

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií



BAKALÁŘSKÁ PRÁCE

Liberec 2011

Ondřej Vacek

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B2646 – Informační technologie

Studijní obor: 1802R007 – Informační technologie

Webová aplikace pro rozpoznávání mluvčích

Web application for recognizing speakers

Bakalářská práce

Autor:

Ondřej Vacek

Vedoucí bakalářské práce:

Ing. Jan Silovský

Konzultant bakalářské práce:

Ing. Jindřich Žďánský, Ph.D.

V Liberci 10. 5. 2011

//nevkládat, volná strana místo zadání

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou 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 bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou 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.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum

Podpis

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Janu Silovskému za cenné rady, připomínky a metodické vedení práce.

Děkuji své rodině, partnerce a mým přátelům za psychickou podporu, čas a prostor, který mi během vytváření této bakalářské práce poskytli.

Abstrakt

Tato práce se zabývá tvorbou webové aplikace pro rozpoznávání mluvčích. Na začátku je krátce popsán způsob zpracování zvuku pro účely rozpoznávání mluvčích a následně jsou stručně vysvětleny metody používané při aplikaci těchto způsobů na reálné úlohy. Dále jsou zhodnoceny dnes nepoužívanější webové technologie, které přicházeli v úvahu pro tvorbu výsledné aplikace. Postupně jsou prozkoumány jejich výhody a nevýhody, díky kterým byla vybrána nejvhodnější z technologií splňující požadavky aplikace. Poté jsou rozebrány vlastnosti vývojových nástrojů, které byli použity při samotné tvorbě webové aplikace. V neposlední řadě je do detailů rozepsán vývoj výsledné aplikace, její architektury a jejích součástí.

Hlavním přínosem je webová aplikace, která umožňuje on-line rozpoznávání mluvčích. K tomu využívá software SRTK (speaker recognition toolkit), který obstarává výsledky rozpoznávání. Tento software byl vyvinut na Ústavu informačních technologií a elektroniky na Technické univerzitě v Liberci a je dílem Ing. Jana Silovského.

Klíčová slova: webové technologie, Adobe Flash, rozpoznávání mluvčích, verifikace, digitální audio

Abstract

This thesis deals with creating web applications for speaker recognition. At the beginning the way of sound processing for speaker recognition is shortly described and then the methods used in applying these ways to the problems of real world are explained. Next the most widely used web technologies that come into consideration for making the final application are reviewed. Their advantages and disadvantages are gradually being explored. This leads to choose the best technology, which meets the requirements of the application. The properties and characteristics of development tools, which were used in the actual creation of web application, are then analyzed. Finally, the development, architecture and the parts of the application itself are described to the details.

The main benefit is a web application that allows on-line speaker recognition. It uses software SRTK (speaker recognition toolkit), which performs the recognition results. This software was developed at the institute of Information Technology and Electronics at the Technical University of Liberec, and it's creator is Ing. Jan Silovský.

Keywords: web technologies, Adobe Flash, speaker recognition, verification, digital audio

Obsah

Prohlášení	3
Abstrakt.....	5
Abstract.....	6
Seznam ilustrací	8
1. Úvod.....	9
2. Algoritmy pro rozpoznávání mluvčích	11
2.1. Úvod do metod rozpoznávání	11
2.2. Trénování modelů.....	12
2.3. Verifikace	12
2.4. Identifikace.....	13
3. Software SRTK.....	13
4. Výběr technologie	14
4.1. Požadavky na aplikaci	14
4.2. Adobe Flash	14
4.3. Microsoft Silverlight.....	15
4.4. HTML5	16
4.5. Rozhodnutí	17
5. Vývojové nástroje	18
5.1. WampServer	18
5.1.1. Apache	18
5.1.2. MySQL.....	18
5.2. FlashDevelop.....	19
5.3. Tour de Flex.....	20
6. Návrh aplikace	21
6.1. Návrh databáze	21
6.2. Zabezpečení aplikace.....	23
6.3. PHP skripty.....	24
6.4. Aplikační rozhraní	25
6.4.1. Trénování.....	25
6.4.2. Rozpoznávání.....	27

6.4.3. Neaktivní funkce	30
6.5. Vzorová data	31
6.6. Architektura MVC.....	31
7. Závěr.....	32
7.1. Analýza technologií	32
7.2. Dosažené výsledky	33
7.3. Možnosti pokračování	33
Seznam použité literatury	34
Příloha č.1	35

Seznam ilustrací

Obrázek 1. ERD diagram databáze.....	21
Obrázek 2. Ukázka aplikace v režimu trénování mluvčích	26
Obrázek 3. Ukázka aplikace v režimu rozpoznávání	28
Obrázek 4. Ukázka aplikace s výsledky rozpoznávání.....	29

1. Úvod

Rozvoj počítačového zpracování řeči a použití lidského hlasu k ovládání elektronických zařízení se poslední dobou stává stejně populární, jako třeba dotykové displeje a bezdrátové snímače pohybu. Ale abychom mohli lépe a přesněji ovládat přístroje svým hlasem, je nutné nejen přesně rozeznat vyslovený příkaz, ale také určit, kdo ho vyslovil.

Tato práce se zabývá právě tou částí počítačového zpracování řeči, která se stará o rozpoznávání osob hovořících v určité nahrávce a dokáže tak například částečně zabezpečit přístroj proti ovládání nežádoucí osobou, nebo člověkem bez dostatečného oprávnění. K tomu slouží metody pro rozpoznávání mluvčích, které jsou stručně popsány v první části tohoto dokumentu.

Současné softwary umožňující rozpoznávání řeči jsou dostupné pouze v komerčních verzích a cena jedné základní licence se většinou pohybuje v řádu tisíců Korun českých. Navíc jsou velmi komplexní, umožňují spoustu nastavení a k jejich ovládání je nutné školení odborným pracovníkem, nebo nákup dalších tištěných dokumentů. Navíc jsou vždy omezeny pouze na jednu počítačovou platformu. Aplikace vytvořená v rámci této práce je přesným opakem stávajících softwarů. Její interface je navržen tak, aby ho dokázal ovládat i laik, který nemá potřebné znalosti v oboru. Díky webovým technologiím je navíc multiplatformní a její použití je zcela zdarma.

Na začátku práce bylo potřeba seznámit se s metodami, které se používají při počítačovém zpracování řeči. Tyto techniky jsou blíže popsány v publikaci [2] a jsou použity v algoritmech programu SRTK (speaker recognition toolkit), který byl vyvinut za účelem rozpoznávání mluvčích v Laboratořích počítačového zpracování řeči na TUL. Tento software byl jedním z klíčových prvků pro tuto práci, neboť právě jeho funkce jsou zde využívány.

Hlavním přínosem této práce bylo vytvoření webové aplikace umožňující snadné ovládání všech funkcí výše zmiňovaného softwaru SRTK, který bylo doposud možné ovládat pouze pomocí příkazů z příkazové řádky operačního systému MS Windows. Aplikace musela být dostupná, snadno ovladatelná, uživatelsky přívětivá a nezávislá na operačním systému, aby byla přístupná co největšímu počtu

potencionálních uživatelů, kteří nemají v povědomí procesy používané při rozpoznávání řeči a chtějí znát pouze výsledky.

Jako první bylo navrženo jednoduché a přehledné rozhraní pro budoucí aplikaci s ohledem na budoucí uživatele, kteří nemají zkušenosti a znalosti problematiky rozpoznávání mluvčích. Prostředí aplikace bylo rozděleno na dva samostatné módy podle použití, které jsou blíže popsány v kapitole Aplikace.

Aby aplikace splňovala i ostatní nároky, bylo nutné, aby byla aplikace dostupná z internetu. Proto bylo potřeba vybrat jednu ze současných moderních webových technologií, jako jsou Adobe Flash, Silverlight, HTML5 a jiné. Po analýze výhod a nevýhod jednotlivých platforem jejich možností byla zvolena ta, která nejvíce vyhovovala požadavkům zpracování.

Nejdelší čas na projektu zabralo samotné kódování ovládacích prvků a jejich algoritmů, propojení na databázi pro ukládání dat a správu uživatelů a samotné napojení ovládání na funkce softwaru SRTK.

Poslední, ale neméně důležitou částí úspěšného výsledku bylo vytvoření stručné nápovědy pro práci se vzniklou aplikací pro uživatele. Nápovědu je možno najít jednak jako součást aplikace a jednak jako přílohu této zprávy.

2. Algoritmy pro rozpoznávání mluvčích

Jelikož tato práce pouze využívá již hotového systému a implementace těchto postupů v softwaru SRTK, nebudou zde uváděny podrobné matematické výpočty a rovnice používané při rozpoznávání mluvčích, ale pouze základní rozdělení na jednotlivé kroky procesu. Tyto algoritmy jsou podrobněji popsány v [1] a metody počítačového zpracování řeči v [2].

2.1. Úvod do metod rozpoznávání

Lidská řeč vzniká vibracemi hlasivek ve kmitočtovém rozsahu přibližně od 200Hz do 4kHz. Experimenty, které byly provedeny v oblasti lidského sluchu, prokázaly, že lidské ucho nevnímá příchozí frekvence v lineární stupnici. Pro tyto účely byla sestavena Melovská stupnice, která přepočítává frekvenci zvuku podle vztahu:

$$Mel(f) = 2595 \cdot \log_{10} \left(1 + \frac{f}{700} \right) \quad (1)$$

kteřá ponechá frekvenci zvuku do 1kHz v lineární stupnici a v oblasti od 1kHz výš převede signál do logaritmické stupnice.

