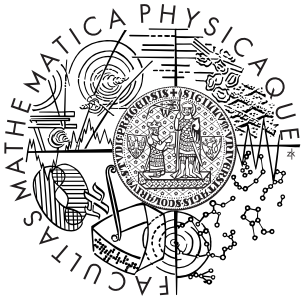
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Richard Kostecký

Modulární systém pro hromadné operace s metadaty souborů

Katedra softwarového inženýrství

Vedoucí bakalářské práce: Mgr. Martin Trčka
Studijní program: Informatika

2007

Poděkování patří zejména vedoucímu práce Martinu Trčkovi a pak také rodině a všem, co měli se mnou trpělivost a psychicky mě podporovali v době psaní této práce.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 7.8.2007 Richard Kostecký

Obsah

Název práce: Modulární systém pro hromadné operace s metadaty souborů
Autor: Richard Kostecký
Katedra (ústav): Katedra softwarového inženýrství
Vedoucí bakalářské práce: Mgr. Martin Trčka
e-mail vedoucího: Martin.Trcka@mff.cuni.cz

Abstrakt: V předložené práci ukazujeme implementaci obecných knihoven pro hromadné operace s metadaty souborů a ukazujeme výhody i nevýhody řešení tohoto problému. Dále předvedeme dvě ukázková spojení přes definované rozhraní k dalším externím knihovnám, také je ukázáno spojení s vlastním uživatelským rozhraním. Toto rozhraní bude předvedeno na implementaci editoru ID3 tagů v hudebních souborech s kódováním MP3. Druhou ukázkovou implementací bude prohlížeč a editor komentářů Exif tagu u obrázků JPEG.

Klíčová slova: metadata, modulární, systém, ID3, Exif

Title: Modular system for mass operations with file metadata
Author: Richard Kostecký
Department: Department of Software Engineering
Supervisor: Mgr. Martin Trčka
Supervisor's e-mail address: Martin.Trcka@mff.cuni.cz

Abstract: The implementation of modular system for metadata mass operations is presented in this thesis. Advantages and disadvantages of possible solutions are shown. There are also two example implementations included to show how to interface this modular system to other modules. Also we show how to create user friendly interface to our modular system. The first example is ID3 tag editor in MP3 files for batch tag editing. The second one is Exif tag viewer and comment writer for JPEG files.

Keywords: metadata, modular, system, ID3, Exif

Kapitola 1

Úvod

Našim cílem je vytvořit obecnou strukturu, která v sobě bude schopna uchovávat rozličná metadata z různých typů souborů. Úkolem je aby tato struktura, kromě vlastního uchovávání dat zvládala i práci s metadaty. Je potřeba zajistit základní zpracování metadat, jako je například jejich načtení ze souboru, editace a uložení. Obvykle je také nutné zabezpečit jejich předání aplikaci k dalšímu zpracování. Ke splnění tohoto úkolu musíme nadefinovat rozhraní, přes které bude naše struktura komunikovat. Prvním úkolem je navrhnout, jak by měla vypadat struktura, ve které budeme uchovávat metadata jednoho souboru. Druhým problémem je zajistit zpracování velkého množství metadat zároveň a rozhodnutí, která metadata budou držena v paměti a která pouze načítána z disku.

Dále je třeba vytvořit ukázkové implementace k našemu rozhraní, jelikož se napojení nejlépe ukáže na příkladě. Vytvoříme dvě ukázková uživatelská rozhraní a také moduly, které budou obsluhovat práci s vlastními metadaty a definovat jejich vlastnosti. Tyto ukázkové implementace budou ID3 tag editor v hudebních souborech MP3 a editor komentářů v Exif metadatech v obrázcích typu JPEG.

Pro tuto strukturu byl upraven původní ročníkový projekt, kterým byl editor ID3 tagů v souborech MP3. Jelikož obecně je lepší vyzkoušet strukturu na více typech metadat, rozhodli jsme se vytvořit druhou implementaci pro prohlížení metadat a editaci komentářů v obrázcích typu JPEG.

Pro práci s konkrétními metadaty obvykle existuje knihovna, která nám zprostředkovává práci s těmito metadaty. Tyto knihovny ovšem nemají žádné, ani přibližně jednotné rozhraní. Toto rozhraní v této práci vytvoříme a popíšeme jeho napojení na konkrétní knihovny na příkladě našich dvou ukázkových aplikací. Dále vytváříme ještě třídu, která nám umožní hromadné zpracovávání metadat. Nad naším jednotným rozhraním je potom jednoduché vytvořit vlastní uživatelské rozhraní, které nemusí být tak jednoduché a hlavně jednostranně zaměřené, jako je tomu u zmíněných ukázkových aplikací.

Kapitola 2

Metadata

2.1 Co jsou to metadata

Definice metadat není jednoduchá, přesněji existuje několik definic snažících se popsat metadata. Nejčastěji používaná definice říká, že metadata jsou data na datech. Jiná je zase definuje jako informace o datech.

