
Specifikace softwarového díla

&

Časový plán implementace

pro

Podpora skriptování v Audacity

Audacity je oblíbený editor zvuku, který ovšem v současné době postrádá možnost automatizovaného vykonávání skriptů. Cílem projektu je doplnit editor Audacity o rozhraní pro podporu skriptovacích jazyků a implementace skriptovací konzole pro jazyk Python.

1.0.1

Jan Jizbický

3. 1. 2011

Obsah

1. Základní informace	1
1.1 Popis a zaměření softwarového díla	1
1.2 Použité technologie.....	1
1.3 Konvence tohoto dokumentu	1
1.4 Odkazy (reference).....	1
2. Stručný popis softwarového díla	2
2.1 Důvod vzniku SW díla a jeho základní části a cíle řešení	2
2.2 Hlavní funkce.....	2
2.3 Motivační příklad užití.....	2
2.4 Prostředí aplikace.....	3
2.5 Omezení díla	3
3. Vnější rozhraní	3
3.1 Uživatelské rozhraní, vstupy a výstupy	3
3.2 Rozhraní software	3
4. Detailní popis funkcionality	4
4.1 Spouštění skriptovací konzole	4
4.2 Vytvoření skriptu	4
4.3 Načítání skriptů pro editaci.....	4
4.4 Ukládání editovaných nebo vytvořených skriptů.....	4
4.5 Zvýrazňování syntaxe v editoru	4
4.6 Vykonávání skriptů	4
5. Obrazovky	5
5.1 Dialogové okno konzole	5
6. Ostatní (mimofunkční) požadavky	6
7. Time-line & Milestones	6
8. Poznámky	7

Tabulka revizí

Jméno	Datum	Důvod změny	Verze
Jan Jizbický	20. 6. 2010	Počáteční text	0.1.0
Jan Jizbický	15. 9. 2010	Doplňen detailní popis funkcionality	1.0.0
D3S	3. 1. 2011	Úprava jako vzorový příklad specifikace	1.0.1

1. Základní informace

1.1 Popis a zaměření softwarového díla

Audacity je oblíbený editor zvukových souborů s vyspělým uživatelským rozhraním, které umožňuje komfortní práci. Slabou stránkou tohoto editoru je, že neumožňuje práci automatizovat vykonáním nazkoušené posloupnosti kroků na větším množství souborů. Podpora skriptovacích jazyků je často žádaná pokročilými uživateli graficky orientovaných aplikací, jak dokazuje například GIMP *script-fu* nebo jazyk Basic v kancelářských balících.

Popisovaný projekt si klade za cíl vytvoření skriptovací konzole a navrhnutí rozhraní mezi existujícím editorem a skriptovací konzolí, které bude umožňovat běžné činnosti dostupných pomocí GUI jako je otevírání souborů, vybírání jejich částí a aplikování filtrů. Pro ověření vlastností návrhu tohoto rozhraní bude implementována podpora pro spouštění skriptů v jazyce Python.

1.2 Použité technologie

- Audacity editor
- WxWidgets (knihovna na tvorbu GUI v Audacity)
- Python (jazyk pro vykonávání skriptů)

1.3 Konvence tohoto dokumentu

1.3.1 Externí odkazy

Odkazy na webové stránky jsou uváděny modrou barvou s podtržením.

Příklad:

- <http://audacity.sourceforge.net/>, Audacity development team, 2011

1.4 Odkazy (reference)

- Webové stránky Audacity:
 - <http://audacity.sourceforge.net/>, Audacity development team, 2011
- Webové stránky WxWidgets
 - <http://www.wxwidgets.org/>, The WxWidgets Team, 2011
 - WxWidgets jsou použity k implementaci grafického rozhraní Audacity
- Embedding a Python Interpreter Into Your Program, Alan J. Salmoni, The Python Papers Volume 3 Issue 2, staženo 3. 1. 2011 z adresy <http://ojs.pythonpapers.org/index.php/tpp/article/viewPDFInterstitial/26/58>

2. Stručný popis softwarového díla

2.1 Důvod vzniku SW díla a jeho základní části a cíle řešení

Důvodem vzniku je doplnění neexistující funkcionality programu Audacity, jak bylo popsáno v části 1.1.

Program Audacity neobsahuje žádné jasně definované API pro podporu skriptování, jeho návrh je mimo jiné součástí této práce.

Skriptovací konzole bude navržena jako zásuvný modul v podobě dynamicky linkované knihovny tak, aby byly nutné co nejmenší zásahy do stávajícího kódu editoru Audacity. Grafické rozhraní bude využívat stejnou knihovnu WxWidgets jako využívá celý projekt Audacity.

Program bude složen ze dvou hlavních součástí:

- Dialogové okno s editorem,
- Engine pro vykonávání skriptů.

Uživatel bude aplikaci ovládat ovládacími prvky v dialogovém okně, tato část bude spouštět engine pro vykonávání skriptu a zobrazovat jeho výstupy.