Takto transformovaný signál je následně upraven zvýrazněním vyšších frekvencí, které mívají obecně nižší amplitudu, a použitím Hammingova okénka se potlačí oříznutí na okrajích signálu. Následně je aplikována FFT pro amplitudové spektrum upravené Melovskou stupnicí.

Posledním krokem je výpočet příznaků MFCC. Příznaky MFCC jsou odvozeny z výsledného spektra a obsahují převážně informace o rezonančních frekvencích signálu. Právě rezonanční frekvence jsou unikátní pro konkrétního mluvčího. Příznaky MFCC jsou v dnešní době nejpoužívanější metodou pro rozpoznávání mluvčích, jak uvádí autor [1].

2.2. Trénování modelů

Trénování slouží k vytvoření unikátní charakteristiky mluvčích v podobě binárního souboru. Těmto souborům se říká modely a slouží k uchování jedinečných příznaků hlasu mluvčí osoby.

V trénovací nahrávce se nejdříve určí části s hlasem a části hluku. Frekvence hluku jsou následně potlačeny aby bylo trénování co nejefektivnější. Poté záleží na vybrané metodě a z nahrávky se vytvoří charakteristika řeči na základě jedinečných příznaků hlasu, jako jsou rezonanční frekvence.

Výstupem trénovacího procesu je model odpovídající hlasu jednoho konkrétního mluvčího uložený v podobě binárního souboru. Při opětovném trénování stejného modelu si pak můžeme obvykle zvolit, zda uložíme pouze nově natrénované hodnoty, nebo se sloučí nové hodnoty se starými.

U osob mladšího věku, zejména u mužů v pubertě, dochází ke změnám v charakteristice jejich hlasu. Tyto změny jistě ovlivňují tvar jejich modelu a doporučuje se proto jejich modely obnovovat s přibývajícím věkem. Jinak může docházet k častým chybám při rozpoznávání těchto osob.

2.3. Verifikace

Verifikace je proces rozpoznávání mluvčího v nahrávce oproti jeho modelu. S touhou úlohou se setkáváme nejčastěji jako součást zabezpečeného přístupu. Obvykle se kombinuje s nějakým identifikačním číslem nebo jiným biometrickým údajem.

Mluvčí tedy zadá svou totožnost a úkolem systému je rozhodnout na základě jeho hlasu, zda je opravdu ten, za kterého se vydává.

Rozhodovací proces probíhá podobě jako trénovací. Nejdříve se od sebe oddělí řeč a okolní šum. Poté je vytvořena charakteristika nasnímaného hlasu a porovnána s modelem mluvčího. Výsledkem je skóre, které určuje míru pravděpodobnosti zkoumaného mluvčího. Systém má pak nastavený určitý mezní práh a pokud je skóre nad touto hranicí, potvrdí pravost mluvčího. V opačné situaci zabráni přístupu, nebo oznámí správě systému, že nenastala shoda.

2.4. Identifikace

Proces identifikace se od verifikace liší pouze tím, že posuzovaná nahrávka není porovnávána proti jednomu modelu, ale oproti omezené množině více modelů. Úkolem systému je tedy určit hovořící osobu ze skupiny potencionálních mluvčích, nebo vyloučit všechny možné mluvčí a označit řečníka za neznámého.

Tato metoda se využívá například u anotace množiny zvukových nahrávek, kde ke každé z nich systém přiřadí mluvčí osobu. Postup se shoduje s verifikací, jen výsledkem je množina hodnot vyjadřujících skóre jednotlivých modelů. Jako výsledný mluvčí je určen vlastník modelu s nejvyšším skóre. Pokud však ani on nepřesáhne předem nastavenou mezní hranici, je mluvčí označen jako osoba neznámá.

3. Software SRTK

Tento program byl vytvořen na Ústavu Informačních technologií a elektroniky při Technické univerzitě v Liberci. Autorem je Ing. Jan Silovský, který tento systém navrhnul, naprogramoval a nadále jej rozšiřuje. Jeho název je zkratkou odvozenou z anglického „speaker recognition toolkit“, což ve volném překladu znamená „sada nástrojů pro rozpoznávání mluvčích“.

Software pracuje ve dvou různých módech. První mód je trénování mluvčích, kdy z trénovací nahrávky, což je audio soubor ve formátu *WAV*, vytvoří model mluvčího. To je binární soubor, který reprezentuje charakteristické rysy řeči. Druhý mód umožňuje rozpoznávání mluvčích, kdy je zkoumaná nahrávka porovnána proti modelu a výsledkem je míra pravděpodobnosti, že v nahrávce mluví zkoumaný mluvčí.

Počítá se s tím, že jeho aktuální verze není konečná a bude se rozšiřovat o další nové funkce. Program je spustitelný pod operačním systémem MS Windows a neobsahuje žádný grafický uživatelský interface. Ovládání programu je řešeno parametrizovanými příkazy předávanými spouštěcímu souboru z příkazové řádky. Webová aplikace, která byla vytvořena v rámci této bakalářské práce, mimo jiné vytváří onen zmiňovaný chybějící interface pro systém SRTK. Umožňuje tak jeho snazší a přehlednější ovládání a navíc je připravena na jeho další rozšiřování.

4. Výběr technologie

Aby byla výsledná aplikace co nejefektivnější a co nejvíce vyhovovala zadaným požadavkům, bylo nutné prozkoumat vlastnosti dnes nejpoužívanějších webových technologií a vyhodnotit, která z nich nejlépe vyhovuje našim potřebám.

4.1. Požadavky na aplikaci

Hlavním cílem aplikace bylo zpřístupnit metody rozpoznávání mluvčích pro širší, laickou veřejnost. Aplikace tedy musela být dostupná na všech dnes rozšířených počítačových platformách. Její snadné ovládání bylo umožněno díky přehlednému a ucelenému prostředí. Součástí aplikace je i stručná nápověda, která je přiložena k tomuto dokumentu.

Platformní nezávislost lze dnes snadno vytvořit díky celosvětově rozšířeným webovým technologiím, které jsou neoptimálnější pro tyto účely. Proto se výběr vhodných prostředků soustředil pouze na webové technologie. V úvahu přicházeli technologie *Flash* od společnosti Adobe, technologie *Silverlight* od společnosti Microsoft a nejmladší technologie *HTML5*. Jejich klady a zápory jsou zhodnoceny v následujících třech kapitolách.

4.2. Adobe Flash

Technologie Flash je nejstarší ze zmiňovaných technologií. Poprvé byl představen v roce 1996 americkou společností Macromedia a v roce 2005 celou tuto firmu s i Flasem koupila firma Adobe, která dnes zaštiťuje jeho vývoj. Používá se především pro tvorbu (převážně internetových) interaktivních animací, prezentací a her. Pro kódování vzhledu aplikace se používá jazyk XML rozšířený od Adobe o XMLNS (jmenný prostor definující vlastní značky) a pro aplikační logiku se využívá programovacího jazyka ActionScript, který byl vytvořen přímo pro Flash a jeho současná verze 3 je plně objektová. Tím je striktně oddělena aplikační logika od rozhraní aplikace a proto úpravy kódu v jedné části neovlivňují část druhou.

Výhody:

- + Nejrozšířenější technologie podporovaná na největším množství zařízení s přístupem na internet.
- + Rozsáhlá online knihovna s detailním popisem všech jednotlivých komponent a jejich funkcí.
- + Sjednocený líbivý vzhled komponent dodává dojem uceleného prostředí.

Nevýhody:

- Vysoký nárok na výkon, což u přenosných zařízení snižuje výdrž baterie.
- Bezpečnostní ochrana proti zneužití (tzv. SandBox) striktně omezuje aplikaci v práci se souborovým systémem a komunikací s databází.
- Programovací prostředí Flash Builder je drahé a freeware alternativy neumožňují vizuální programování komponent.

4.3. Microsoft Silverlight

Aplikační platforma Silverlight byla poprvé představena v roce 2007 firmou Microsoft jako konkurenční technologie pro Flash. První verze umožňovala programovat aplikace pouze v jazyce JavaScript. Od té doby prošel Silverlight značným vývojem a poslední verze číslo 5 je již objektová a umožňuje vybrat si z programovacích jazyků C#, VB.NET, JavaScript a IronPython.

Výhody:

- + Podpora a rychlý vývoj od společnosti Microsoft.
- + Volba programovacího jazyka umožní autorovi vybrat si podle svých zkušeností.
- + Programovací prostředí Visual Studio obsahuje spoustu pomůcek a návodů, navíc v rámci licence MSDNAA pro TUL je zdarma.

Nevýhody:

- Není podporován ve všech internetových prohlížečích.
- Některé funkce fungují pouze v OS Windows.
- Oproti Flash má stále méně komponentů a funkcí.

4.4. HTML5

Tato specifikace rozšiřuje jazyk HTML o nové, dynamické elementy, díky kterým je možné provádět interakci s uživatelem lépe, než doposud. Návrh těchto specifikací předložila v roce 2009 organizace W3C zabývající se definováním a správou webových standardů. Tvůrcům šlo především o to, doplnit do stávajících prvků, jakými jsou nadpisy, odkazy a tabulky, na klasické webové stránce elementy pro audio, video a komunikaci s uživatelem. Specifikace navíc umožňuje používat stránku v režimu off-line, kdy jsou všechny její prvky předem načteny na lokální disk uživatele. Také se organizaci W3C podařilo zjednodušit zápis některých stávajících elementů, takže se zdrojový kód webové stránky zpřehlední. Bohužel specifikace HTML5 ještě není kompletní a proto vývojářské týmy internetových prohlížečů jen velmi pozvolna implementují nové prvky do svých browserů.