Rozdíl mezi daty a metadaty není velký, někdy se nedá rozpoznat, zda se jedná o data nebo metadata. Například nadpis je určitě součástí dat, ale zároveň může být součástí metadat, protože popisuje text který za ním následuje. Případně si také data s metadaty mohou měnit role. Text básně jsou data, ale pokud tuto báseň bereme jako slova písně, tak tentýž text lze brát jako metadata k audio souboru, který obsahuje tuto píseň.

2.2 Využití metadat

Metadata slouží k popisu dat. Jsou uložena v téměř všech typech souborů vytvářených počítačovými aplikacemi a ukládají mnoho různých údajů. Některé tyto údaje jsou na první pohled zřetelné, jiné nejsou uživateli vůbec viditelné a zpracovává je pouze aplikace. Přestože tato skrytá metadata mohou aplikaci, či programátorovi mnoho napovědět o uložených datech a tím zlepšit a zrychlit jejich zpracovávání. Nevýhodou je často velmi složitý způsob uložení metadat a existuje mnoho typů uložení. Další nevýhodou je často se měnící způsob uložení metadat s vývojem aplikací. Potřeby programátora se s časem mění a proto se mění i položky a místo jejich uložení. Některá metadata jsou navržena pro optimalizace kompresních algoritmů, například u obrázků popsáním co se nachází na pozadí a které oblasti je nutné mít v popředí. Pomocí tohoto uspořádání je například možno provést kompresi jednotlivých částí odděleně. Tím lze zlepšit kompresní poměr. Jiná metadata mohou být určena čistě k popisu vlastního datového souboru a netýkají se jeho vnitřní struktury. Příkladem takovýchto metadat je komentář u digitální fotografie.

2.3 Typy metadat

Metadata se dají rozdělit na několik typů. Existují metadata, která popisují obsah, tedy popisují zdroj jako celek, například jméno písně či autora. Další metadata mají logickou funkci a udávají informace o rozdělení vlastních dat, kde popisují obsah souboru. Například místo uložení náhledu v obrázku.

Metadata lze také rozlišovat, podle způsobu uložení. Existují dvě základní možnosti uložení metadat souborů a to „interní“ a „externí“. Interní uložení znamená, že metadata jsou uložena zároveň s daty. Výhodou tohoto přístupu je jednoduché nalezení metadat a snadnější manipulace s nimi. Takto uložená metadata se týkají pouze toho souboru ve kterém jsou uložena. Tento postup však často vede k velké redundanci dat. Druhou možností je uložení externě, které umožňuje lepší vyhledávání, často využívané při práci s databázemi. Nevýhodou tohoto externího přístupu je složitější práce s metadaty při změně dat. V této práci se budeme zabývat prvním případem, tedy metadaty uloženými spolu se souborem a jejich editací.

Metadata mají mnoho způsobů využití. Uplatnění nacházejí například v souborových systémech, protože každý souborový systém obsahuje metadata. Příkladem jsou časové známky a atributové příznaky, případně další data, které určuje bližší specifikace souborového systému. Často se setkáváme s metadaty u hudebních souborů, kde jsou uloženy údaje o skladbě. Tato metadata obsahují autora a jméno skladby, údaje o její délce, název alba a spoustu dalších položek. V metadatech obrázků fotoaparáty ukládají informace za jakých podmínek byla fotka pořízena. Případně zde může být uloženo jméno autora, komentář či jiné informace.

2.4 Metadata v hudebních souborech

Digitální hudební soubory navíc k vlastní hudební stopě obsahují i další s touto stopou spojené informace: jméno písně, autora, alba, rok vydání. Přidávání těchto informací je známo jako "tagging", z tohoto důvodu přidané informace označujeme jako tag. Pro formát wav neexistuje specifikace žádného tagu, ale další formáty hudby již většinou obsahují tagy metadat. Zde se budeme zabývat ID3 tagy hudebních souborů MP3.

ID3 tagy v MP3 souborech mají z historického hlediska dvě základní verze a to verzi 1 a verzi 2.

Verze 1 obsahuje celkem 128 bytů dat, která začínají řetězcem TAG. Tento tag se nachází na konci souboru a maximální velikost jedné položky je 30 bytů. Později byla tato verze přidáním komentáře rozšířena na verzi 1.1. Tato verze zcela postrádá jakoukoliv podporu pro národní jazykové sady.

Verze 2 toho s verzí 1 nemá mnoho společného. Podporuje streamované audio a proto tag může být umístěn na začátku souboru. Také již nemá fixní velikost tagu a má proměnnou velikost. Velikost jedné položky je omezena na 16 MB přičemž maximální velikost celého tagu je 256 MB. Současnou

poslední variantou verze 2 je verze 2.4.

