TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: N2612 – Elektrotechnika a informatika
Studijní obor: Informační technologie

Software pro efektivní zpracování řečových databází

Software for effective processing of spoken documents

Diplomová práce

Autor: **Martin Čičkán**
Vedoucí práce: Ing. Jindřich Žďánský, Ph.D.

V Liberci 18. 5. 2009

Prohlášení

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

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé diplomové práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom(a) toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

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

Datum

Podpis

Anotace

Cílem diplomové práce je seznámit se s problematikou zpracování mluvených dokumentů a vytvořit počítačovou aplikaci pro jejich efektivní ruční zpracování. Program bude obsahovat potřebné nástroje pro efektivní zpracování mluvených dokumentů: podporu přehrávání zvukových a video souborů, zobrazení grafické podoby zvukových dat, nástroje pro správu seznamů mluvčích, kteří se v přepisech vyskytují, členění textového přepisu do přehledných úrovní a jeho ukládání ve vhodném formátu.

Pro usnadnění ruční práce bude vytvářený software obsahovat podporu rozpoznávání spojité řeči (technologie v2t) pro automatické přepisování částí nebo celých mluvených dokumentů. Při návrhu grafického uživatelského prostředí aplikace bude kladen důraz na její intuitivní ovládání. Vybrané funkce budou moci být ovládány hlasem pomocí implementované technologie v2t.

Annotation

The aim of the diploma thesis is to introduce the problems of processing the spoken documents and produce a computer application for its handmade processing. The program will consist of necessary tools for effective processing of spoken documents: support for playback of audio and video files, projection of graphical form of audio data, tools for managing the lists of speakers, who occur in the list, segmentation of rewritten texts into well-arranged levels and their saving in a proper format.

To simplify the handwork, the software will content the support of recognition of a continuous speech (technology v2t) for automatic rewriting of parts or whole spoken documents. By designing the graphical user's environment of the application the intuitive control will be emphasized. It will be possible to control the chosen functions by voice using the implemented technology v2t.

Obsah

1 Úvod

Cílem diplomové práce bylo seznámit se s problematikou zpracování mluvených dokumentů a na tomto základě vytvořit počítačovou aplikaci, která umožní přepis řečových dokumentů (audiovizuálních souborů) do textové podoby.

Program by měl umožňovat načítat a přehrávat soubory nejen základních zvukových formátů, ale měl by obsahovat podporu i pro ostatní multimediální formáty audiovizuálních souborů. Další základní funkcí vytvářené aplikace bude grafické zobrazení načtené audio části souboru do podoby vlny. Tato funkce umožní jednodušší orientaci v audio souboru a časovou synchronizaci vytvářeného textového přepisu s původním řečovým dokumentem.

Vytvářená aplikace by měla také umožnit přehrát společně s audio souborem i video. Velká část řečových materiálů, které je potřeba přepisovat, totiž obsahuje kromě zvukové části i video, které zjednodušuje identifikaci jednotlivých mluvčích.

Program bude také podporovat správu seznamů mluvčích, kteří se v mluvených dokumentech vyskytují.

Pro usnadnění ručního přepisu řečových dokumentů bude v aplikaci využita technologie v2t dodaná laboratoří Speechlab Technické univerzity v Liberci, která umožní automatický přepis řečových dokumentů v českém jazyce do textové podoby.

Vytváření přepisů je značně časově náročné, proto bude program navržen tak, aby byl přehledný a většina funkcí rychle dostupných.

Práce je rozdělena na několik tématických částí, které se věnují způsobu vytváření jednotlivých částí aplikace. Úvodní část práce se zabývá problematikou zpracování mluvených dokumentů a popisuje důvody jejich vytváření a způsob použití. Ve třetí kapitole bude popsáno vývojové prostředí pro tvorbu software a platforma, pro kterou bude aplikace vytvořena. Kapitoly 4 až 8 budou věnovány tvorbě samotné aplikace, návrhu uživatelského rozhraní, návrhu struktury textového přepisu a způsobu práce s audio a video soubory. V kapitole 9 bude popsána technologie v2t pro rozpoznávání spojité řeči a její způsoby implementace v aplikaci. Kapitola 10 se bude zabývat datovou strukturou navrhovaného software a jeho hardwarovými a softwarovými požadavky. Závěrečná část práce se bude věnovat dosaženým výsledkům při tvorbě aplikace a dalším možnostem vývoje.

