinu formátovacích a typografických funkcí, které jsou pro formátování elektronických knih potřeba – změna typu a velikosti fontu, zvýraznění textu tučně, kurzívou, podtržením, barvou, změna barvy pozadí, vložení odkazů a obrázků, zarovnání odstavců nalevo/napravo/nastřed, vložení seznamů s odrážkami a číslovaných seznamů a odsazení textu.

Editovatelný text se zobrazuje v komponentě třídy *EditorWebBrowser*. Klasická třída *WebBrowser* nemohla být použita z důvodu, že do její editovatelné instance není možné načíst dva dokumenty, což je v tomto editoru potřeba při přepínání mezi jednotlivými kapitolami elektronické knihy. Třída *EditorWebBrowser* má dvě proměnné, zobrazený dokument třídy *HTMLDocument* a webový prohlížeč třídy *WebBrowser*, a jednu metodu s názvem *NewWB*.

Metoda *NewWB* se volá z metody *OpenHTMLContent* třídy *EditorWindow*. Při každém zavolání je nejdříve zkontrolováno, zda existuje instance komponenty *webBrowser* a pokud ano, tak je zničena pomocí metody *Dispose* a odstraněna z kolekce elementů uživatelského rozhraní okna *EditorWindow*. Po této kontrole je vytvořena nová instance *webBrowser*, která je přidána do zmíněné kolekce a pomocí metody *NavigateToString* je v ní zobrazen text, který je předán vstupním parametrem metody. Pokud se tento vstupní parametr rovná prázdnému stringu, otevře



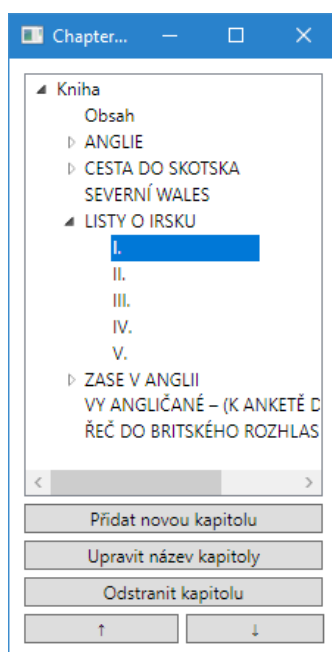
Obrázek 4.7: Okno editoru

metoda šablonu pro prázdný dokument. Aby otevřený dokument bylo možno editovat, je nutno uložit objekt *Document* komponenty *webBrowser* do nové proměnné, jejíž parametr *designMode* musí být nastaven na hodnotu "On". Takto získaná proměnná je následně předána statické třídě *Format*, která zajišťuje úpravy textu.

Třída *Format* pracuje s dokumentem pomocí metody *execCommand*, který k vybranému textu přidává zvolené vlastnosti pomocí tagů. Pro každé tlačítko na ovládacím panelu editoru je definována vlastní metoda, která volá metodu *execCommand* s příslušnými parametry podle funkce tlačítka, jako *Bold* pro tučný text či *JustifyLeft* pro zarovnání doleva. Parametry pro výběr barev jsou získány pomocí dialogového okna *ColorDialog*, ve kterém si uživatel může z palety barev vybrat požadovanou barvu. Pro vložení odkazů a obrázků jsou použity dialogová okna *LinkWindow* a *ImageWindow*, ve kterých uživatel vyplní adresu odkazu/obrázku, alternativní text v případě obrázku a text odkazu v případě odkazu.

4.4.1 Práce s kapitolami

Editor knih je schopný strukturovat text knihy do kapitol, které může libovolně přidávat, odebírat a přesouvat jejich pořadí. Při otevření okna *ChapterEditWindow* [obrázek 4.8] metoda *InitializeChapters* ověří, zda kapitoly nejsou uloženy v proměnné *chapters* hlavního okna. Pokud ne a otevřená kniha je formátu EPUB, je načtena pomocí třídy knihovny *EpubReader*, která automaticky rozdělí knihu do kapitol a podkapitol. Z nich jsou vytvořeny instance třídy *ChapterWrap*, která v sobě uchovává titulky kapitol, text kapitoly, pořadí kapitoly a své podkapitoly v kolekci typu *ObservableCollection*. Pokud je otevřená kniha formátu, který nepodporuje dělení na kapitoly, je do seznamu načtena vcelku. Načtené kapitoly jsou zobrazeny ve víceúrovňovém seznamu typu *TreeView*.