2.2 Hlavní funkce

Skriptovací konzole bude umožňovat:

- Editování, načítání a ukládání skriptů v editoru se zvýrazňováním syntaxe.
- Automatizaci akcí přístupných pomocí GUI Audacity. Této automatizace bude dosaženo spouštěním skriptů v jazyce Python.

2.3 Motivační příklad užití

Uživatel načte ze souboru a spustí skript, který provede následující akce na vybraném zvukovém souboru nebo všech zvukových souborech ve vybraném adresáři:

- Načte stopu
- Vybere část od 2 do 3 minuty
- Na vybrané část aplikuje filtr *Noise reduction*.
- Otevře soubor s názvem *add_me.mp3*
- Připojí tento soubor za editovanou stopu
- Uloží editovanou stopu
- Ohlásí úspěch

2.4 Prostředí aplikace

Konzole bude odladěna pro Audacity verze 1.2.6 s drobnými úpravami. I když bude implementována jako plugin, bude třeba použít vlastní kompilovanou verzi. Pro účely práce bude

odladěna pouze verze pro Windows, ale nebudou použity žádné Windows-specific konstrukce, které by bránily pozdějšímu rozšíření na ostatní platformy, které editor Audacity podporuje. Interpreter jazyka Python bude integrován jako součást konzole a tudíž nemusí být v systému nainstalován.

2.5 Omezení díla

Pro účely práce bude odladěna pouze verze pro Windows.

3. Vnější rozhraní

3.1 Uživatelské rozhraní, vstupy a výstupy

Konzole bude přebírat do co možná nejvyšší míry zvyklosti GUI programu Audacity.

Audacity konzole bude implementována jako dialogové okno aplikace, načítání, ukládání editování a spouštění skriptů bude umožněno pomocí ovládacích prvků tohoto dialogového okna.

Ve fázi spuštění skriptu jsou vstupem konzole:

- samotný skript
- audio soubory odkazované ve skriptu
- kontext právě otevřeného projektu Audacity

Výstupem bude:

- uložený zvukový soubor nebo soubory
- změněný kontext aktuálního projektu

3.2 Rozhraní software

3.2.1 Začlenění konzole do Audacity

Dialogové okno konzole, které zastřešuje celou funkcionalitu díla, dostane jako argument *handle* na aktuálně otevřený projekt programu Audacity, který umožňuje přistupovat ke kontextu programu a vykonávat jednotlivé akce, jako je například otevírání souborů a aplikování filtrů.

3.2.2 Začlenění interpreteru jazyka Python

Interpreteru bude předán *handle* na Audacity projekt a bude manipulovat s projektem pomocí mechanismů pro vkládání interpreteru do programů v jazyce C a C++, jak je popsáno v dokumentu „*Embedding a Python Interpreter Into Your Program*“, viz reference.

Způsob předání samotného textu skriptu mezi konzolí a interpreterem bude stanoveno během implementace.

4. Detailní popis funkcionality

4.1 Spouštění skriptovací konzole

Umožňuje vytvořit nový skript.

4.2 Vytvoření skriptu

Uživatel označí editační pole, aby získalo focus. Poté do něj vepíše text stejně jako v jiném textovém editoru.

4.3 Načítání skriptů pro editaci

Umožňuje načíst existující skript pro editaci. Uživatel z menu *Soubor* vybere položku *Načíst*, což vyvolá zobrazení dialogového okna pro výběr souboru z knihovny WxWidgets. Do editačního pole bude načten text souboru v kódování UTF-8. Text bude zobrazen.

Pokud soubor nejde z jakéhokoli důvodu otevřít, bude uživateli zobrazena chybová zpráva (standardní okno s chybou a popisem, z jakého důvodu se nepodařilo soubor otevřít).

4.4 Ukládání editovaných nebo vytvořených skriptů

Umožňuje uložení editovaného nebo nově vytvořeného skriptu. Uložení skriptu půjde vyvolat klávesovou zkratkou Ctrl-S (pokud již je známa cesta k souboru) nebo pomocí položek menu *Soubor*, *Uložit* a *Uložit jako* s funkcionalitou obvyklou z běžných textových editorů.

Pokud není cesta známa (nový skript nebo varianta *Uložit jako*), vyvolá akce zobrazení standardního dialogu pro výběr souboru k uložení z knihovny WxWidgets. Pokud soubor nejde z jakéhokoli důvodu uložit, bude uživateli zobrazena chybová zpráva (standardní okno s chybou a popisem, z jakého důvodu se nepodařilo soubor uložit).

4.5 Zvýrazňování syntaxe v editoru

Zajišťuje zvýraznění syntaxe pro usnadnění editace a usnadnění orientace v aktuálně otevřeném skriptu.