ID3v2 tag se skládá z těchto částí: hlavičky, nepovinné rozšířené hlavičky, položek nesoucích data, vycpávky a patičky. Poslední dvě části tagu se navzájem vylučují. Hlavička obsahuje základní informace o ID3 tagu a také příznaky jaká data tag obsahuje, dále také jeho verzi a délku. Rozšířená hlavička obsahuje další pomocné údaje o tagu. Vycpávka je umístěna za položkami a slouží k tomu, aby nebylo nutné přepisovat celý soubor při přidávání nějaké malé položky, pokud tato položka zabírá méně prostoru než je prostor zabraný vycpávkou. Jelikož tag nemůže mít vycpávku mezi položkami, je tato část tagu neslučitelná s patičkou. Patička se používá pro urychlení hledání tagu, pokud probíhá hledání od konce souboru. Obsahuje stejné položky jako hlavička, ale má jiný identifikátor.

Datové položky ID3 tagu obsahují hlavičku položky, která obsahuje identifikátor položky (Frame ID), údaj o délce a další hodnoty. Pro každý frame lze také zadat kódování, přičemž toto kódování platí pro všechny textové i numerické řetězce a URL. Je možno si vybrat z ISO-8859-1, UTF-16 a UTF-8. Kódování UTF nejsou příliš podporována přehrávači. Každá položka má vlastní data. Ta podle typu položky mohou mít i více záznamů pod jedním identifikátorem. Velikost položky nemá zvláštní omezení. Celá položka se i s řídícími záznamy musí vejít do 16 MB. ID3v2 obsahuje 84 předdefinovaných položek a vyskytují se zde i další které lze dodefinovat. Pak ovšem lze očekávat nulovou podporu těchto položek jinými aplikacemi než ta, pro kterou byla tato položka dodefinována. Mezi předdefinovanými položkami jsou například autor, jméno skladby, název alba, rok vydání přičemž všechny tyto hodnoty jsou uváděny jako řetězce. Jednou z definovaných položek je obrázek, který může vyjadřovat celkem 21 různých hodnot z tagu. Bližší specifikaci ID3 tagu, lze najít v [3].

Problémem ID3v2 je, že pro spousty různých typů položek je potřeba velké množství parserů, které nám tyto data vytáhnou. Místo psaní těchto parserů lze použít již hotovou knihovnu například ID3lib, která nám tento problém řeší a poskytuje přístup k datům.

2.5 Metadata v obrázcích

Dalším krásným příkladem metadat v souborech jsou metadata obsažena v obrázcích. Tato metadata jsou k obrázku přidána také jako tag, podobně jako u hudebních souborů. K obrázku se mimo jiná data dají uložit informace popisující obsah tohoto obrázku nebo jiné fráze přibližující tento obrázek. Toto ukazuje jedno z možných využití tagu a to urychlení hledání mezi obrázky. Jinak nelze podle vlastního obrázku vyhledávat a spojování s textem, který se nachází v blízkosti těchto obrázků také není nejlepším řešením.

Digitální fotografie přináší rozšíření do možností metadat v obrázcích. Digitální fotoaparáty ukládají do těchto obrázků data, která zlepšují a hlavně

zrychlují pozdější zpracování.

Pro metadata v obrázcích existuje několik formátů. Jako ukázkový případ si vezmeme Exchangeable image file format známý spíše pod zkratkou Exif. Tento formát metadat byl vyvinut sdružením výrobců fotoaparátů Japan Electronics and Information Technology Industries Association (JEITA), pro formáty souborů kódované pomocí JPEG, TIFF a RIFF. Exif tag obsahuje mezi základními údaji datum a čas pořízení snímku, dále nastavení fotoaparátu při pořízení snímku, typ a výrobce fotoaparátu, informace o poloze, pokud tyto informace dodává fotoaparát. Případně tyto informace mohou být později dodány ručně. Mezi další položky metadat patří popis obrázku a informace o autorovi a copyrightu. Nevýhodou Exif tagu je stejně jako v TIFF formátu možnost rozptýlení tagu po celém souboru. To může při editaci způsobit problém, neboť při nesprávném přepsání ukazatelů může být poškozen celý obrázek. Podle standardu je maximální velikost Exif tagu 64 kB, což působí další problém. Při uložení náhledu, který používají fotoaparáty pro zobrazení obrázku na LCD displeji, bývá tento obrázek často větší než maximálních 64 kB. Výrobci toto omezení řeší uložením druhého většího náhledu na konec souboru. Tento náhled bývá ztracen, pokud je obrázek otevřen a následně uložen v nějakém standardním editoru obrázků a stává se tak nekompatibilním s fotoaparátem, který tento snímek pořídil.