Obrázek 4.8: Okno s tvorbou kapitol

Pokud chce uživatel přidat novou kapitolu a v seznamu je jedna z kapitol aktivní, je nová kapitola přidána jako podkapitola, a pokud není, je přidána na konec seznamu kapitol. Jelikož v seznamu *TreeView* není možno přímo upravovat položky seznamu, je nutno upravit kopii kolekce kapitol a tu po modifikaci zobrazit v seznamu. Nejdříve je zavolána metoda *AddChapter*, jejíž vstupní parametry jsou kořenová kapitola *ChapterWrap*, index vybrané kapitoly a název nové kapitoly. Pokud se index kapitoly rovná indexu kořenové kapitoly, zavolá se druhá metoda *AddChapter* bez parametru *index*, kterou je přidána nová kapitola do seznamu podkapitol. Pokud se index nerovná, projdou se postupně všechny podkapitoly kořenové kapitoly, dokud není nalezena kapitola s hledaným indexem. Následně je seznam kapitol obnoven

z kopie metodou *RefreshChapters*.

Postup při přejmenování kapitoly je podobný – uživatel v dialogovém okně vyplní požadovaný název kapitoly a pokud je v seznamu jedna z kapitol aktivní, je uložen její index a poté, co je hloubkovým hledáním v kolekci kapitol nalezena, je změněn její název. Seznam kapitol je nutno obnovit z kopie metodou *RefreshChapters*. Stejný postup je použit i při posunu kapitol a přejmenování.

Před změnou aktivní kapitoly se dialogové okno dotáže uživatele, zda chce svou práci uložit. Pokud uživatel v dialogovém okně zvolí možnost Ano, uloží se změny do kolekce kapitol uložené v hlavním okně programu.

4.5 Vícejazyčná podpora

Program obsahuje uživatelské rozhraní ve více jazycích. Aby bylo možné dynamicky přepínat mezi jazyky při běhu programu bez nutnosti ho restartovat, bylo využito knihovny *WpfLocalizeExtension* [20]. Ta umožňuje získat a změnit jazykovou verzi současné instance pomocí třídy *LocalizeDirectory* a metody *SetCurrentThreadCulture*. Jazyky je možno přepínat pomocí tlačítka zobrazujícího vlajku současné jazykové verze, umístěném v pravé horním rohu hlavního okna.

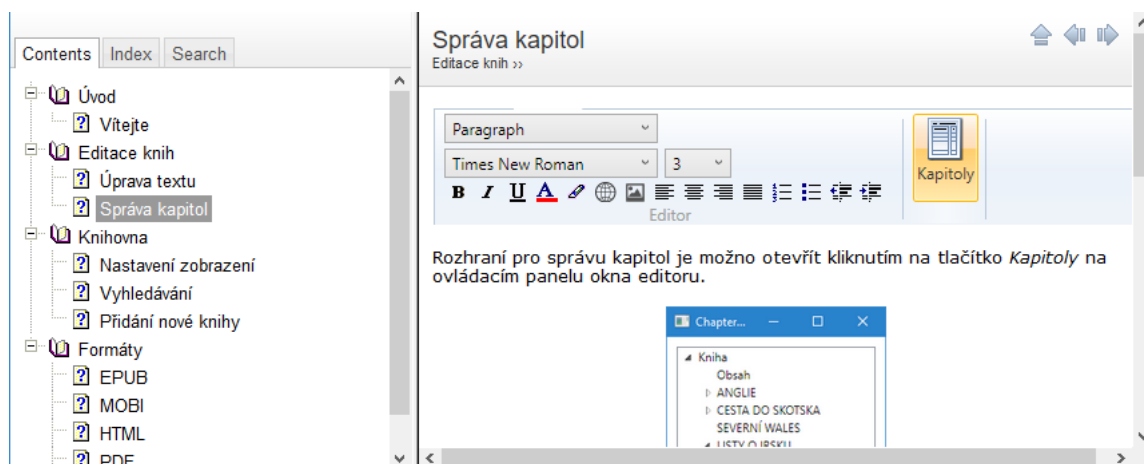
	Name ▲	Value ▾	Value
▶	AuthorColumn	Author	Autor
	BookArchive	Library	Knihovna
	BookArchiveButton	Library	Knihovna
	DateColumn	Last Changed	Poslední změna
	HomeRibbon	Home	Domovská nabídka
	LanguageColumn	Language	Jazyk
	LibraryLocationButton	Select...	Procházet...
	LibraryLocationGroupBox	Library Location	Umístění knihovny
	LibrarySettingsButton	Library Settings	Nastavení knihovny
	MenuItemAdd	Add To Library	Přidat do knihovny
	MenuItemClose	Exit	Zavřít
	MenuItemExport	Export To Format	Exportovat do formátu
	MenuItemExportEPUB	EPUB	EPUB
	MenuItemExportHTML	HTML	HTML