4.6 Vykonávání skriptů

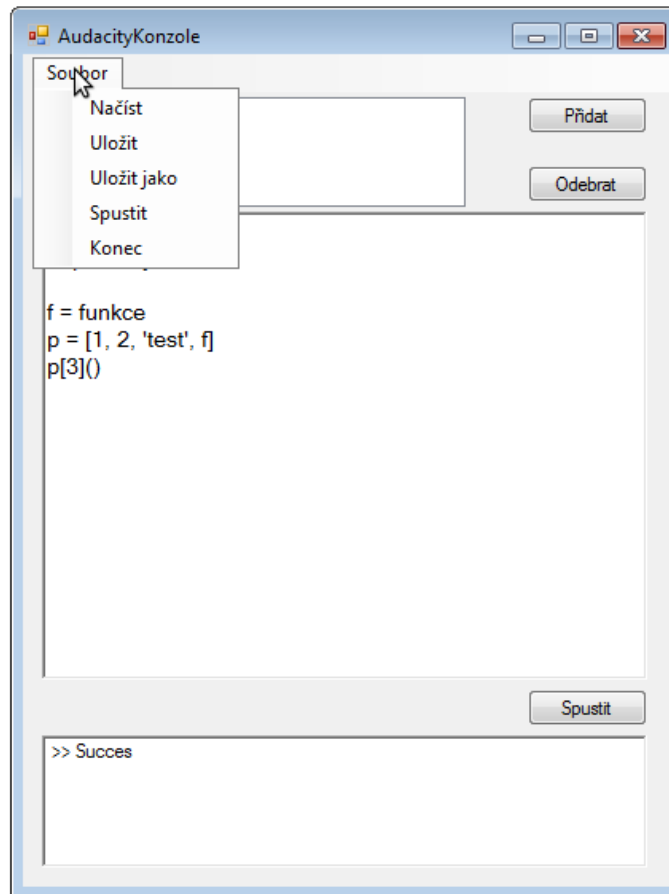
Umožňuje vykonávat vytvořené nebo načtené skripty

Spuštění skriptu je provedeno tlačítkem *Spustit* pod editovatelným polem, alternativně výběrem z menu *Soubor*, *Spustit*. Pro spuštění skriptu je právě načtený skript předán interpreteru, který ho vykoná. Prostředí, ve kterém bude skript vykonáván, bude poskytovat funkce pro manipulaci s prostředím projektu, načítáním a zapisováním stop.

Veškeré chyby a varování za běhu interpretru budou vypsány s popisem do výstupního textového pole. Pokud interpreter skončí jinak než úspěchem, bude uživateli zobrazeno standardní chybové okno.

5. Obrazovky

5.1 Dialogové okno konzole



Základní struktura dialogového okna konzole se skládá z menu, textového pole pro zobrazení editovaného skriptu a několika tlačítek.

Obrazovka ukazuje strukturu rozšíření aplikace:

- Menu – hlavně pro načítání, ukládání a spouštění skriptů-
- Horní část – obsahuje ovládací prvky pro přidání a odebrání souborů a adresářů, na kterých se má provést skript.
- Střední část (horní textové pole) – zobrazuje právě načtený skript, který může být editován nebo spuštěn tlačítkem *Spustit*.

- Spodní textové pole – zobrazuje výstup interpreteru skriptu.

6. Ostatní (mimofunkční) požadavky

Aplikace bude odladěna jen pro operační systém Windows 7, nicméně implementace nebude využívat nic, co by do budoucna bránilo snadnému využití na ostatních platformách, na které je portován program Audacity.

Do existujících zdrojových kódů bude zasahováno co možná nejméně, aby bylo možno tyto změny co nejnázve prosadit do hlavní vývojové větve programu. Dílo tedy bude psáno jako téměř izolovaný plugin.

7. Time-line & Milestones

Datum	Milník	Způsob prezentace
10/01/11	Finální verze této specifikace	Existující dokument.
10/02/11	Založení repository s kódem Audacity, rozchození překladu	Osobní prezentace
10/03/11	Přidání dialogového okna a vykonání modelového příkladu pomocí hardcoded volání (ne pomocí interpreteru)	Osobní prezentace
10/04/11	Návrh API	Dokumentace, konzultace
10/05/11	Napojení interpreteru. Možnost spouštění skriptu umožňující vybranou část funkcionality API	Osobní prezentace, kód v repository
10/06/11	Implementace ostatních částí funkcionality, případné úpravy API	Kód v repository, prezentace
10/08/11	Doladění do finální verze	Předvedení

Dodatek A: Vymezení pojmů

- Konzole – dialogové okno vyvolané z programu Audacity, které slouží pro ovládání celého díla

Dodatek B: To Be Determined List

- Přesný způsob předání textu skriptu z dialogového okna do interpreteru.

8. Poznámky

Tato specifikace je více než inspirována šablonami:

- Software Requirements Specification by by Karl E. Wiegers
- SAFE™ Development System Requirements