2 Problematika zpracování mluvených dokumentů

Mluveným (řečovým) dokumentem se rozumí zvukový respektive audiovizuální zdroj řeči (audio a video soubory, rádiové a televizní vysílání, apod.) Pro zpracování informací z řečových dokumentů, se využívá jejich textových přepisů. Textovým přepisem mohou být různé formy dokumentů, ve kterých je textem popsáno, co bylo v řečovém dokumentu vysloveno. Kdo danou část řekl (mluvčí) a v jakém místě řečového dokumentu se promluva nachází (časový index).

2.1 Důvody vytváření textových přepisů

Řečové dokumenty se nejčastěji zpracovávají převážně ze tří hlavních důvodů. Jednotlivé důvody budou popsány v následujících podkapitolách.

2.1.1 Vyhledávání a popis multimediálních souborů

Řečové dokumenty jsou ukládány v audio a video souborech. V těchto souborech se špatně vyhledává i orientuje, pokud není známo ve které jejich části se nachází požadovaná informace. Aby bylo usnadněno vyhledávání informací v řečových dokumentech, vytvářejí se jejich textové přepisy. Tyto textové přepisy pak lze dále rozdělit na dvě skupiny:

a) ***Kompletní přepis multimediálního souboru*** – nejčastěji se používá v tištěných médiích a na internetu (například: rozhovory a komentáře). Umožňuje rychlé textové vyhledávání a může obsahovat i časové indexy do původního řečového dokumentu. Ve většině případů se ale nejedná o přesné přepisy mluvených dokumentů, protože v nich nejsou zahrnuty různé chyby, kterých se dopouští jednotliví mluvčí (například: nespisovná řeč, koktání, přeřeknutí atd.).

b) ***Částečný přepis multimediálního souboru*** – oproti předchozímu bodu (a) se nejedná o přepis celého řečového dokumentu, ale pouze jeho částí. Používá se pro charakteristiku daného souboru a umožňuje získat informace, o čem daný řečový dokument pojednává (například: jednotlivá témata ve zpravodajství). Usnadňuje hlavně vyhledávání tématických částí v multimediálních souborech.

2.1.2 Řečové systémy

Řečovým systémem rozumíme aplikaci, která je schopna automaticky rozpoznávat řeč (například: technologie v2t – viz. kapitola 9). Pro trénování a zjišťování úspěšnosti řečového systému je zapotřebí vytvořit přesný přepis audio souborů včetně časových

indexů (v původním zvukovém dokumentu) a chyb řeči (koktání, nespisovná řeč, přeřeknutí, atd.). Také je vhodné mít k dispozici informace o mluvčích, kteří se v přepisech vyskytují (zda jde o muže, ženu, jejich podrobnější hlasovou charakteristiku apod.).

2.1.3 Titulkování multimediálních souborů

Textové přepisy se také velice často používají pro tvorbu titulků k audiovizuálním materiálům. Titulky mohou být využívány pro neslyšící, případně pro další zpracování – překlady do jiných jazyků, tvorba dabingu. Textový přepis pro titulkování musí obsahovat časové indexy (do audiovizuálního dokumentu) a pokud je tvořen překlad do jiných jazyků, tak i informaci o mluvčích, kteří se v dokumentu vyskytují.

2.2 Požadované vlastnosti software pro vytváření přepisu

Pro efektivní ruční vytváření textových přepisů řečových dokumentů (tzv. transkripci) je zapotřebí vytvořit software, který bude mít dále popsané vlastnosti. Vlastnosti software budou navrženy tak, aby bylo umožněno vytvářet textové přepisy řečových dokumentů požadovanými způsoby (viz. kapitola 2.1).