Obrázek 4.9: Ukázka jazykových souborů

Lokalizační řetězce jsou uloženy v souborech *Resources* [obrázek 4.9] a rozděleny podle jednotlivých jazyků. Výchozím jazykem programu je angličtina a pokud

v souboru jazyku chybí překlad určitého řetězce, zůstane text v angličtině. Řetězce s dynamickým textem jsou označeny anotací *lex:Loc*.

4.6 Nápověda



Obrázek 4.10: Ukázka nápovědy

Pro uživatele programu je připravena webová nápověda [obrázek 4.10], vytvořena v programu *HelpNDoc* [21]. Nápověda obsahuje popis všech ovládacích prvků, seznamuje uživatele s možnostmi programu, podporuje fulltextové vyhledávání se zvýrazněním hledaných slov a prohlížení tématu podle klíčových slov. *HelpNDoc* umožňuje export vytvořené nápovědy do všech běžných formátů, ke kterým kromě HTML patří i PDF nebo DOC. Aby byla zajištěna funkčnost na všech prohlížečích, byla zvolena šablona Legacy framed HTML template, jelikož standardní šablona obsahovala skripty, které při spuštění z lokálního zdroje nebyly podporovány prohlížečem Google Chrome.

5 Popis práce s formáty

S jednotlivými formáty je v rámci aplikace pomocí příslušné dvojice tříd `[formát]Open` a `[formát]Create`, které zajišťují respektive otevření (v některých případech dekodování) souboru a vytvoření nového souboru v požadovaném formátu. Každá třída má různý počet metod podle potřeb určitých formátů.

5.1 HTML

Jelikož ostatní formáty podporované programem jsou nějakým způsobem odvozené z HTML, bylo jeho zakomponování do programu logickým krokem. Komponenta `webBrowser` zobrazuje dokumenty ve formátu HTML implicitně, není potřeba tedy nic nastavovat a jednoduše přečíst text ze souboru. Příslušné třídy HTML jsou jednoduché a daly by se nahradit existujícími jednořádkovými metodami, ale pro konzistenci byly vytvořeny stejně jako u ostatních formátů.

5.1.1 Čtení

Třída `HTMLOpen` obsahuje jedinou funkci jménem `Open` s parametrem *fileName*, který reprezentuje cestu k HTML souboru. Návrátovou hodnotou je obsah HTML souboru získaný metodou `File.ReadAllText` z knihovny `System.IO` v datovém typu *String*. Tento textový řetězec zobrazují v programu v komponentě `webBrowser` metodou `NavigateToString`.

5.1.2 Vytváření

Třída `HTMLCreate` obsahuje jedinou funkci jménem `Create` s parametry *htmlContent* (textový řetězec obsahující HTML dokument) a *path* (cesta pro uložení souboru). Do souboru je zapsáno metodou `File.WriteAllText` z knihovny `System.IO`. Jelikož HTML soubor nemůže obsahovat metadata, při exportu jsou tyto informace ztraceny.

5.2 EPUB

5.2.1 Čtení

Třída `EPUBOpen` pro čtení využívá volně dostupnou knihovnu *EpubReader* [22] pro jednoduchou práci s formátem EPUB. Pomocí ní je z EPUB archivu možno extrahovat metadata, CSS soubory, HTML soubory i obrázky. Je tedy využita v metodě `Open` se vstupním parametrem `fileName`, reprezentujícím cestu k EPUB archivu a vracející jeho textový obsah. Archiv je načten do paměti pomocí metody `ReadBook`, která jej uloží do třídy `EpubBook`. Hledané HTML soubory jsou uloženy v proměnné `Content.Html` v třídě `Dictionary`, jde k nim tedy přistupovat skrz `for each` cyklus. Pomocí `for each` cyklu jsou sloučeny do jednoho textového řetězce `htmlContent`, který slouží jako návratová hodnota.