Výhody:

- + Rychlost a nízký výpočetní výkon díky přímé podpoře internetového prohlížeče.
- + Platformní nezávislost zaručí samotný prohlížeč, který je pro všechny systémy stejný.
- + Snadné programování pouze pomocí jazyka HTML s podporou PHP.

Nevýhody:

- Standard nemá svou konečnou podobu a specifikace se proto můžou ještě měnit.
- Neexistuje ucelená dokumentace pro tvorbu webových aplikací s popisem funkcí.
- Podpora jednotlivých elementů se liší podle prohlížeče což způsobuje nestabilitu aplikace.

4.5. Rozhodnutí

Tyto tři zmiňované technologie jistě nejsou jediným řešením pro vývoj aplikace. Existuje řada dalších technologií a způsobů, jak dosáhnout stejného výsledku, jako jsou například jazyk PHP, JavaScript, AJAX a další. Avšak všechny tyto způsoby byly shledány za zbytečně komplikované, neúčinné nebo zdlouhavé a proto nebyly při hledání nejvhodnější technologie brány v potaz a jejich vlastnosti zde nejsou nijak rozepsány.

Po porovnání vlastností vybraných technologií bylo k jednotlivým z nich přiřazeno pomyslné skóre. Pro Flash hovořila jeho rozšířenost a také možnost aplikaci zkompileovat pomocí technologie Adobe Air jako desktopovou. Silverlight zase vynikal díky skvělé podpoře ze strany společnosti Microsoft a také svou velkou škálou nabízených programovacích jazyků. Technologie HTML5 dopadla nejhůře. I přes svou teoretickou rychlost aplikace je doposud velmi málo zdokumentovaná a její podpora nedosahuje takové úrovně jako u zbývajících dvou.

Jako vývojový nástroj pro budoucí aplikaci byla nakonec vybrána technologie Flash od společnosti Adobe. Tento nejrozšířenější způsob psaní webových aplikací zajistí snadnou dostupnost programu pro nejširší veřejnost a jeho přesné oddělení výpočetní logiky od rozhraní aplikace umožní budoucí rozšiřování jejích funkcí. Možnosti technologie Adobe Air mohou být později použity pro vytvoření off-line interface k softwaru SRTK. Historie i podrobné parametry této technologie jsou k dispozici v publikaci [3].

5. Vývojové nástroje

5.1. WampServer

Tento software [4] je souborem několika nejčastěji používaných programů pro chod webového serveru. Je zdarma pro všechny běžné platformy a první písmeno jeho názvu určuje první písmeno platformy, pro kterou je sada určena (Wamp – Windows, Mamp – MacOS apod.). V této práci posloužil jako testovací webový server pro přístup k vyvíjené aplikaci na adrese *localhost* (IP adresa 127.0.0.1). Verze použitého programu byla 2.1a.

5.1.1. Apache

Modul webového serveru, který se stará o komunikaci přes protokol *http*, popřípadě *https*. Obsahuje démona, což je proces naslouchající příchozí komunikaci na určitém portu. Pro protokol *http* je stanoven port 80, protokol *https* pak komunikuje přes port 443. Navrhovaný systém neposílá žádné důvěrné informace, které by musely být šifrovány. Proto si zde vystačíme s protokolem *http*. Apache také zajišťuje kompilaci kódu PHP pro dynamické prvky stránky a při příchozím dotazu odesílá data stránek klientovi.

Při testování aplikace bylo zjištěno, že dochází k chybám při nahrávání některých souborů na server. Tato zdánlivá závada byla způsobena implicitním nastavením maximální velikosti souboru na 1,5MB. Tato velikost je pro účely aplikace malá a proto je nutné její limit zvýšit na 10MB, které povoluje samotná aplikace.

5.1.2. MySQL

Databázová aplikace MySQL balíčku Wamp slouží převážně k ukládání dat souvisejících s během webové aplikace. Data mají strukturu tabulek, kde každý řádek interpretuje jednu položku a jeden sloupec zastupuje vlastnost této položky. Do databází se nejčastěji ukládají informace o uživateli a s nimi související data a nastavení příslušné aplikace. Pro komunikaci s touto databází se používá jazyka SQL, nejčastěji v kombinaci s PHP skripty. Pro správu databází a tabulek slouží webové rozhraní phpMyAdmin, což je volně šiřitelná webová aplikace, která je rovněž součástí balíčku Wamp.

5.2. FlashDevelop

Jak již bylo zmíněno v kapitole o webové technologii Flash, nejlevnější licence vývojového prostředí Flash Builder od jeho mateřské firmy Adobe stojí přes pět tisíc Kč. Proto bylo pro vývoj aplikace použito alternativní vývojové prostředí s názvem FlashDevelop [5]. Tento software je zdarma pro nekomerční použití a pro kompilaci zdrojového kódu využívá Flex SDK (Software Development Kit – sada vývojových nástrojů). Toto SDK obsahuje sady knihoven pro jednotlivé komponenty a funkce. Mezi hlavní knihovny patří knihovna FX, která definuje některé složitější datové typy a obsahuje statické funkce pro práci nad daty. Dalšími použitými knihovnami jsou pak knihovny MX a Spark, které obsahují vizuální komponenty pro interakci s uživatelem, jako jsou tlačítka, textová pole, rolovací menu atd. Rozdíl mezi těmito dvěma knihovnami je takový, že Spark se objevila až v poslední verzi SDK a na rozdíl od knihovny MX zajišťuje stejný grafický vzhled všech komponent, což dodává aplikaci ucelenou formu. Proto také byla v cílové aplikaci převážně použita knihovna Spark.

Nevýhodou programu FlashDevelop je oproti komerčnímu vývojovému prostředí Flash Builder od firmy Adobe to, že není WYSIWYG. Tato zkratka pochází z anglického „*What You See Is What You Get*“, což v překladu zní „co vidíš, to dostaneš“. V praxi to pak znamená, že vzhled výstupního produktu této aplikace je viditelný už při jeho tvorbě. Například v programu Microsoft Word je již od začátku psaní textu patrné, jak bude vypadat výsledný vytištěný dokument. V programování to pak znamená, že grafická část aplikace se netvoří pomocí jejího zdrojového kódu, ale vzniká umístěním jednotlivých komponent na pomyslné pozadí aplikace. Vývojové prostředí poté vytvoří zdrojový kód za vás. Tento způsob značně urychluje tvorbu grafické části aplikace, avšak takovéto softwary bývají obvykle velmi drahé. FlashDevelop bohužel není WYSIWYG aplikací, takže interface byl napsán pomocí zdrojového kódu.

Jak bylo zmíněno už v kapitole Flash, pro kódování se používají dva programovací jazyky. Pro statický vzhled aplikace je to jazyk XLMNS, který (jak název vypovídá) pochází z programovacího jazyka XML a další dvě písmena NS znamenají anglické „*Name Space*“, neboli vlastní jmenný prostor. To znamená, že jsou definované vlastní XML značky, které jsou obsaženy v knihovnách SDK a zastupují jednotlivé prvky v aplikaci. Každá taková značka je reprezentována jako objekt a je možno ji

přiřadit vlastnost *id*, díky které je pak adresovatelná v aplikační logice programu. Pro aplikační logiku je použit programovací jazyk ActionScript, který byl vyvinut firmou Macromedia a později převzala vývoj firma Adobe. Jeho syntaxe se podobá programovacímu jazyku Java. Je plně objektový a to i včetně všech základních datových typů.

FlashDevelop by se dal popsat jako speciálně upravený textový editor pro psaní aplikace pomocí technologie Flash. Jelikož se často vyskytují současně části kódu v jazyce XMLNS a ActionScript v jednom souboru, editor automaticky přepíná dvě různá zvýraznění zdrojového kódu podle toho, ve kterém z nich se zrovna nachází kurzor. Podle odkazovaných knihoven v hlavičce kódu aplikace (část *import*) pak při psaní objektů automaticky navrhne všechny vlastnosti a metody dané třídy s krátkým popisem. To napomáhá k rychlejšímu vývoji aplikace. V neposlední řadě dovoluje aplikaci ladit (tzn. krokovat její postup ve zdrojovém kódu za běhu) a sledovat obsah vnitřních proměnných. K tomu slouží vestavěný Flash přehrávač, takže není možné ladit aplikaci přímo ve webovém prohlížeči.