a) Vytvářený textový přepis musí být možno rozčlenit do přehledné struktury. Členění umožní zpřehlednit vytvářený přepis, což je důležité při přepisování řečových dokumentů, kde se vyskytuje mnoho témat a více mluvčích. Typickým příkladem je zpravodajská relace.
b) Strukturu textového přepisu zmiňovanou v bodě (a) musí být možno ukládat do vhodného formátu, aby s ní bylo možno dále pracovat. Formát by měl být zvolen s ohledem na budoucí požadavky zpracovávat vytvořený textový přepis i v jiných programech. (například při trénování řečových systémů, vytváření titulků apod.)
c) Pro vytvoření přepisu je důležité, aby program umožňoval načíst a přehrávat audio soubory. Pro zvýšení efektivity, by měl program umět pracovat s různými formáty audio i video souborů. V případě podpory pouze některých formátů audio souborů (např. WAV), by musely být ostatní formáty převáděny na podporovaný formát pomocí jiného software.
d) Pro usnadnění práce s načteným audio souborem je důležité, aby byl program schopen zobrazit jeho grafickou reprezentaci v podobě vlny. Grafická

reprezentace umožňuje jednodušší orientaci a vyhledávání příslušných míst v audiovizuálních materiálech.

e) Vytvářený software by měl dále umožnit překrývat časy promluv jednotlivých mluvčích, kteří se v přepisu řečového dokumentu vyskytují. Jedná se o případy, kdy se v řečovém dokumentu vyskytuje více mluvčích, kteří mluví naráz. Tento jev se vyskytuje například v televizních debatách, kdy ještě před skončením promluvy jednoho mluvčího, začíná mluvit další.
f) Společně s audio daty by měl vytvářený program umožňovat zobrazení videa, pokud je k dispozici. Zobrazené video by mělo pomoci při identifikaci mluvčích, pokud se jich v přepisovaném řečovém dokumentu vyskytuje více.
g) Aplikace by měla umožňovat efektivně a pohodlně pracovat s podrobnějšími informacemi o mluvčích, kteří se vyskytují ve vytvářeném přepisu řečového dokumentu.
h) Mimo přepisu samotných řečových dat, je také potřeba zaznamenávat do přepisu i různé neřečové zvuky. Program by měl proto obsahovat nástroje pro rychlejší přepis takových zvuků (například v podobě jejich seznamu) a umožňovat jejich rychlejší vkládání do textového přepisu.
i) Pro usnadnění ruční transkripce je vhodné, aby vytvářený software uměl přepisovat řečové dokumenty nebo alespoň jejich části automaticky. Toho lze docílit implementací řečové technologie *v2t*, popsané v kapitole 9.
j) Jelikož je ruční vytváření transkripcí náročné časově (vytvoření přepisu řečového dokumentu trvá déle než je jeho původní délka), je zapotřebí, aby bylo ovládání aplikace efektivní a intuitivní. Vytvářená aplikace by tak měla být navržena s ohledem na tuto skutečnost.

Výše zmíněné požadavky budou při vývoji aplikace zohledněny a jejich implementace bude popsána v následující části diplomové práce, která se bude zabývat tvorbou samotné aplikace.

3 Vývojové prostředí, platforma, programovací jazyk

3.1 Vývojové prostředí

Vytvářená aplikace bude programována pro platformu .NET Framework. Jedná se o verzi .NET 3.0, která obsahuje podsystém WPF (Windows Presentation Foundation)

Jako vývojové prostředí pro tvorbu aplikace bylo zvoleno *Microsoft Visual Studio 2008 Professional Edition*, které přímo obsahuje nástroje pro tvorbu aplikací v podsystému .NET Framework WPF (popsáno níže), bez nutnosti tuto podporu doinstalovávat, jako v předchozích verzích.

3.2 WPF (Windows Presentation Foundation)

Jedná se o podsystém platformy .NET Framework. WPF je následníkem *Windows Forms* - dosavadního způsobu vývoje převážně formulářových aplikací pod .NET. Tento podsystém se objevil s verzí .NET 3.0.

WPF je zaměřeno na tvorbu graficky bohatých aplikací. Celý systém zobrazení aplikace (formulářů, tlačítek, atd.) běží pod rozhraním DirectX. Zobrazení aplikace má tak na starost grafická karta. To s sebou přináší snížení zátěže procesoru, který může zpracovávat jiné úlohy.

Další vlastností WPF je oddělení návrhu uživatelského rozhranní od aplikační logiky. Vzhled aplikace je popsán pomocí speciálního značkovacího jazyka XAML (eXtensible Application Markup Language), který vychází z XML (viz. kapitola 5.3).