5.2.2 Vytváření

Jak už bylo zmíněno v teoretické části práce, EPUB soubor je ve skutečnosti ZIP archivem s jinou příponou. Jelikož .NET neobsahuje ve výchozí instalaci žádnou podporu pro vytváření ZIP archivů, pro jeho vytvoření byla použita volně dostupná knihovna *DotNetZip* [23]. V metodě `Create` s parametry `htmlContent` (textový řetězec obsahující HTML dokument) a `path` (cesta pro uložení souboru) je otevřená kniha nejdříve uložena do dočasného HTML souboru a pomocí metod `CreateContent` a `CreateTableOfContents` jsou vytvořeny další potřebné soubory.

`CreateContent` vytvoří XML dokument s obsahem podle specifikace. Do tagu `metadata` jsou zapsána metadata příslušná otevřené knize, do tagu `manifest` jsou zapsány všechny soubory obsahu (což je v tomto případě pouze jeden HTML soubor s veškerým textem a seznam kapitol) a do tagu `spine` se vloží jediná položka, soubor s textem. `CreateTableOfContents` je taktéž vždy totožný a obsahuje jediný `navPoint`.

Po vytvoření všech těchto potřebných souborů je třeba je zabalit do archivu. Je vytvořen datový proud `FileStream` v režimu `Create` (přepisujícím existující soubory), který je zabalen do streamu typu `ZipOutputStream`. Jako první krok je důležité vložit do souboru nezkomprimovaný soubor `mimetype` (čehož je docíleno tak, že jeho úroveň komprese `CompressionLevel` je nastavena na `None`). Po jeho zapsání je možno vložit ostatní soubory, tedy odkaz na kořenový soubor, soubor s obsahem, seznam kapitol a samotný text knihy.

5.3 MOBI

5.3.1 Čtení

Při programovém hledání bloku textu v MOBI souboru algoritmus postupuje v několika krocích:

- Najít začátek EXTH hlavičky
- Přečíst délku EXTH hlavičky, přeskočit na konec
- Přečíst název knihy, vložený za EXTH hlavičku
- Číst soubor, dokud není nalezen nenulový bajt
- Zjistit délku textu tak, že čteme soubor, dokud nejsou nalezeny dva nulové bajty za sebou, značící konec textu

Hlavní metodou třídy `MOBIOpen` je metoda `Open` se vstupním parametrem `fileName`, reprezentujícím cestu k MOBI souboru. Nejdříve je otevřen datový proud `FileStream` v režimu `Open`, který je předán jako vstupní parametr metodě `FindStartOfEXTH`. Datový proud je následně zabalen do proudu typu `BufferedStream` o velikosti 4 bajty, což je délka hledaného řetězce EXTH. Soubor je čten proudem, dokud není nalezen jeho konec (v kterémžto případě je návratovou hodnotou `-1`, značící, že se hledaný řetězec v souboru nevyskytuje) nebo pokud se buffer (pole bajtů) nerovná řetězci EXTH, taktéž uloženému v poli bajtů. Pokud se tak stane, je návratovou hodnotou současná pozice datového proudu mínus čtyři (délka hledaného řetězce).

Pokud se vrátí číslo rozdílné od `-1`, je ho možno společně s datovým proudem předat metodě `FindStartOfText`. Ta slouží k nalezení začátku zakódovaného textu, k čemuž je možno využít pevně dané souborové struktury MOBI – po EXTH hlavičce vždycky následuje název knihy a po něm následuje zakódovaný text oddělený nulovými bajty. Délka EXTH hlavičky je vždy uložena ve čtyřech bajtech po bajtech s hodnotou EXTH a když je přečten a převeden do datového typu `integer`, výsledkem je hodnota, o kterou je možno v datovém proudě přeskočit. Poté následuje titulky knihy (nenulové bajty) a mezera (nulové bajty) a metoda pokračuje ve čtení datového proudu, dokud nenajde první nenulový bajt, který je začátkem textu.