5.3. Tour de Flex

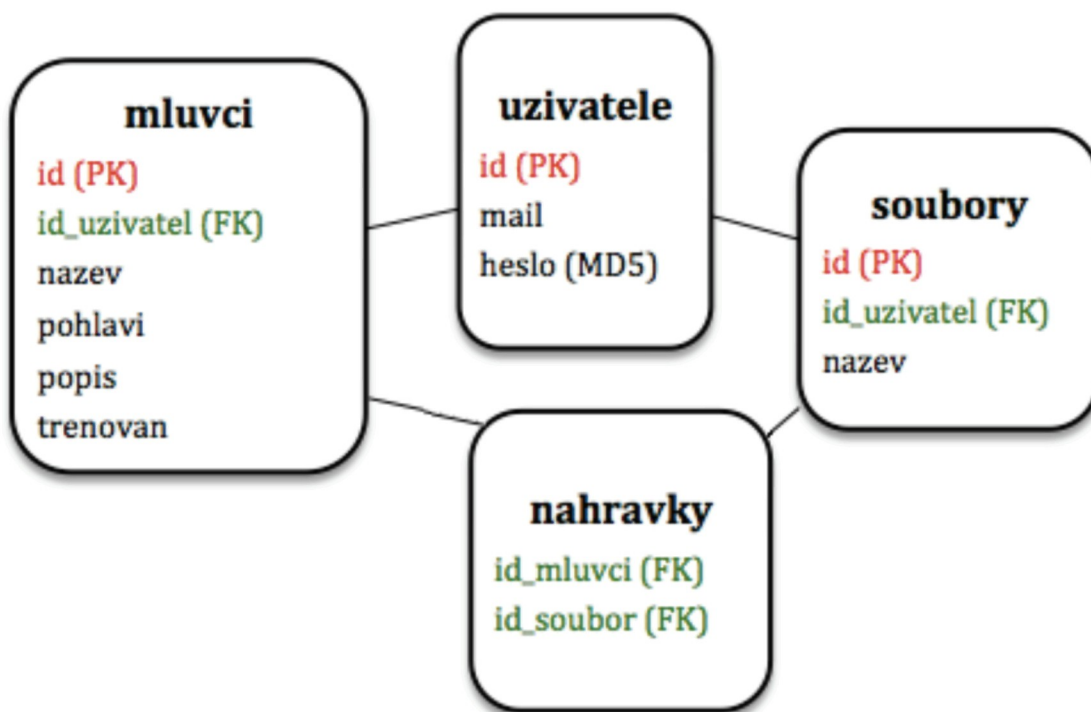
Vývojářský tým firmy Adobe zabývající se technologií Flash vytvořil velice přehlednou a užitečnou aplikaci Tour de Flex. Je celá naprogramovaná pro Flash a je dostupná jak on-line, tak díky technologii Air jako off-line aplikace. Obsahuje přes šest stovek různých ukázek použití komponentů z Flex SDK a jejich vlastností. Aplikace slouží jako výukový program a na každou dostupnou komponentu nalezneme minimálně jeden příklad jejího použití ve webové aplikaci. Pod příkladem se nachází jeho zdrojový kód, aby bylo patrné, jak se komponenta používá. Díky těmto tutoriálům je možné dosáhnout většího potenciálu vlastní aplikace a objevit skryté vlastnosti a funkce, které mohou ulehčit vývoj. Tato aplikace spolu s knihou [3] velmi napomohla při tvorbě webové aplikace.

6. Návrh aplikace

Hlavním cílem této bakalářské práce byla webová aplikace. V předchozích kapitolách byl odůvodněn výběr technologie pro její tvorbu a popsány nástroje, které byly při jejím vytváření použity, nebo jsou nezbytné pro její chod. V následující kapitole budou podrobně rozebrány jednotlivé části aplikace, ze kterých se skládá.

6.1. Návrh databáze

Jak již vyplynulo z kapitoly **WampServer**, jako databázový systém byla vybrána aplikace MySQL. Její výhodou je nenáročná instalace, snadná přenositelnost a hlavně cena, jelikož její licence je zdarma. Databáze byla navržena s ohledem na rychlost a efektivitu v 3NF takto:



Obrázek 1. ERD diagram databáze

Tabulka **uzivatele** nese informace o registrovaných uživateli. Z bezpečnostních důvodů není možné používat aplikace bez registrace. Ta vám zajistí, že vaše nahrávky ani vaši mluvčí nebudou používat jiní lidé. Jedinou výjimkou je uživatel s $id = '1'$. Soubory a mluvčí tohoto uživatele se zobrazují všem ostatním jako příkladová data a ostatní je nemůžou mazat. Sloupec *id* je zde privátním klíčem (anglicky Primary Key) a používá se pro ověření všech operací v databázi. Sloupec *mail*

slouží jako unikátní přihlašovací jméno uživatele a sloupec *heslo* je zamaskován hašovací funkcí (jednosměrné šifrování) MD5, aby nemohlo dojít k jeho odcizení. Když uživatel zadá svoje přihlašovací údaje, najde se jeho e-mail, ze zadaného hesla se vytvoří MD5 hash a ten se porovná s tím, co je uložený v databázi. Pokud se shodují, je uživatel přihlášen.

Tabulka **mluvci** obsahuje informace o všech mluvčích registrovaných uživatelů. Pokud si uživatel vytvoří nového mluvčího, jeho data jsou uložena sem. Mluvčího lze kdykoliv v aplikaci vymazat. Sloupec *id* slouží jako unikátní identifikátor mluvčího a je zároveň i primárním klíčem. Sloupec *id_uzivatel* odkazuje na uživatele, který tohoto mluvčího vlastní. Pouze tento uživatel má oprávnění mluvčího trénovat a smazat. Ve sloupcích *nazev*, *pohlavi* a *popis* jsou uložena informativní data o příslušném mluvčím a nejsou nijak vázaná na běh programu. Slouží pouze jako bližší identifikace této osoby. Sloupec *trenovan* nabývá hodnotu 0 nebo 1 a určuje, zda byl na mluvčího použitý proces trénování, neboli zda existuje datový soubor s modelem tohoto mluvčího. Tento model není součástí databáze, ale nachází se v osobních souborech každého uživatele. Pokud je hodnota 0, nemůže být mluvčí použit při rozpoznávání.

Tabulka **soubory** uchovává informace o všech zvukových souborech nahraných na server. Počet souborů není limitován a velikost jednoho souboru je limitována na 10MB. Sloupec *id* opět jednoznačně určuje soubor a je primárním klíčem. Sloupec *id_uzivatel* určuje uživatele, který tento soubor nahrál. Tento uživatel má také jako jediný právo soubor smazat. Sloupec *nazev* obsahuje název nahraného souboru. Při nahrávání souborů na server je původní název souboru zachován a tak je tento údaj důležitý nejen pro zobrazení jejich seznamu uživateli, ale také pro samotné načítání souboru. Díky hierarchickému uložení souborů na serveru do složek uživatelů mohou mít dva různí uživatelé nahraný jiný soubor pod stejným názvem.

Tabulka **nahravky** určuje, které souboru mohou být použity jako trénovací nahrávky pro mluvčí. Sloupec *id_mluvci* odkazuje na mluvčího, který může být trénován, sloupec *id_soubor* odkazuje na soubor, kterým bude trénován. Primární klíč v této tabulce tvoří kombinace obou dvou sloupců. Uživateli je povoleno určit více souborů jako trénovacích pro jednoho mluvčího a také zadat jeden soubor jako trénovací pro více mluvčích. Pokud uživatel smaže soubor nebo mluvčího, jsou automaticky smazány i řádky s jeho id.

6.2. Zabezpečení aplikace

Aby mohla být webová aplikace (nejen ve Flashi) dostupná z internetu, potřebuje být umístěná na internetové stránce. Pro účely vyvíjeného softwaru byla vytvořena strohá jednoduchá webová stránka v jazyce HTML a PHP, která zpřístupňuje aplikaci přes internet. Její vzhled je velmi strohý. Důraz zde nebyl kladen na grafickou stránku, ale na funkčnost. Stránky mají pouze uvítat návštěvníka, zajistit jeho registraci nebo přihlášení a zobrazit webovou aplikaci ve Flashi. Rozličnější grafické kreace by sloužily spíše jako rušivý element.

Na úvodní stránce je umístěn průvodní text, který vítá uživatele a pobízí ho k registraci, nebo přihlášení, pokud již registrován je. Uživatel se zde také dočte upozornění, že výsledky rozpoznávání mluvčích mají pouze informativní hodnotu a nelze je předkládat například jako důkaz. Toto upozornění nemá být chápáno tak, že systém pracuje chybně, ale je to spíše právní ochrana proti nařčení z nepravdivosti výsledků. Dále je na stránce umístěn přihlašovací formulář pro registrované uživatele a pod ním odkaz na novou registraci.

Registrační formulář vyžaduje po uživateli vyplnit jeho e-mailovou adresu, dvakrát heslo pro kontrolu překlepů a kontrolní kód, který zaručí, že uživatel není robot. E-mail je vyplňován pouze pro jednoznačnou identifikaci uživatele a není nijak jinak využíván, ani dále šířen. Mnohé internetové průzkumy ukázaly, že uživatelé upřednostňují registraci pod svou e-mailovou adresou, protože je menší riziko, že ji zapomenou. Heslo je zakódováno pomocí standardní hašovací funkce MD5. Tato funkce je jednosměrná šifra a její rozkódování je velice obtížné. V případě úniku dat z databáze proto nehrozí prozrazení tohoto hesla.

Kontrolní kód (anglicky *captcha*) se v dnešní době stal samozřejmostí všech registračních formulářů. Zde bylo naprogramováno jednoduché zobrazení čtyř znaků se stínem. Jeho úkolem je rozlišit počítačový program (neboli robota) od živé bytosti. Funguje na principu toho, že člověk je na rozdíl od robota schopný z obrázku přečíst a opsat několik znaků. Počítačový piráti stále zdokonalují své roboty ve čtení těchto kódů a proto se neustále vyvíjejí nové způsoby zobrazování těchto kódů tak, aby je robot nepřečetl, ale člověk ano. Registrace robota je nežádoucí, protože vždy slouží k napadení aplikace.

Po úspěšné registraci a přihlášení se spustí samotná aplikace a načte data uživatele z databáze. Kromě ovládacích prvků aplikace je k dispozici ještě miniaturní menu vpravo nahoře, které nabízí možnost odhlásit se, nebo otevřít nápovědu. Ta je díky jazyku JavaScript otevřena v novém, menším okně, takže ji uživatel může využívat během práce s aplikací. Na začátku nápovědy je menu s funkčními odkazy na jednotlivé kapitoly pro lepší orientaci.