Aktuální verzí Exif tagu, je verze 2.2. Exif specifikace. Není oficiálně udržovaná, protože za ní nestojí žádná organizace či lidé, kteří by ji upravovali pro současné požadavky. Přesto je tento typ tagu užíván téměř ve všech fotoaparátech. Exif tag je pouze upravován výrobci fotoaparátů ke splnění jejich požadavků, ale momentálně neexistuje žádný jednotný přístup. Exif tag je složen z těchto částí: TIFF hlavička, primární data o obrázku, privátní Exif data obrázku, GPS data a popisná data o náhledu, který je uložen v obrázku. Mezi daty uloženými v tomto tagu jsou obvykle šířka a výška obrázku, kompresní schéma, orientace, ale také i název obrázku, použitý software, komentář k obrázku nebo jméno osoby, která daný obrázek vytvořila. Pro textová data uložena v obrázku lze použít kódování ASCII, JIS, Unicode. Přestože tento typ tagu není blíže udržován byla vydána oficiální specifikace. Tato specifikace je k dispozici na [1].

Pro ulehčení přístupu k datům lze použít nezávislé knihovny Exiv2, která umožňuje zjednodušený přístup k metadatům. Řeší také konzistenci dat po uložení a nenarušení struktury obrázku, parsování dat a další věci, které by mohly způsobit problémy. V našem ukázkovém programu nebudeme přímo používat knihovnu, ale program Exiv2 od stejného programátora. Lépe tak lze ukázat možnosti napojení a spolupráci systému s jinými komponentami.

Kapitola 3

Analýza potřebných funkcí

3.1 Práce s knihovnou ID3lib

Prvním úkolem práce bylo zjistit, jaká všechna pole budou pro práci s tagy potřeba a jaké operace bude potřeba definovat. Zde byly použity poznatky získané při zpracovávání předchozího ročníkovém projektu, který se týkal editoru ID3 tagů v hudebních souborech MP3. Odsud také pochází mnoho postřehů pro další analýzu.

3.2 Funkce pro práci s metadaty jednoho souboru

Při práci s knihovnou ID3lib bylo zjištěno, že jsou za potřebí pouze tři základní operace s metadaty a to načtení (*Load*), uložení (*Save*) a aktualizace (*Update*). Aktualizace se oproti uložení liší v tom, že nepotřebuje znát všechny pole daného tagu. Tato funkce pouze přepíše hodnoty, které jsou definovány a zbytek hodnot ponechá v původním stavu. Proto pro samotnou práci ve vrstvě, která pracuje s metadaty jednoho souboru, stačí pouze tyto tři funkce. Jejich definice musí být nutně v obecné vrstvě virtuální, protože se budou pro každý typ metadat měnit. K těmto třem základním operacím je přidána ještě jedna operace pro kontrolu správných dat. Funkce zkontroluje výskyt správných položek a zkontroluje, zda jsou tyto položky správného typu. Také umožní přejmenování položek, což může být užitečné pro různé jazykové verze.

3.3 Funkce pro práci s více soubory

Problém nastává při určování, které operace budou potřeba při hromadném zpracování. Zde již nestačí jednoduché funkce, ale musí zde být hromadné operace. Dobrým příkladem je funkce *UpdateAll*, kde již funkce typu update dostává větší význam. Obvykle při měnění tagu není vhodné zapisovat všechny položky. Mění se pouze ty položky, které jsou společné pro všechny

editované soubory. Například v hudebních souborech jsou to autor nebo album, ale již to pravděpodobně nebude název skladby, protože ten již skladby nemají společný. Takže je v našem zájmu takovéto hodnoty ponechat. V této vrstvě má spíše menší význam funkce *SaveAll*, ale ta také nemusí být vždy zbytečná, protože někdy je výhodné přepsat celý tag a neponechávat žádnou původní hodnotu.

Funkce *LoadAll* nemá velký význam. Důvodem pro nepoužívání této funkce je omezená kapacita paměti. Přestože velikost tagů bývá omezena, nemusí to znamenat malou velikost. Sice metadata v obrázku mají velikost omezenu na 64 kB, což by nebyl takový problém. Ale při velkém počtu obrázků by i takto zabraná paměť mohla velmi rychle narůstat, pokud nechceme dávat omezení počtu zpracovávaných obrázků. U našeho druhého příkladu s hudebními souboru by už omezená kapacita mohla být problémem. Jeden ID3 tag může mít velikost až 256 MB a tak bychom nemuseli být v paměti schopni udržet ani malé množství tagů tohoto typu. Z důvodu zachování obecnosti je vhodné počet načítaných tagů nijak neomezovat. Na druhou stranu občas může být načtení všech tagů výhodou. Hlavní funkcí nebude zobrazení dat, ale příprava k editaci. Pak lze v paměti postupně upravovat hodnoty tagů a až po skončení editace je uložit všechny najednou. Krásným příkladem tohoto typu zpracování může být rozpoznávání tagu z cesty k souboru. Z těchto důvodů tuto funkci nadefinujeme. Načtení není možné udělat zcela obecně, proto podpora této funkce vychází z blíže specifikovaného modulu.