Po nalezení začátku textu, je možno dekomprimovat komprimovaný text metodou `DecompressMOBI`. Dekomprimace textu probíhá pomocí následujícího algoritmu [8]:

- Pokud se přečtený bajt rovná $0_{(16)}$, je uložen do výsledného textu tak, jak je
- Pokud se přečtený bajt rovná $N = 1_{(16)}$ až $8_{(16)}$, je N následujících bajtů uloženo do výsledného textu tak, jak jsou

- Pokud se přečtený bajt rovná $9_{(16)}$ až $7F_{(16)}$, je uložen do výsledného textu tak, jak je
- Pokud se přečtený bajt rovná $80_{(16)}$ až $BF_{(16)}$, tvoří s následujícím bajtem dvojici délka-vzdálenost. Je tedy přečten následující bajt, z těchto 16 bitů jsou odstraněny první 2, zbude tedy 14 bitů. Z nich prvních 11 bitů udává vzdálenost, o kterou se má algoritmus zpět vrátit a poslední 3 bity délku řetězce, ke které se má přičíst číslo 3 a následně řetězec o této délce zkopírovat do výsledného textu.
- Pokud přečtený bajt má jinou hodnotu (tedy $C0_{(16)}$ až $FF_{(16)}$), symbolizuje dvojici znaků – mezeru a znak, který je získán exkluzivním logickým součtem přečteného bajtu a $80_{(16)}$

Aby šlo algoritmicky zjistit, jak velké pole bajtů má být vytvořeno, je metodou `FindLengthOfText` zjištěna délka textu. Funguje stejně, jako algoritmus na dekomprimaci, ale pouze počítá, kolik znaků bude mít získaný text a skončí ve chvíli, kdy narazí na dva nulové bajty za sebou. Když je délka textu zjištěna, může být alokováno pole bajtů o zjištěné velikosti, do kterého bude metoda `DecompressMOBI` ukládat dekomprimovaný text. Po dokončení dekomprimace je pole bajtů převedeno do čitelného textového řetězce pomocí metody `Encoding.UTF8.GetString`, který může být zobrazen v komponentě *webBrowser*.

5.3.2 Vytváření

Kvůli komplexnosti a nepřesné zdokumentovanosti MOBI formátu byl pro jeho vytvoření zvolen oficiální nástroj od společnosti Amazon, KindleGen [24]. Jedná se o program příkazové řádky, který umožňuje z různých formátů vytvářet komprimované MOBI soubory. Pokud uživatel chce zkonvertovat knihu ve formátu EPUB, jeho obsluha je jednoduchá – stačí mu předat v argumentu jméno souboru, který chce zkonvertovat a KindleGen v jeho složce vytvoří soubor se stejným názvem ve formátu MOBI. Ve třídě `MOBICreate` a její metodě `Create` se vstupními parametry *htmlContent* (textový řetězec obsahující HTML dokument) a *path* (cesta pro uložení souboru) je nejdříve vytvořena kniha ve formátu EPUB pomocí metody `EPUB.Create` s názvem souboru *temp.epub*. Poté je definována instance třídy `Process`, které je parametrem `StartInfo` předáno jméno procesu (*kindlegen.exe*), argument s názvem souboru na konverzi (*temp.epub*) a styl okna skryté, aby se nezobrazovalo okno příkazové řádky s KindleGen. Poté je vytvořený soubor *temp.mobi* přesunut na umístění zadané vstupním parametrem *path*.

5.4 PDB

5.4.1 Čtení

Jelikož PDB používá totožnou kompresi jako MOBI, i postup pro čtení PDB knih je podobný. Hlavní metodou třídy *PDBOpen* je metoda *Open* se vstupním parametrem *fileName*, reprezentujícím cestu k PDB souboru. Metodou *FindStartOfText* je nalezen začátek textu, nacházející se za PDB hlavičkou. Poté je pomocí metody *FindLengthOfText* získána délka textu pomocí algoritmu popsaneho v kapitole 2.3.1, je alokováno pole bajtů a následně dekomprimován text o zjištěné délce a převeden do čitelného textového řetězce pomocí metody *Encoding.UTF8.GetString*.