6.3. PHP skripty

V rámci počítačové bezpečnosti byl stanoven bezpečnostní mechanismus s názvem SandBox pro aplikace postavených na technologiích, jako je Flash. Tento mechanismus striktně odděluje procesy této aplikace od ostatních procesů systému a přiřazuje jim mnohem menší práva. Kvůli tomu je nutné používat při komunikaci s ostatním softwarem, jako je MySQL databáze, nebo systém SRTK, nějakého prostředníka, který takováto práva má. Ve webových aplikacích se nejčastěji používá jazyk PHP a v něm psané skripty, neboť webový server, na kterém aplikace běží, umí tyto skripty zpracovávat. Za skript je označován malý program, který nepracuje samostatně, ale je určen pro potřeby větší aplikace. V případě jazyka PHP jsou data předávána skriptu pomocí metod POST nebo GET (stejně jako u jazyka HTML) a výsledky se vracejí v podobě textového výstupu skriptu.

Každý jednotlivý úkon aplikace obstarává samostatný skript. Pokud je vyžadována komunikace směrem ven, zavolá se příslušný skript a v parametrech volání se předají potřebné informace metodou POST. Tato metoda je bezpečnější proti podvržení falešných dat. Skript se volá stejným způsobem, jakým se načítá internetová stránka. Poté co se na základě předaných parametrů vykoná, výsledek se vrátí jako zdrojový kód stránky a aplikace jej zpracuje.

Celkem bylo napsáno 12 různých skriptů, které buď komunikují s databází, nebo posílají příkazy softwaru SRTK. Databázové skripty posílají dotazy pomocí jazyka SQL a starají se o přidávání, čtení a mazání mluvčích a souborů. Skripty pro SRTK pak zahajují procesy trénování a rozpoznávání. Při volání každého skriptu je aplikaci poslán spolu s parametry speciální řetězec *session id*, který je generován pro každého uživatele zvlášť. Skript pak ověří, jestli se jedná o akci přihlášeného uživatele a jestli na takovou akci má daný uživatel právo. Jedná se o bezpečnostní mechanismus proti zneužití skriptů.

6.4. Aplikační rozhraní

Cílem aplikace bylo zjednodušit uživateli práci se systémem SRTK. Proto bylo jeho rozhraní navrženo s ohledem na přehlednost a jednoznačnost ovládacích prvků. Jelikož systém SRTK umožňuje spuštění ve dvou různých módech, bylo rozhraní aplikace také rozděleno na dva aplikační módy. Prvním je trénování modelů mluvčích a druhý zpřístupňuje rozpoznávání mluvčích v nahrávce.

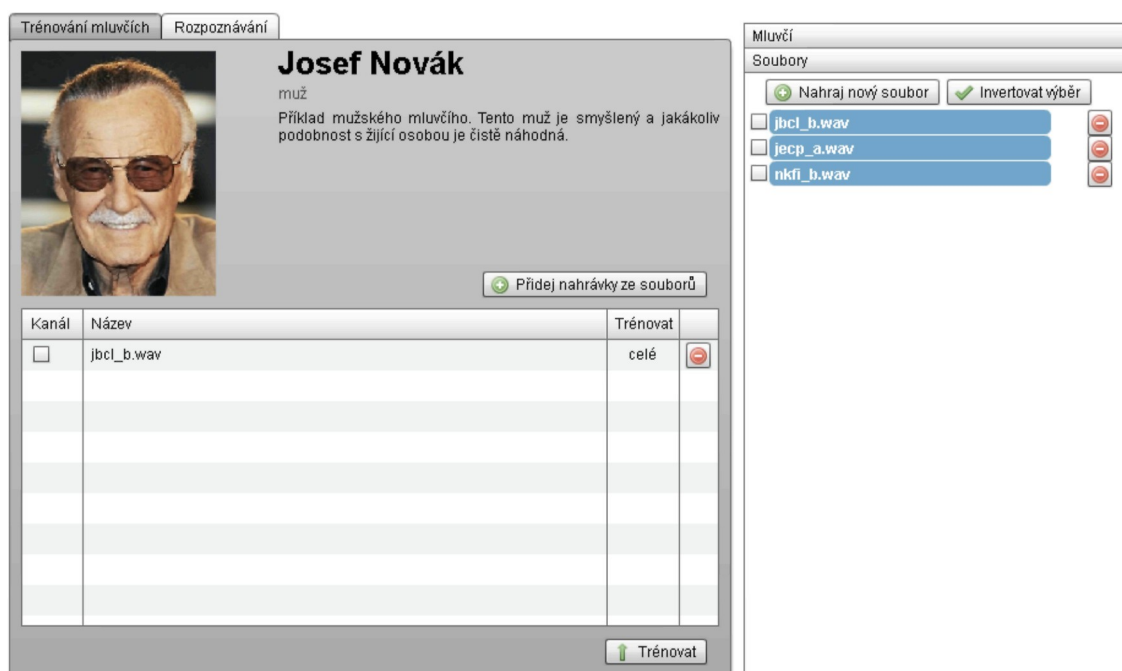
Protože uživatel potřebuje v obou módech jak přístup ke svým souborům, tak přístup k mluvčím, byl vytvořen speciální panel, který je dostupný v obou módech a ve kterém se nacházejí oba seznamy mluvčích a dat. Podle módu, ve kterém se uživatel zrovna nachází se v seznamu zobrazují kontrolní prvky potřebné pro práci s jeho položkami.

6.4.1. Trénování

Jak již bylo zmíněno na začátku této zprávy, trénování mluvčích je proces, kdy je vytvářen jeho model, což je binární soubor obsahující charakteristiky jeho řeči. Jako první věc, kterou musí nově přichozí uživatel udělat, je vytvoření svého prvního mluvčího. Pod horní lištou seznamu mluvčích se nachází tlačítko „Nový mluvčí“, po jehož stisknutí se objeví dialogové okno. Zde musí uživatel vyplnit pole *jméno* a *příjmení* (stačí jedno z nich) a dále může nastavit mluvčímu jeho pohlaví, obrázek a připsat popis. Tyto informace slouží pouze uživateli a na trénování nebo rozpoznávání nemají vliv. Po potvrzení údajů tlačítkem „Uložit“ se nový mluvčí uloží do databáze a objeví se v seznamu.

Nyní si uživatel může zobrazit jeho detaily kliknutím na jeho jméno v seznamu. Objeví se vyplněné informace a prázdná tabulka trénovacích souborů. V pravém horním rohu svítí červený křížek, který symbolizuje, že mluvčí ještě nebyl trénován a tudíž nemůže být použit pro rozpoznávání. Pravý panel se automaticky přepne na seznam souborů. V něm může uživatel nahrát nové soubory na server. Tuto akci může provést kdykoliv za běhu aplikace a velikost souboru je limitována na 10MB, ale je možné tento limit ve zdrojovém kódu aplikace upravit. Uživatel zaškrtně trénovací soubory v seznamu a stiskne tlačítko „Přidej nahrávky ze souborů“. Soubory se uloží do databáze jako trénovací nahrávky k vybranému mluvčímu a zobrazí se v tabulce detailu mluvčího.

Tabulka trénovacích nahrávek mluvího obsahuje čtyři sloupce. V prvním se nachází výběr kanálu, který bude použit při trénování. Protože software SRTK umožňuje na stereo nahrávce trénování pouze z jednoho kanálu, systém dá uživateli na výběr který to bude a nedovolí mu vybrat oba kanály najednou. Pokud je nahrávka monofonní, je zde pouze jedno výběrové políčko. Uživatel tedy zaškrtně, ze kterých nahrávek a popřípadě kanálů chce mluvího natrénovat. Druhý sloupec nese jméno zdrojového souboru pro snadnou identifikaci. Třetí sloupec slouží pro výběr úseků v nahrávce, ale tato funkce je zatím neaktivní. Je podrobněji popsána v kapitole 6.4.3 Neaktivní funkce. Poslední sloupec slouží ke smazání trénovací nahrávky. Smaže se ale pouze spojení s mluvicím a samotný soubor zůstane v seznamu souborů dostupný.



Obrázek 2. Ukázka aplikace v režimu trénování mluvího

Samotné trénování zahájí uživatel tlačítkem „Trénovat“, které se nachází pod tabulkou trénovacích nahrávek. Aplikace na dobu trénování zešedne, stane se neaktivní a objeví se dialogové okno, které upozorňuje na probíhající trénování. Doba trénování je závislá na počtu a délce trénovacích nahrávek a může se lišit podle množství řeči v nahrávce. Při testování aplikace se tato doba přibližně rovnala $\frac{1}{100}$ délky trénovacích nahrávek. Po dokončení trénování je uživatel upozorněn novým dialogovým oknem a aplikace se stane opět aktivní.