Další potřebnou funkcí pro hromadné operace je funkce, která nám přidá soubor do seznamu editovaných objektů. Tuto funkci nazveme *Add* a jedinou její funkcí bude přidání objektu do seznamu objektů, které budeme editovat, pokud tento soubor již není v tomto seznamu. Pokud se v seznamu nachází je, tak neudělá nic.

Sesterskou funkcí funkci *Add* je funkce *Remove*, která nám ze seznamu odstraní soubor, který jsme se rozhodli needitovat. S tímto je spojena i funkce *RemoveAll*, která nám vyprázdní seznam odkazů na soubory, které chceme editovat. Spolu s funkcemi pro přidávání a odebírání souborů by byla vhodná i funkce, která překopíruje seznam editovaných souborů pro zobrazení například v uživatelském prostředí.

Vzhledem k tomu, že ne vždy budeme přidávat pouze jeden soubor, zpravidla jich budeme přidávat více, tak ještě budeme potřebovat funkci, která nám bude schopna dodat seznam souborů pro hromadnou editaci. A zde se naskýtá rovnou několik možností, jak lze hromadně přidávat soubory a to:

- Pouhým přidáním všech souborů vyskytujících se v dané složce a celé její podadresářové struktuře.
- Použitím masky pro přidání souborů odpovídající této masce. Většinou editujeme soubory s nějakou příponou, například soubory *.mp3 pokud

chceme editovat ID3 tagy ve skladbách kódovaných pomocí MP3 kodeku. Ale vhodnější je větší flexibilita přidávání.

Kapitola 4

Implementace modulárního systému

4.1 Vývoj

Projekt je napsán v jazyce C++ a je vyvinut pro použití na platformě Windows. K vývoji bylo použito Microsoft Visual Studio 2005. Vzhledem k závislosti části kódu na systému souborů, není tento projekt přenositelný na jiné platformy. K vývoji ukázkových aplikací bylo taktéž použito Microsoft Visual Studio a jejich interface je vytvořen v Microsoft .NET framework 2.0, který je tudíž nutný pro běh těchto aplikací. Více informací o tomto .NET frameworku lze najít na [5]. Pro ukázkové aplikace byla využita knihovna ID3lib pro práci s metadaty hudebních souborů MP3, více informací o knihovně lze získat na [4]. Pro druhou aplikaci byl použit program Exif2 pro ukázku napojení na jiné moduly. O tomto programu lze najít více informací na stránkách [2].

4.2 Základní datová struktura

Jedním z hlavních problémů je kde a jak ukládat data. Nejtěžší věcí je nadefinovat, která data je potřeba uchovávat. Vezmeme to postupně od nejmenších částí. Začneme u definice jedné položky. Ta může být uložena jako různé datové typy. Uložení je možno lehce vyřešit pomocí například použitého unionu, případně by jde uchovávání dat vyřešit pomocí templatů. Hlavním problémem jsou možné varianty, co lze do jedné položky uložit. Kromě vlastních dat musí být s touto položkou uloženy i další informace jako je například typ ukládaných dat. Nutnou položkou je i jméno, nebo nějaký jiný identifikátor, podle kterého se bude tato položka dále rozpoznávat v modulu, který bude přímo pracovat s metadaty. Také se nesmí zapomenout na to, že data jsou velmi často nějak omezena, ať již velikostí, či maximálním rozsahem. Teoreticky nemusí být toto omezení definováno a může se o něj starat modul, který blíže definuje přímou práci s metadaty. Snažíme se jít cestou

maximálního zjednodušení modulu, je vhodné toto omezení umístit zde. Sice zde budeme definovat pouze základní omezení, není problém dodefinovat další omezení podle potřeby. Tímto máme vyřešenu jednu datovou položku. Ještě by k této položce šlo přidat kódování, které by bylo při některých implementacích vhodné, ale to by bylo zbytečně redundantní. Většinou budeme požadovat jedno kódování pro celá metadata, a tak bude lepší toto kódování definovat v modulu. Případně lze uměle přidat do tagu položku, která ponese informaci o znakové sadě.

Celá struktura jedné položky je hotova, takže nyní lze přejít na vyšší úroveň, na celá metadata. Tato úroveň musí obsahovat seznam jednotlivých položek. Tyto položky jsme zde umístili do kontejneru map, aby šlo rychle vyhledávat mezi jednotlivými položkami. Vyhledávání bude probíhat podle řetězce, který bude jednoznačně identifikovat položku. Tento řetězec nemusí být nutně stejný jako ten, který je uveden v datové položce, ale bylo by to vhodnější. Další datovou položkou, kterou musí tento tag obsahovat je cesta k souboru o jehož metadata se jedná. Pak lze volat funkce bez parametru a tím zjednodušit přístup. Žádné další datové položky pro metadata k jednomu souboru už nejsou potřeba.