5.5 PDF

5.5.1 Čtení

K práci s PDF soubory je v aplikaci využita knihovna *iTextSharp* [26] verze 5. Aby bylo možné z PDF souboru získat text v odstavcích, je nejdříve nutné vytvořit implementaci rozhraní *ITextExtractionStrategy*, jejíž návratovou hodnotou je text. Toto rozhraní obsahuje metodu s názvem *RenderText*, která je zavolána pro každý úsek textu na určité stránce PDF dokumentu. Vrací objekt typu *TextRenderInfo*, ze kterého lze kromě čistého textu získat informace o úseku textu, jako jsou například jeho souřadnice, jeho font a další. A jelikož řádek textu může být složen z více úseků textu, rozeznávaných PDF souborem, je možné skrz porovnání souřadnic úseků zjistit, zda patří na stejný řádek.

Pokud při porovnání souřadnic v implementaci rozhraní *ITextExtractionStrategy* dojde k zjištění, že se souřadnice řádku posunula, je string *result* přidán do pole stringů *strings*, symbolizující jednotlivé řádky. Pokud se souřadnice neposunula, je kontrolovaný úsek textu přidán na konec stringu *result*.

V metodě *Open* třídy *PDFOpen* je nejdříve otevřen PDF soubor pomocí třídy *PdfReader*, které je jako vstupní parametr předána cesta k souboru. Následně jsou for-cyklem pomocí metody *GetTextFromPage* třídy *PdfTextExtractor*, které jsou jako parametry předány otevřený dokument, číslo stránky a objekt třídy implementovaného rozhraní, projity všechny stránky a získán z nich text, který je pomocí HTML tagů *
* rozdělený na odstavce.

5.5.2 Vytváření

Pro vytvoření PDF souboru je opět využita knihovna *iTextSharp*. Jako první je vytvořen objekt *Document*, symbolizující PDF dokument. Ten je společně s datovým proudem *FileStream*, který vytvoří prázdný soubor na zadaném umístění, jako parametr předán metodě *GetInstance* třídy *PdfWriter*. Následně je s pomocí metody *GetPlainTextFromHtml* vstupní HTML dokument vyčištěn od HTML a JavaScript značek pomocí regulárních výrazů a výsledný čistý text zpracován objektem *StringReader*, v němž je možno k textu přistupovat po řádcích.

Cyklem *while* je následně zpracován celý text. Jeden řádek objektu *StringReader* se rovná odstavci textu, je z něho tedy vytvořen objekt typu *Paragraph*, kterému je pomocí parametru *Alignment* s hodnotou *Element.ALIGN_JUSTIFIED* nastaveno zarovnání do bloku a v parametru *Font* jsou nastaveny vlastnosti fontu, kterými jsou typ fontu a velikost. Takto vytvořený paragraf je do dokumentu přidán metodou *Add*. Po zpracování všech řádků je dokument uzavřen metodou *Close* a připraven ke čtení.

V metodě *GetPlainTextFromHtml* je text vyčištěn čtveřicí regulárních výrazů:

- $(\| \langle script(.+?) \rangle \| \langle style(.+?) \rangle \| \langle /script \rangle \| \langle /style \rangle \| \rangle)$ – odstraní všechny značky skriptů a stylů, včetně textu v nich obsažených
- $\langle . *? \rangle$ – odstraní všechny značky
- $\[s+\$[/r|n]*$ – odstraní všechny znaky nových řádků
- $\ $ – odstraní všechny nezlomitelné mezery

6 Porovnání s ostatními aplikacemi pro správu elektronických knih

Následující tabulka shrnuje funkcionalitu vytvořené aplikace a porovnává ji s její konkurencí.

	vytvořená aplikace	Calibre	Adobe Digital Editions	Alfa eBook Manager	Reedsy Book Editor	Vellum
čtení knih	ano	ano	ano	ano	ne	ne
editace knih	ano	ano	ne	ne	ano	ano
elektronická knihovna	ano	ano	ano	ano	ne	ne
nahrávání knih do zařízení	ano	ano	ano	ano	ne	ne
úprava metadat	ano	ano	ne	v placené verzi	auto. generováno	ano
zdarma	ano	ano	ano	omezená verze	ano	ne
open-source	možná	ano	ano	ne	ne	ne

Tabulka 6.1: Porovnání aplikací

Navržený program zvládá všechny základní operace, co by běžný uživatel měl potřebovat, chybí mu ale pokročilejší funkce, jako například podpora novinkových zdrojů v Calibre, stahování metadat z internetu či podpora pro půjčování elektronických knih z knihoven, do budoucna by ji ale bylo možné o tyto pokročilejší funkce rozšířit. Oproti složitějším programům ale může být výhodou nízká hardwarová náročnost a jednoduchost používání, uživatel ho tak může volit jako jednodušší alternativu složitějším programům. Nevýhodou programu je, že v současnosti není multiplatformní a podporuje pouze Windows.