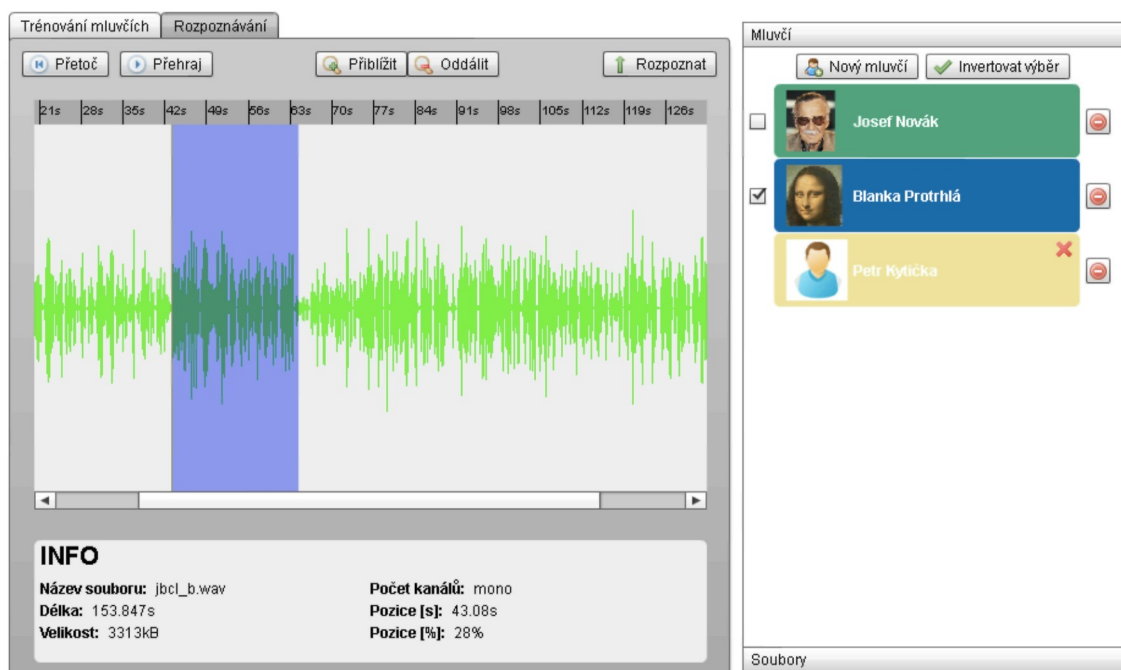
6.4.2. Rozpoznávání

Rozpoznávání mluvčích je metoda, která se snaží na dané nahrávce zjistit pravděpodobnost, s jakou na ní mluví porovnávaný mluvčí. Pro přepnutí do aplikačního módu rozpoznávání slouží záložka vlevo nahoře. Po kliknutí na tuto záložku se v hlavní části objeví zatím neaktivní přehrávač a pravý panel se přepne na seznam souborů. U každého souboru se navíc objeví ikonka pro načtení tohoto souboru do přehrávače.

Přehrávač zvukových souborů byl jedním ze stěžejích prvků aplikace a také obsahoval největší množství problémů, které bylo nutné překonat. Musel obsahovat nejen základní ovládací prvky, jakými jsou tlačítka „Přehraj“ a „Zastav“, ale v požadavcích bylo i zobrazení zvukové stopy, možnost přiblížení stopy a výběr daného úseku. První problém nastal hned při načítání zvukového souboru. Systém SRTK pracuje se zvukem v nekomprimované podobě ve formátu *WAV*. Flash bohužel pracuje se svým vlastním komprimovaným audio formátem nazvaným *SWF audio*, které podporuje pouze vzorkovací frekvence 5.5 kHz, 11 kHz, 22 kHz a 44 kHz. Navíc umožňuje implicitně načítat pouze audio ve formátu *MP3*.

První problém s nekompatibilitou zvukového formátu byl vyřešen použitím externí knihovny *ru.etcmedia.WaveSound.as*, která je volně dostupná z [7]. Knihovna nahrazuje standardní načítání zvuků a tím umožňuje použití souborů formátu *WAV*. Bohužel i tato knihovna je ale limitována vzorkovacími frekvencemi formátu *SWF audio*. Proto byl k aplikaci připojen program SOX v14.3.2 [8], který umožňuje rychlý převod zvukových formátů přes příkazový řádek. Před tím, než se audio soubor nahraje do přehrávače, je PHP skriptem překódován na vzorkovací frekvenci 11 kHz, aby mohl být zobrazen a přehrán. Změna vzorkovací frekvence se vztahuje pouze na data, která se dostávají do přehrávače. Pro rozpoznávání a trénování je používán originální nezměněný soubor.

Pro zobrazení souboru v přehrávači slouží tlačítko „Načíst tento soubor“ se zelenou šipkou směřující nahoru. Toto tlačítko se nachází vedle každého názvu v seznamu souborů. Jelikož se musí soubor stáhnout ze serveru, rychlost nahrávání se odvíjí od velikosti souboru a datové propustnosti internetu uživatele. Po dokončení stahování jsou aktivovány ovládací prvky přehrávače a zobrazena zvuková stopa nahrávky. Pod přehrávačem jsou informace o souboru. Bohužel nejsou kompletní, protože některé údaje jsou zkreslené převodem vzorkovací frekvence.



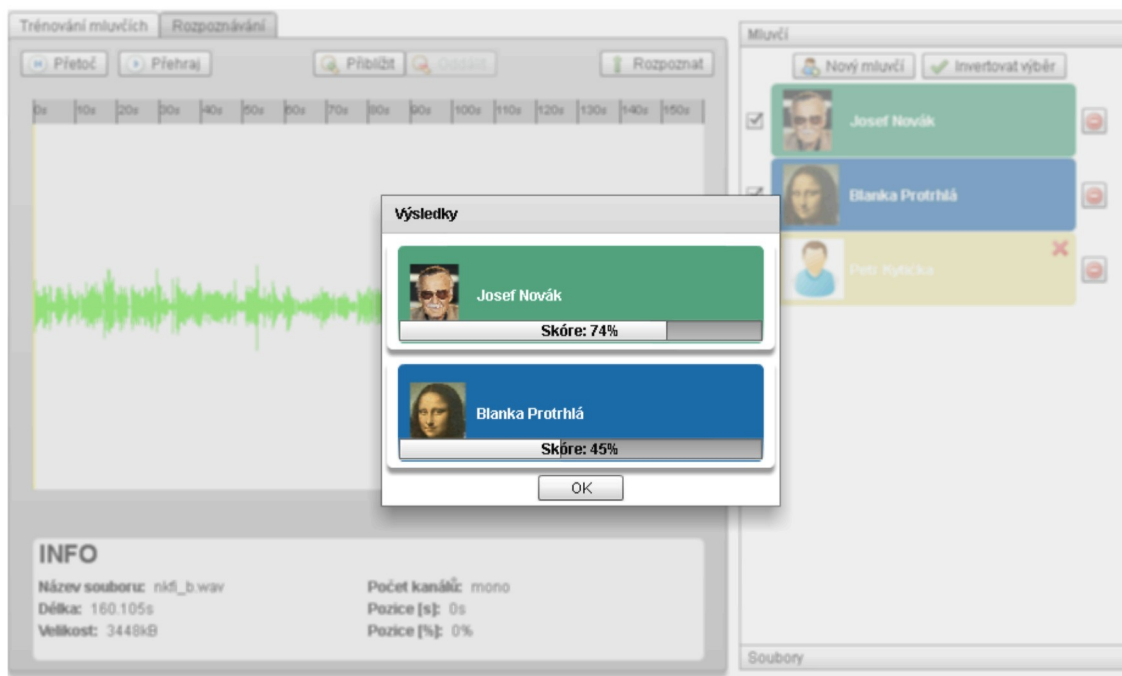
Obrázek 3. Ukázka aplikace v režimu rozpoznávání

Přehrávač umožňuje vedle spuštění ukázky a skoku na začátek také přiblížit a oddálit zvukovou stopu. Maximální oddálení je určeno šířkou okna, do kterého se zobrazí celá nahrávka. Jeden krok přiblížení je o 10% a maximální přiblížení je 10x. Celkem tedy lze přiblížit zvukovou stopu o 100%. Nad zvukovou stopou je umístěna časová osa, která mění své rozlišení dynamicky podle délky nahrávky a stupně přiblížení. Nejmenší rozlišení osy je 1s. Kurzor označuje právě přehrávané místo v nahrávce a je znázorněn žlutou svislou čarou. Kliknutím na zvukovou stopu přemístí uživatel pozici kurzoru na místo kliku. Pozice kurzoru je pak i textově uvedena v informacích pod přehrávačem a to jak v sekundách, tak v procentech (100% je délka nahrávky). Přetažením stisknuté myši přes zvukovou stopu pak může uživatel provést výběr, jak je to patrné z obrázku 3. Při výběru se kurzor přesune na jeho začátek a po stisknutí tlačítka „Přehraj“ se zastaví na konci výběru. Výběr lze zrušit kliknutím na zvukovou stopu mimo něj.

Po načtení nahrávky se také pravý panel přepne na seznam mluvcích. Pokud byl mluvcí trénován a je k dispozici jeho model, objeví se vedle jeho jména zaškrťací pole pro jeho vybrání na rozpoznávání. Mluvcí kteří nebyli trénováni nemohou být zaškrtnuti a navíc je u jejich jména červený křížek upozorňující na chybějící model.

Pro samotné zahájení procesu rozpoznávání musí mít uživatel načtený soubor v přehrávači a musí být vybrán alespoň jeden mluvčí. Poté uživatel klikne na tlačítko rozpoznat a data se odešlou softwaru SRTK na zpracování. Při procesu rozpoznávání systém SRTK neinformuje o probíhajícím stavu. Aby bylo možné uživateli alespoň přiblížit informace o tom, jak probíhá jeho rozpoznávání, jsou požadavky na tuto akci posílány po jednom mluvčím. Vzhledem k softwaru SRTK se sice jedná o drobné zdržení v řádu desetin sekundy, neboť musí provádět některé úkony opakovaně, ale v celkovém výsledku toto zdržení nehraje žádnou roli.