Na závěr je ještě třeba dodefinovat položky, které budou využity při hromadné editaci metadat. Těchto datových položek není mnoho. Je nutné mít seznam souborů, přesněji cest k souborům, které budou editovány. Dále pak je třeba mít jednu datovou reprezentaci tagu, specielně pro případ hromadného updatu. Ten se bude provádět z tohoto tagu, protože budeme ukládat vždy stejná metadata. Také budeme mít strukturu, která bude spravovat data více tagů najedou, například pro funkce *LoadAll* a *SaveAll*, případně pro jiné externí moduly pracující s touto aplikací.

4.3 Implementace funkcí

Začneme opět u implementace funkcí vrstvy pro práci s metadaty jednoho souboru. Některé funkce sice v naší vrstvě nebudeme implementovat, protože jsou implementačně závislé na modulu se kterým jsou spojeny, ale uvedeme zde pár postřehů z praktické implementace napojení na knihovnu ID3lib při vytváření ukázkového editoru ID3 tagů písní kódovaných kodekem MP3. V této vrstvě budou definovány tyto funkce:

- *Update*
- *Load*
- *Save*
- *verify_tag*

Nyní se podíváme na každou zvlášť. Při implementaci ukázkového editoru se objevuje první důležitý poznatek o funkci *Update*. Tato funkce by stejně

musela být implementována v blíže definující vrstvě a není problém ji implementovat v našem modelu pomocí funkcí *Load* pro načtení metadat ze souboru a *Save* pro uložení metadat do soubor. V této vrstvě bude ještě definována funkce *verify_tag*, která provede kontrolu, zda vyplněná políčka patří do tagu a zda jsou správného typu.

Nyní přejdeme k implementaci funkcí vrstvy pro hromadné zpracování metadat. V této vrstvě používáme dvě skupiny funkcí a to skupinu pro práci se seznamem souborů a skupinu pro hromadné operace s metadaty. Nejdříve se podíváme na funkce pro práci ze seznamem, a to:

- *Add*
- *AddMore*
- *Remove*
- *RemoveAll*
- *NahrajSeznam*

První funkcí je *Add*. Na této funkci není nic zvláštního, pouze přidá dodanou cestu do vektoru, který obsahuje seznam odkazů na soubory, jejichž metadata budeme editovat. Stejně tak na první pohled vypadá, že ani na funkcích *Remove* a *RemoveAll* není nic zvláštního. Nicméně je zde jeden lehce přehlédnutelný detail. Jelikož v paměti držíme rozpracované tagy, tak je musíme zrušit a smazat jejich obsah dříve, než je smažeme ze seznamu editovaných souborů. Pro přidávání souborů je přítomna funkce *AddMore*, která přidá soubory ze složky i všech jejích podsložek. Tato funkce dostane dva parametry, cestu ke složce a podmínku. Přidá do seznamu položek pro editaci všechny soubory ze složky splňující podmínku. Funkce *NahrajSeznam* dostane jako parametr ukazatel na kolekci a po provedení této funkce bude v této kolekci seznam editovaných souborů.

V druhé skupině funkcí vrstvy hromadného zpracování metadat jsou tyto funkce:

- *LoadAll*
- *SaveAll*
- *UpdateAll*

Jako první funkci pro hromadné operace si vezmeme *LoadAll*. Její úlohou je načtení všech tagů souborů, které jsou uloženy v seznamu souborů pro editaci. Tato funkce sice není nezbytně nutná, ale mohla by se v některých případech ukázat jako prospěšná. Další nezbytnou funkcí je hromadné uložení pomocí *SaveAll*. Tato funkce podle zadaného parametru buď pouze postupně pro každý soubor zavolá funkci pro uložení tagu a této funkci přiřadí kopii metadat, nebo pro každý tag ze struktury, která je drží, zavolá *Save*.

Složitější je to s funkcí *UpdateAll*. Jak jsme již dříve vysvětlili, není třeba definovat chytrou funkci *Update*, proto vše musí zvládnout funkce *UpdateAll* sama. Implementace této funkce projde všechny soubory, které hodláme editovat. U každého souboru načte jeho metadata, projde všechny položky přepisovaných dat a tato data nahradí novými hodnotami. Pokud není položka v původním souboru uložena, měla by se přidat, což by měla zajišťovat funkce *Update*. Funkce *Update* se spouští po každém přepsání hodnot.

Kapitola 5

Konkrétní implementace

5.1 Editor ID3Tagů v hudebních souborech

K této konkrétní implementaci byla použita knihovna ID3lib, která zajišťuje práci s vlastními metadaty v hudebních souborech. Knihovna sice umožňuje práci s metadaty, ale neposkytuje žádné rozumné rozhraní. Je nutné si toto rozhraní vytvořit a až nad tímto základním rozhraním, pak následně vybudovat rozhraní pro práci pomocí naší struktury. Příkladem proč je vhodné si vytvořit pomocné rozhraní je, že knihovna sice umožňuje načíst položku z metadat s příslušným identifikátorem, ale již neprovádí další parsování těchto dat, či případné konverze při použití jiné než doporučené znakové sady. Proto je nutné si tyto funkce nejdříve vytvořit a teprve potom definovat rozhraní, které budeme používat.