Co se týče podporovaných formátů, tak navržená aplikace podporuje oba zásadní formáty elektronických knih (EPUB a MOBI) a rozšiřuje podporu formátů o tři další (PDB, PDF, HTML) s plnou podporu importu i exportu knih, čímž programy jako Adobe Digital Editions nebo Reedsy Book Editor překonává. V této statistice ovšem stále vede program Calibre.

7 Závěr

Cílem diplomové práce bylo vytvořit aplikaci pro konvertování formátů čteček elektronických knih. Nejprve došlo k seznámení s nejvyžívanějšími formáty a byly vybrány ty, které budou v diplomové práci podporovány. Byl vybrán archivační formát EPUB, který je nejrozšířenější neproprietární formát a je podporovaný největším množstvím elektronických čteček, komprimovaný formát MOBI určený pro čtečky Amazon Kindle, starší, ale rozšířený formát PalmDOC, ze kterého formát MOBI vychází a jako formát, který bude sloužit jako hlavní šablona a ze kterého se budou vytvářet všechny ostatní formáty při konverzi, byl vybrán klasický HTML dokument.

Byla vytvořena aplikace v jazyku C# v grafickém rozhraní Windows Presentation Foundation s uživatelským rozhraním Ribbon. Aplikace umožňuje otevřít knihy ve čtyřech uvedených formátech a libovolně mezi nimi exportovat. Pro každý formát byly vytvořeny třídy obsahující metody pro export a import, jejichž specifika byla popsána v textu práce. Kromě prostého zobrazení textu knihy ve formátu HTML umí aplikace číst i informace o knihách uložená v metadatech, pokud je ten určitý formát obsahuje. S pomocí navržené aplikace je knihy také možno vytvářet a upravovat.

Kromě funkcionality konverze obsahuje aplikace i jednoduché rozhraní pro archivaci knih a jejich uspořádávání do virtuální knihovny, k čemuž využívá metadata uložená v souborech elektronických knih, zobrazuje a uspořádává je do složek na disku. Do budoucna by řešení šlo rozšířit například o podporu více formátů, o rozšířenou archivní funkcionalitu nebo o rozšířené možnosti při zobrazení textu. Program by také bylo možno konvertovat na další operační systémy, jako například predominantně Mac a Linux.

Literatura

- [1] PETZOLD, Charles. *Mistrovství ve Windows Presentation Foundation: [aplikace = kód + markup]*. Brno: Computer Press, 2008. Mistrovství. ISBN 9788025121412.
- [2] Example of a ribbon (user interface element). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-05-24]. Dostupné z: [https://en.wikipedia.org/wiki/File:Example_of_a_ribbon_\(user_interface_element\).png](https://en.wikipedia.org/wiki/File:Example_of_a_ribbon_(user_interface_element).png)
- [3] MAREŠ, Amadeo. *1001 tipů a triků pro C# 2010*. Brno: Computer Press, 2011. ISBN 9788025132500.
- [4] HTML5. *W3C Recommendation*. [online]. 28.10.2014 [cit. 2017-05-24]. Dostupné z: <https://www.w3.org/TR/html5/>
- [5] Comparison of e-book formats. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-05-24]. Dostupné z: https://en.wikipedia.org/wiki/Comparison_of_e-book_formats
- [6] ePub. *MobileRead Wiki*. [online]. 2017 [cit. 2017-05-24]. Dostupné z: <https://wiki.mobileread.com/wiki/EPUB>
- [7] MOBI. *MobileRead Wiki*. [online]. 2017 [cit. 2017-05-24]. Dostupné z: <https://wiki.mobileread.com/wiki/MOBI>
- [8] PalmDoc. *MobileRead Wiki*. [online]. 2017 [cit. 2017-05-24]. Dostupné z: <https://wiki.mobileread.com/wiki/PalmDoc>
- [9] What is PDF? Adobe Portable Document Format. *Adobe Acrobat DC* [online]. Adobe Systems Incorporated [cit. 2018-05-12]. Dostupné z: <https://acrobat.adobe.com/us/en/acrobat/about-adobe-pdf.html>
- [10] *PDF Reference, Sixth edition, version 1.23* [online]. Adobe Systems Incorporated [cit. 2018-05-12]. Dostupné z: https://www.adobe.com/devnet/acrobat/pdfs/pdf_reference_1-7.pdf