3.3 Programovací jazyk

V platformě .NET lze pro psaní aplikace využít třech programovacích jazyků. Jsou jimi: *Visual C++, Visual Basic a Visual C#.*

Pro tvorbu aplikační logiky byl zvolen programovací jazyk C#. Tento jazyk má syntaxi vycházející z jazyka C/C++. Oproti těmto jazykům, má však například vylepšenou správu paměti (obsahuje *garbage collector*, starající se o uvolňování paměti již nepoužívaných proměnných). Jedná se o objektově zaměřený jazyk, který byl vyvinut společně s platformou .NET firmou Microsoft.

Jelikož je aplikace vytvářena ve WPF, tak dalším programovacím jazykem pro vytvoření uživatelského rozhranní je XAML (viz. kapitola 3.2)

4 Uživatelské rozhraní aplikace

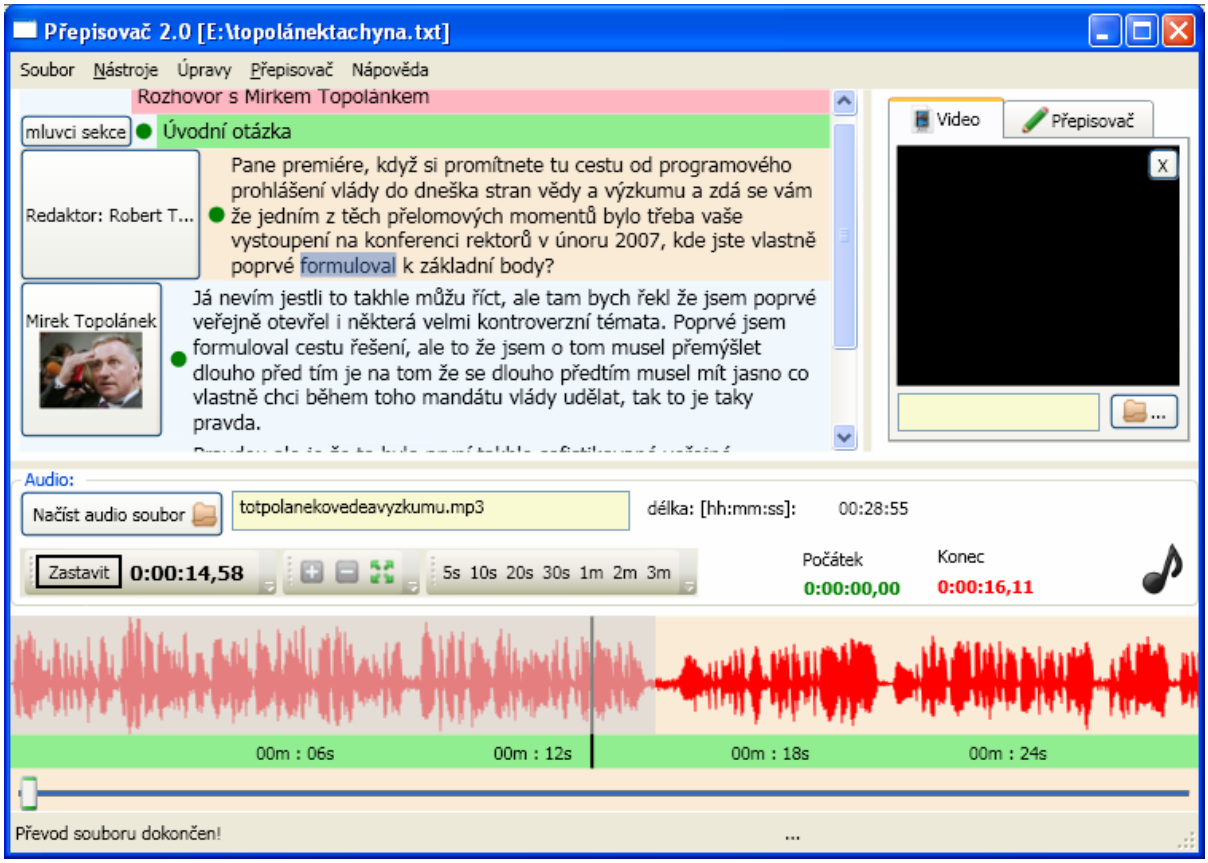
4.1 Návrh uživatelského rozhraní

Pro vytvářenou aplikaci, která bude schopna efektivního zpracování řečových dokumentů, bylo zapotřebí vyvinout uživatelské rozhraní. Návrh uživatelského rozhraní vychází z problematiky zpracování řečových dokumentů (popsané v kapitole 2) a především z uvažovaných požadavků na možnosti programu (viz. kapitola 2.2).