Během rozpoznávání je aplikace opět neaktivní a v popředí je umístěno jedno dialogové okno. To obsahuje pouze informaci o tom, že probíhá rozpoznávání a pod ní je umístěn ukazatel, který má jako maximum nastaven počet rozpoznávaných uživatelů. Po každém rozpoznávaném mluvčím se ukazatel posune a tak má uživatel alespoň nějakou informaci o probíhající činnosti. Když proces doběhne, dialogové okno s ukazatelem zmizí a objeví se nové okno s výsledky. Ty jsou seřazeny sestupně podle tzv. Skóre, neboli procentuální míry pravděpodobnosti, že v nahrávce mluví právě tento mluvčí. Ve výsledcích jsou zobrazeni pouze 3 mluvčí s nejvyšším skóre. Pokud se rozpoznávání nepovede, mají všichni rozpoznávaní mluvčí skóre 0. To se stává například při analýze syntetického zvuku bez řečových signálů.



Obrázek 4. Ukázka aplikace s výsledky rozpoznávání

Výsledky rozpoznávání jsou pouze informativní a záleží na uživateli, jak s nimi naloží. Rozpoznávání se dá bez omezení opakovat a na stejných datech vždy dojde ke stejným výsledkům. Takové metodě se říká sekvenční. Výsledky se nijak neukládají, takže pro uchování je nutné je opsat, vykopírovat nebo vytisknout.

6.4.3. Neaktivní funkce

Protože je software SRTK stále ve vývoji, byli už při vytváření webové aplikace známy některé funkce, kterými by mohl být v budoucnu rozšířen. Proto byla podpora těchto funkcí implementována do webové aplikace, ale zatím zůstává neaktivní. Zde je výčet některých z neaktivních funkcí:

- *Výběr úseku trénovací nahrávky* – pokud uživatel ví, že jeho osoba mluví v nahrávce pouze určitý úsek a nemá prostředky nebo čas tento úsek z nahrávky vyříznout, stačí kliknout na sloupec „Trénovat“ v tabulce trénovacích nahrávek a pomocí dialogového okna zadat čas v sekundách, po který mluvčí hovoří. Těchto úseků je možné definovat i více. Tato funkce je v aplikaci dostupná, ale nemá zatím žádný vliv na trénování mluvčího, vždy je trénován na celé nahrávce.
- *Rozpoznávání na vybraném úseku nahrávky* – výběr úseku na zvukové stopě přehrávače zatím slouží pouze k přehrání daného úseku. Do budoucna ale aplikace počítá s odesláním pouze tohoto úseku na rozpoznání. Funkce opět slouží pro nahrávky s více mluvčími.
- *Zobrazení mluvčího pod zvukovou stopou* – je možné, že v budoucnu bude umět software SRTK určit nejen skóre mluvčího na celé nahrávce, ale pokud se bude jednat o řeč více osob, dokáže s jistou pravděpodobností určit, kde který mluvčí hovoří. Takovýto výsledek by se poté zobrazil pod zvukovou stopou jako pole mluvčích a při kliknutí na mluvčího by se vybral úsek nahrávky, ve kterém mluví. Tato funkce je připravena v aplikaci, ale prozatím je její zdrojový kód zakomentován.

6.5. Vzorová data

Aby si nově přichozí uživatel mohl rovnou vyzkoušet funkce webové aplikace, byl vytvořen systém vzorových dat. Data uživatele s *id* = '1' (tedy prvního registrovaného uživatele) jsou označena jako vzorová a zobrazují se všem ostatním uživatelům. Tento uživatel by měl být správcem této aplikace a serveru a měl by ji používat pouze ke správě těchto vzorových dat. Ostatní uživatelé samozřejmě nemají právo tato data upravovat ani mazat.

6.6. Architektura MVC

Jak již bylo několikrát v této zprávě zmíněno, programování ve Flashi je rozděleno do dvou programovacích jazyků podle použití. Forma jazyka XML se využívá k tvorbě statického rozhraní aplikace a jeho částí. Pro programovou logiku je pak k dispozici jazyk ActionScript, který je plně objektový. Toto rozdělení je velice příhodné pro vytváření aplikace podle architektury MVC.

MVC je zkratka tří anglických slov: *Model*, *View* a *Controler*, a tato architektura určuje, jak by měla vypadat struktura budoucí aplikace. V části *model* jsou uloženy vzorové třídy, které zastupují data, nad kterými aplikace pracuje. Jsou to například třídy *Speaker* (mluvčí), která obsahuje data o uživatelových mluvčích, *File* (soubor), která nese název, identifikační číslo a oprávnění k souborům, nebo třída *Record* (nahrávka), která vytváří trénovací nahrávky. Část *view* tvoří aplikační rozhraní, neboli interface. Obsahuje všechny komponenty pro interakci s uživatelem a definuje vzhled aplikace. Je naprogramován pomocí XML značek a jejich atributů. Aplikační logika se nazývá *controler*. Obsahuje výkonný kód aplikace, obstarává obsluhu komponent a manipuluje s daty. Její kód se píše do speciálních XML značek pomocí jazyka ActionScript.

Toto rozdělení aplikace je dnes velice rozšířené a slouží k oddělení nezávislých částí aplikace od sebe. Díky němu je možné například vyměnit kompletní vzhled aplikace bez zásahu do jejího funkčního kódu, nebo změnit typ dat, nad kterými pracuje. Způsob vývoje aplikací na technologii Flash ve dvou programovacích jazycích velmi ulehčuje dodržení této architektury. Navíc umožňuje společnosti Adobe vyvíjet oba jazyky nezávisle na sobě a měnit například chování komponent bez nutnosti úprav v aplikacích.

7. Závěr

Hlavním cílem této bakalářské práce bylo vytvořit webovou aplikaci, která by využívala prostředků systému SRTK a tím zpřístupnila jeho funkce v přijatelné podobě. Na webovou aplikaci bylo vztaženo několik funkčních požadavků, které s sebou nesly jisté problémy. Všechny tyto problémy byly nakonec zdárně vyřešeny a bylo dosaženo všech cílových výsledků, které plynuly ze zadání práce.

7.1. Analýza technologií

Prvním úkolem při vývoji webové aplikace bylo vybrat nejvhodnější technologii pro její zpracování. Proto byla provedena analýza dnes nejpoužívanějších technologií, které přicházeli v úvahu. Těmi byly Flash od společnosti Adobe, SilverLight od firmy Microsoft a nejnovější webový standart HTML5, který vyvíjí organizace W3C. Ostatní webové technologie byly záměrně zamítnuty již na začátku analýzy, neboť nesplňovali některé ze základních požadavků webové aplikace, jakými bylo například on-line přehrávání zvuku na stránce.

Ke každé ze tří kandidátních technologií bylo uvedeno několik kladných bodů, které ji upřednostňují, a několik jejích záporných vlastností. Pak byly tyto body porovnány s požadavky webové aplikace a bylo rozhodováno, jak tyto vlastnosti přispějí nebo uškodí při vývoji.

Po dlouhé úvaze s přihlédnutím k získaným poznatkům během analýzy byla vybrána výsledná technologie, ve které se vytvoří webová aplikace. Tou technologií se stal Flash, který vyvíjí společnost Adobe. Tato technologie nejlépe vyhovovala požadavkům aplikace. Navíc je nejpoužívanější webovou technologií pro tvorbu on-line aplikací. Umožňuje snadné a, díky ucelenému vzhledu komponent, přehledné ovládání aplikace a obsahuje také největší podporu ze stran vývojářů internetových prohlížečů.

Díky výborné dokumentaci a spoustě příkladových aplikací obsažených v [6] byl vývoj aplikace velice efektivní. Technologie Flash má přes své velké výpočetní nároky celkem značný potenciál a správným a efektivním kódováním jednotlivých částí aplikace se tyto nároky dají značně eliminovat.

7.2. Dosažené výsledky

Výsledná webová aplikace je plně funkční a může být použita jako plnohodnotné rozhraní pro systém SRTK. V rámci aplikace byly vytvořeny webové stránky, které zajišťují registraci a přihlašování uživatelů. Na registraci byla implementována jednoduchá ochrana proti počítačovým pirátům. Dále byla vytvořena databáze, která uchovává informace o uživateli a jejich datech.

Po přihlášení uživatele na internetových stránkách se zobrazí webová aplikace. Ta je vytvořena pomocí technologie Flash a proto musí mít uživatel nainstalovanou podporu této technologie v internetovém prohlížeči. Webová aplikace umožňuje vytvářet a trénovat jednotlivé mluvčí a provádět rozpoznávání těchto mluvčích na zvukových nahrávkách. Výsledky trénování a rozpoznávání poskytuje systém SRTK, který nevzniknul v rámci této bakalářské práce. Webová aplikace také umožňuje přehrávat zvukové soubory a zobrazit jejich zvukovou stopu. Ta následně slouží i k pohybu po nahrávce a je umožněno její zvětšování a zmenšování.

Podařilo se nejen umožnit ovládání systému SRTK ve verzi, v jaké se nachází teď, ale aplikace obsahuje i několik funkcí, které počítají s použitím do budoucna. Systém navíc umožňuje ukládání souborů na server, takže jsou uživateli dostupné, ať se přihlásí odkudkoliv.

7.3. Možnosti pokračování

Finální podoba softwaru SRTK není ještě známá, neboť je také stále ve vývoji. Proto obsahuje webová aplikace některé neaktivní funkce, které jsou popsány v kapitole 6.4.3. Je ale pravděpodobné, že ani tyto funkce nepokryjí všechny možnosti systému SRTK. Je tedy možné webovou aplikaci nadále rozvíjet a doplňovat nové možnosti, kterých zatím není schopná.