- [11] PDF. *MobileRead Wiki*. [online]. 2017 [cit. 2018-05-12]. Dostupné z: <https://wiki.mobilerread.com/wiki/PDF>
- [12] About. *Calibre* [online]. [cit. 2018-05-12]. Dostupné z: <https://calibre-ebook.com/about>
- [13] Calibre User Manual. *Calibre 3.23.0 documentation* [online]. [cit. 2018-05-12]. Dostupné z: <https://manual.calibre-ebook.com/>
- [14] Adobe Digital Editions. *Adobe Digital Editions* [online]. Adobe Systems Incorporated [cit. 2018-05-12]. Dostupné z: <https://www.adobe.com/cz/solutions/ebook/digital-editions.html>
- [15] Features. *Alfa Ebooks Manager* [online]. Alfa.NetSoft [cit. 2018-05-12]. Dostupné z: <http://www.alfaebooks.com/features>
- [16] Compare Features. *Alfa Ebooks Manager* [online]. Alfa.NetSoft [cit. 2018-05-12]. Dostupné z: <http://www.alfaebooks.com/download>
- [17] The Reedsy Book Editor: A Powerful Writing Tool. *Reedsy* [online]. Reedsy, 2018 [cit. 2018-05-12]. Dostupné z: <https://reedsy.com/write-a-book>
- [18] Vellum | Create Beautiful Books. *Vellum* [online]. 180g, 2018 [cit. 2018-05-12]. Dostupné z: <https://vellum.pub/>
- [19] UwpDesktop 10.0.14393.3. *NuGet Gallery* [online]. Microsoft, 2018 [cit. 2018-05-12]. Dostupné z: <https://www.nuget.org/packages/UwpDesktop>
- [20] WpfLocalizeExtension 3.1.2. *NuGet Gallery* [online]. Microsoft, 2018 [cit. 2018-05-12]. Dostupné z: <https://www.nuget.org/packages/WpfLocalizeExtension/>
- [21] HTML help authoring tool, Word, PDF and eBook documentation generator. *HelpNDoc* [online]. IBE Software, 2018 [cit. 2018-05-12]. Dostupné z: <https://www.helpndoc.com/>
- [22] versfx/EpubReader: .NET library for reading EPUB files. *GitHub*. [online]. 2017 [cit. 2017-05-24]. Dostupné z: <https://github.com/versfx/EpubReader>
- [23] DotNetZip Library - Home. *CodePlex*. [online]. 2017 [cit. 2017-05-24]. Dostupné z: <http://dotnetzip.codeplex.com/>
- [24] KindleGen. *Amazon*. [online]. 2017 [cit. 2017-05-24]. Dostupné z: <https://www.amazon.com/gp/feature.html?docId=1000765211>

- [25] Pinvoke.net: SHOpenWithDialog (shell32). *Pinvoke.net* [online]. Red Gate [cit. 2018-05-12]. Dostupné z: <https://www.pinvoke.net/default.aspx/shell32/SHOpenWithDialog.html>
- [26] IText PDF, easy PDF generation for Java or .NET developers. *IText* [online]. iText Software, 2018 [cit. 2018-05-12]. Dostupné z: <https://itextpdf.com/>
- [27] UWP APIs callable from a classic desktop app. *MSDN-the microsoft developer network* [online]. Microsoft, 2018 [cit. 2018-05-12]. Dostupné z: [https://msdn.microsoft.com/en-us/library/windows/desktop/mt695951\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/mt695951(v=vs.85).aspx)

A Obsah přiloženého DVD

- text diplomové práce
 - diplomova_prace_2018_Karel_Sir.pdf
 - složka se zdrojovým kódem práce v \LaTeX , obrázky použitémi v práci a šablonou pro vysázení
- zdrojové kódy aplikace