Dále bylo potřeba vytvořit uživatelské rozhraní, které by nám umožnilo relativně jednoduché použití aplikace. Toto rozhraní bylo uděláno maximálně funkční, aby byl tento program použitelný a konkurence schopný mezi standardními programy pro editaci ID3 tagů v hudebních souborech. Tento program by měl pokračovat ve vývoji i v budoucnu zrychlením a přidáním dalších vlastností než jen základní módy práce, které jsou zabudovány v této aplikaci. Jedná se o tyto módy:

- Editace pomocí přímo zadaných dat, kde se přímo zadávají data později ukládaná do metadat.
- Vytvoření metadat podle zadané masky. Po zadání této masky se podle ní vygenerují hodnoty, které budou uloženy do metadat. Toto se zpravidla používá při první editaci tagů, případně při opravě špatně zadaných tagů.
- Nahrání ID3 tagu pro celé album písní z databáze freeDB2.

Pro tento program byly vytvořeny 2 moduly, které se napojují na vrstvu hromadných operací s metadaty. Jedná se o modul zajišťující rozeznávání tagu ze zadané masky a cesty s souboru. Druhý modul se stará o komunikaci

s databází freeDB2. Oba tyto moduly naplní strukturu, která drží seznam tagů a následně se na tuto strukturu zavolá funkce *SaveAll*. Toto umožňuje během procesu zobrazit uživateli náhled změn. Pokud je uživatel s těmito změnami spokojen, potvrdí jejich uložení, jinak uložení provedeno nebude.

5.2 Prohlížeč metadat v obrázcích

Tato implementace byla vyvinuta jen jako ukázka jak lze používat naše rozhraní. Tato aplikace pracuje s externím programem Exiv2, který používá konzolové rozhraní pro výpis tagu a příkazový řádek pro změnu hodnot v tagu. Zde byl pouze vytvořen systém na volání příkazu pro editaci metadat a skládání tohoto příkazu. Také k tomuto programu bylo vytvořeno uživatelsky přívětivé prostředí.

5.3 Vytváření nových aplikací

V této kapitole popíšeme, co vše je potřeba udělat pro naprogramování nové aplikace, která by používala námi definované rozhraní. Za tímto účelem je potřeba vytvořit základní části a to:

- Modul pro vrstvu pro práci s jedním souborem.
- Uživatelské rozhraní, které bude spolupracovat s vrstvou pro hromadné operace se soubory.
- Modul pro práci s vrstvou pro hromadné operace.

Modul pro práci s jedním souborem musí dědit ze třídy Metadata. A měl by definovat základní funkce a to *Load*, *Save* a *Update*. Žádná z těchto funkcí není povinná, ale obecně pro práci s medatady jsou nezbytné. Funkce *Load* by měla naplnit mapu *values* správnými hodnotami. Tato struktura je dostatečně popsána ve třídě *Metadata*. Také je vhodné v konstruktoru naplnit pole *mapovani_hodnot* a *field_types* pro správné fungování funkce *verify_tag*.

Uživatelské rozhraní a modul pro práci s vrstvou pro hromadné operace budou používat stejné rozhraní. Je nutné mít vytvořen objekt *MetadataAll*, který bude spravovat soubory pro editaci. Soubory lze přidávat pomocí již popsaných funkcí *Add* a *AddMore*. Také odebírání probíhá pomocí již popsaných funkcí *Remove* a *RemoveAll*. Ukládat nově vytvořené tagy lze do struktury *Tagy* v tomto objektu. Po zavolání funkce *SaveAll* budou tyto tagy uloženy. K jejich uložení lze také vyžít funkce *UpdateAll* a *SaveAll(true)*, které uloží do tagů souborů v seznamu data, která jsou v objektu *MData*.

Kapitola 6

Instalace

6.1 Obsah CD

Obsah CD je rozdělen na složky, kde jsou umístěny zdrojové kódy našeho programu, spustitelné verze ukázkových příkladů a také všechny požadavky potřebné k spuštění našeho programu kromě samotného operačního systému. Také se zde nacházejí testovací data, na kterých si lze vyzkoušet funkčnost ukázkových programů. Tato data musí být překopírována na pevný disk či jiné médium, na které lze zapisovat, jinak editory nemohou fungovat.

Ve složce **Requriements** jsou umístěny programy, které je potřeba mít nainstalovány, aby bylo možno náš program spustit. Zde se také nachází instalační soubor .NET framework 2.0.