Webová aplikace také nebyla v průběhu této práce dostupná z žádného internetového zdroje, takže není znám její ohlas. Je možné, že některé části aplikace přijdou uživatelům matoucí a bude nutné je upravit. Navíc se i technologie Flash rychle vyvíjí kupředu a za dobu tvorby webové aplikace bylo oznámeno vydání novější verze, takže je možnost aplikaci upravit podle nových specifikací.

Seznam použité literatury

- [1] SILOVSKÝ, J. *Rozpoznávání mluvčích v záznamech televizních a rozhlasových pořadů*. Liberec, 2006. 56 s. Diplomová práce na Fakultě mechatroniky, informatiky a mezioborových studií Technické univerzity v Liberci na Ústavu informačních technologií a elektroniky. Vedoucí diplomové práce Prof. Ing. Jan Nouza, CSc.
- [2] NOUZA Jan. *Počítačové zpracování řeči – cíle, problémy, metody a aplikace*. 1. vyd. Liberec: Technická univerzita, 2001. 122 s. ISBN 80-7083-551-6.
- [3] KHERMAN, Philip. *ActionScript ve Flashi: podrobná příručka*. 1. vyd. Praha: Computer Press, 2002. 507 s. ISBN 80-7226-615-2 (brož.).
- [4] BOURDON, Romain. *WampServer* [počítačový program]. Ver. 2.1a. [Francie], 2010 [citováno 2011-05-10]. Dostupné z <<http://www.wampserver.com/en/>>. Balíček serverových aplikací. Vyžaduje Windows 2000 a vyšší.
- [5] PALMU, Mika. *FlashDevelop* [počítačový program]. Ver. 3.3.4. 2005 [citováno 2011-05-10]. Dostupné z <<http://www.flashdevelop.org/downloads/releases/FlashDevelop-3.3.4-RTM.exe>>. Vývojové prostředí pro Adobe Flash. Vyžaduje Microsoft Windows .NET 2.0.
- [6] WARD, James. *Tour de Flex* [počítačový program]. Ver. 1.2.1. 2009 [citováno 2011-05-10]. Dostupné z <<http://download.macromedia.com/pub/developer/air/TourDeFlex.air>>. Příklady a tutoriály v jazyce Flash. Vyžaduje Adobe Air.
- [7] KOLYAKO, Denis. *WaveSound.as* [knihovna pro Adobe Flash]. Ver. 1.0 2007 [citováno 2011-05-10]. Dostupné z <<http://etc.ru/pre/WAVPlayer/srcview/>>. Knihovna pro zpracování WAV souborů v jazyce ActionScript 3.0. Určeno pro Adobe Flash.
- [8] BAGWELL, Chris. *SOX* [počítačový program]. Ver. 14.3.2 2011 [citováno 2011-05-10]. Dostupné z <<http://sourceforge.net/projects/sox/files/sox/>>. Program pro převod zvukových formátů.

Příloha č. 1

Webová aplikace pro rozpoznávání mluvčích – Návod

1. Úvod

Tato webová aplikace umožňuje on-line rozpoznávání mluvčích. K tomu využívá software SRTK (speaker recognition toolkit), který obstarává výsledky rozpoznávání. Tento software byl vyvinut na Ústavu informačních technologií a elektroniky na Technické univerzitě v Liberci a je dílem Ing. Jana Silovského.

Aplikace funguje ve dvou různých módech. První je trénovací, kde naučíte systém, jak mluví vaše osoba, neboli mluvčí. Druhá je rozpoznávací, ve které zvolíte nahrávku, na které nevíte kdo mluví a zvolíte, kteří mluvčí na ni mohou pravděpodobně mluvit.

Výsledkem aplikace je procentuální skóre pro každého mluvčího, které vyjadřuje míru pravděpodobnosti, že ve zkoumané nahrávce mluví právě on.

2. Seznam mluvčích

Seznam mluvčích se nachází v pravé části aplikace a obsahuje vaše mluvčí. Jeho zobrazení může být překryto seznamem souborů. Pro zobrazení klikněte na pruh „Mluvčí“.

2.1. Vytvoření mluvčího

Nového mluvčího vytvoříte kliknutím na tlačítko „Nový mluvčí“ v seznamu mluvčích v pravé části aplikace. Poté se objeví dialogové okno, do kterého můžete vyplnit podrobnosti o daném mluvčím. Všechna pole, kromě křestního jména, jsou dobrovolná a slouží pouze pro vaši orientaci. Systém s nimi nepracuje. Pro dokončení akce stiskněte tlačítko „Uložit“ a mluvčí se uloží do aplikace. Zároveň se objeví v seznamu mluvčích v pravé části aplikace.

2.2. Smazání mluvčího

Mluvčího smažete kliknutím na červené tlačítko „Smazat mluvčího“ vpravo od jeho jména v seznamu mluvčích.

3. Seznam souborů

Seznam souborů se nachází v pravé části aplikace a obsahuje vaše soubory nahrané na server. Jeho zobrazení může být překryto seznamem mluvčích. Pro zobrazení klikněte na pruh „Soubory“.

3.1. Nahrání souboru

Pro nahrání souboru na server klikněte na tlačítko „Nahraj nový soubor“. Zobrazí se dialogové okno, ve kterém vyberte soubor umístěný ve vašem počítači. Tlačítkem „Otevřít“ zahájíte nahrávání. Nahrávání souboru může trvat v závislosti na velikosti souboru i několik minut, obvykle je to ale doba v řádu sekund. Během nahrávání je aplikace zšedivělá a neaktivní. Po dokončení nahrávání se soubor objeví v seznamu souborů.

3.2. Smazání souboru

Soubor smažete kliknutím na červené tlačítko „Smazat soubor“ vpravo od jeho názvu v seznamu souborů.

4. Mód „Trénování mluvčích“

4.1. Zobrazení detailu

Nejprve si otevřete detail mluvčího. To provedete tak, že kliknete na jeho název v seznamu mluvčích. Zároveň s tím se vám automaticky seznam přepne na seznam souborů. Pokud mluvčí ještě nebyl trénován, upozorní vás na to červený křížek v pravém horním rohu.

4.2. Přidání trénovacích nahrávek

Ze seznamu souborů vyberte soubory, které jsou trénovací, tak, že je označíte zaškrtnutím před jménem. Pro výběr všech souborů použijte tlačítko „Invertovat výběr“. Poté klikněte na tlačítko „Přidej nahrávky ze souborů“. Soubory se objeví v tabulce trénovacích nahrávek.

4.3. Odebrání trénovacích nahrávek

Pro odebrání trénovací nahrávky klikněte na červenou ikonku „Odebrat“ v tabulce trénovacích nahrávek.

4.4. Trénování

Pro zahájení trénování vyberte zaškrtnutím v prvním sloupci tabulky trénovacích nahrávek ty soubory, na kterých chcete zahájit trénování. Pokud je nahrávka stereofonní, jsou k dispozici dvě zaškrtnutí políčka, pro každý kanál jedno. Vyberte tedy kanál, na kterém chcete trénovat. Systém nedovoluje trénování na obou kanálech najednou. Poté klikněte na tlačítko „Trénovat“ pod tabulkou trénovacích nahrávek.

Proces trénování může trvat v závislosti na počtu a délce jednotlivých nahrávek i několik minut, obvykle je to ale doba v řádu sekund. Během trénování je aplikace zšedivělá a neaktivní. Po skončení procesu se objeví upozornění, že trénování proběhlo úspěšně.

5. Mód „Rozpoznávání“

5.1. Načtení souboru

Pro zobrazení souboru klikněte na ikonu zelené šipky „Načíst tento soubor“, která se nachází vpravo od názvu souboru v seznamu souborů.

5.2. Přehrávač

Po načtení souboru se vám automaticky aktivuje jeho přehrávač. Tlačítka „Přehraj/Zastav“ a „Přetoč“ ovládáte samotné přehrávání. Tlačítka „Přiblížit“ a „Oddálit“ přibližujete nebo oddalujete zobrazení zvukové stopy. Pro posouvání po nahrávce můžete kliknout myší do zvukové stopy. Pro přehrání výběru z nahrávky vyberte tento úsek přetažením myši přes zvukovou stopu.

Pod přehrávačem se nacházejí stručné informace o souboru a aktuální pozice přehrávače v sekundách a procentech.

5.3. Rozpoznávání

Pro rozpoznávání musíte mít načtený soubor, na kterém chcete rozpoznávat. Ze seznamu mluvčích vyberte ty, které chcete trénovat, tak, že je označíte zaškrtnutím před jménem. Pro výběr všech mluvčích použijte tlačítko „Invertovat výběr“. Vybrat jdou pouze ti mluvčí, kteří byli již trénováni. Poté klikněte na tlačítko „Rozpoznat“.

Proces rozpoznávání může trvat v závislosti na počtu mluvčích a délce nahrávky i několik minut, obvykle je to ale doba v řádu sekund na jednoho mluvčího. Během rozpoznávání je aplikace zšedivělá a neaktivní a aplikace vás průběžně informuje, který mluvčí je právě rozpoznáván. Po skončení procesu se objeví výsledky rozpoznávání. Každému mluvčímu je přiřazeno skóre v podobě procent. Tato procenta určují míru pravděpodobnosti, že ve zkoumané nahrávce mluví právě on. Jsou seřazeny sestupně podle jejich skóre a pokud jich je více jak 3, jsou zobrazeny pouze 3 nejpravděpodobnější.

Pozor, výsledky rozpoznávání jsou pouze informativní a neslouží jako právní důkaz.