Test data obsahují ve dvou podsložkách music a pictures testovací data pro naše dvě ukázkové aplikace.

Ve složce **Source** jsou uloženy všechny zdrojové soubory ukázkových projektů.

Složka **Binary** obsahuje v odpovídajících podsložkách spustitelné verze našich dvou aplikací.

6.2 Instalace

Instalace ukázkových programů je jednoduchá. Je pouze třeba vytvořit kopii složky s programem do uživatelem zvoleného adresáře. Tím se zkopírují i všechny potřebné knihovny a programy, které jsou ve složce přiloženy. Programy se spustí klasickým dvojklikem na spustitelný soubor.

Jediným problémem, který by mohl nastat je absence požadovaného .NET framework 2.0, což lze vyřešit jeho nainstalováním. Framework lze získat ze stránek firmy Microsoft, případně je také umístěn na CD přiloženém k bakalářské práci.

Kapitola 7

Závěr

7.1 Závěr

Pro účely této práce byl vytvořen obecný modulární systém, který umožňuje jednoduchý a hlavně shodný přístup k mnoha různým typům metadat. Tento systém je tvořen dvěma vrstvami a to vrstvou pro práci s jedním souborem a vrstvou pro hromadné operace se soubory. Tyto vrstvy umožňují provádět načítání metadat ze souborů a jejich uchovávání. Také umožňují ukládání metadat, a to jak načtených ze souboru a upravených, tak i nově vytvořených.

Vytvořili jsme všechny základní funkce pro práci s metadaty. Tyto funkce zajišťují všechny základní operace jako načtení, editaci a uložení metadat. Většina původně uvažovaných funkcí není v obecném modelu k užitku a tudíž nemělo smysl je definovat. Tyto okolnosti byly zjištěny až v průběhu práce na systému a nebylo s nimi počítáno při začátku práce.

Dále byly vytvořeny dva moduly zajišťující práci s konkrétními metadaty, a to ID3 tagy hudebních souborů MP3 a Exif tagy obrázků. Také byly vytvořeny moduly, které pracují s vrstvou pro hromadné operace. Těmito moduly jsou modul pro rozeznávání tagu z masky a cesty k souboru a modul pro práci s externí databází freeDB2. Za modul pro práci s vrstvou pro hromadné operace by se také dalo považovat uživatelské rozhraní, které s touto vrstvou spolupracuje a dodává jí data.

Jako ukázka funkčnosti systému byly vytvořeny z výše popsaných modulů dvě ukázkové aplikace. Prvním programem je jednoduchý editor a prohlížeč Exif tagů, který ukazuje základní využití našeho modulárního systému v nejjednodušší podobě. Tato aplikace má pouze jeden modul připojen na vrstvu pro práci s jedním souborem. Pro práci s ní bylo vytvořeno uživatelsky přívětivé prostředí. Druhou složitější aplikací je editor ID3 tagu v MP3. Tento editor má také jeden modul připojen k vrstvě pro práci s jedním souborem, ale k vrstvě pro hromadné operace jsou připojeny moduly hned dva. Také zde bylo vytvořeno uživatelsky přívětivé prostředí. Program využívá dva, pro něj definované, moduly a to jeden pro rozeznávání tagu z masky a druhý pro práci s externí databází freeDB2.

7.2 Možnosti rozšíření

Jednou z možností rozšíření vytvořeného modulárního systému by mohlo být napsání univerzálního konzolového rozhraní, pomocí kterého by bylo možno editovat metadata obecně pomocí příkazů. Pro toto rozhraní by ovšem musela proběhnout důkladná analýza používaných příkazů, protože jednoduchá implementace by měla za následek obrovskou složitost dotazů pro načtení požadovaných položek a změnu metadat.

Dalšími možnostmi rozšíření je dopsání a bližší rozšíření modulů o další funkce. Také by byla možná optimalizace práce s ID3lib a případné napojení na knihovnu zlib, která je součástí projektu ID3lib a slouží pro zrychlení práce knihovny ID3lib. Více o knihovně zlib lze najít na [4]. Dopsaní modulů pro jiné typy metadat by mohlo být dalším rozšířením této práce.

Další možností pokračování projektu je jeho rozšíření na jiné platformy než je Microsoft Windows a vytvořit ukázkové nebo plně funkční aplikace fungující i mimo původní operační systém.

Také by se tento systém dal rozšířit i na jiné typy metadat než jsou souborová metadata, ale to by se již jednalo o zcela jiný postup a musela by mu předcházet důkladná analýza, zda lze vůbec nějaký takový dostatečně obecný systém vytvořit.

Literatura

[1] Exif neoficiální stránka, http://www.exif.org

[2] Exiv2, http://www.exiv2.org

[3] ID3 tag specifikace, http://www.id3.org

[4] ID3lib, http://www.id3lib.org

[5] MSDN Library for Visual Studio 2005, http://msdn2.microsoft.com