Uživatelské rozhraní bylo navrženo pomocí jazyka XAML (eXtensible Application Markup Language) pro podsystém (platformy .NET) WPF – viz. kapitola 2.2. V následujícím popisu aplikace bude upřesněn způsob jejího návrhu a budou popsány jednotlivé komponenty, které byly použity. Také bude popsán princip a možnosti ovládání programu.

4.2 Popis navrhovaného uživatelského rozhraní

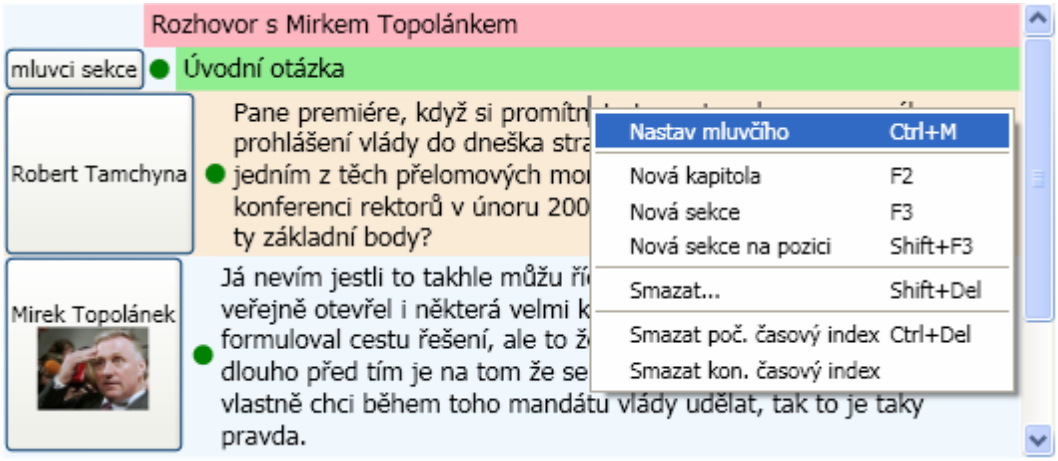
4.2.1 Hlavní formulář aplikace



Obrázek 4.1: Hlavní ovládací formulář aplikace

Nejdůležitější část uživatelského rozhraní je umístěna na hlavním formuláři programu. Tento formulář byl navržen s ohledem na zpřístupnění všech potřebných funkcí tak, aby si zachoval přehlednost a měl potřebnou funkčnost. Podoba hlavního formuláře aplikace je vidět na obrázku 4.1. Celý formulář lze rozdělit do šesti samostatných oblastí, které budou dále popsány.

(A – textový přepis)



Obrázek 4.2: Detail oblasti pro textový přepis řečového dokumentu

Nejdůležitější oblastí celého formuláře je část (A), ve které je vytvářen a zobrazován samotný přepis. Na formuláři zaujímá největší plochu, která se dá ještě na úkor ostatních částí (B, C – pokud nejsou aktuálně potřeba) rozšířit pomocí posuvníků (S*plitter*). Samotný přepis je reprezentován vizuálními komponentami *TextBox*. Každý *TextBox* reprezentuje jednu úroveň přepisu (Návrhem a popisem úrovní se zabývá kapitola 5). Každý *TextBox* je ještě umístěn na samostatném kontejneru, v tomto případě *Grid* (koponenta, která umožňuje zarovnávat další vizuální komponenty, např. tlačítka apod.), který může dále obsahovat tlačítko pro změnu a zobrazení aktuálního mluvčího a dále identifikační zelený bod, který určuje, zda má element přiřazen svůj časový index začátku v příslušném řečovém dokumentu. Každý *Grid* je pak vertikálně umístěn na *StackPanel*u (komponenta, která umožňuje poskládat libovolné vizuální komponenty za sebou a jejich velikost pak dynamicky přizpůsobovat podle velikosti formuláře programu, případně rozlišení obrazovky). Při zobrazení většiny textových přepisů, nestačí pro zobrazení plocha vyhrazen na formuláři. Proto je celý *StackPanel* umístěn v komponentě *ScrollViewer*, která v případě že velikost celého panelu přesáhne vymezený prostor, umožní posouvat jeho obsah pomocí vertikálního posuvníku (tzv. scrolování).

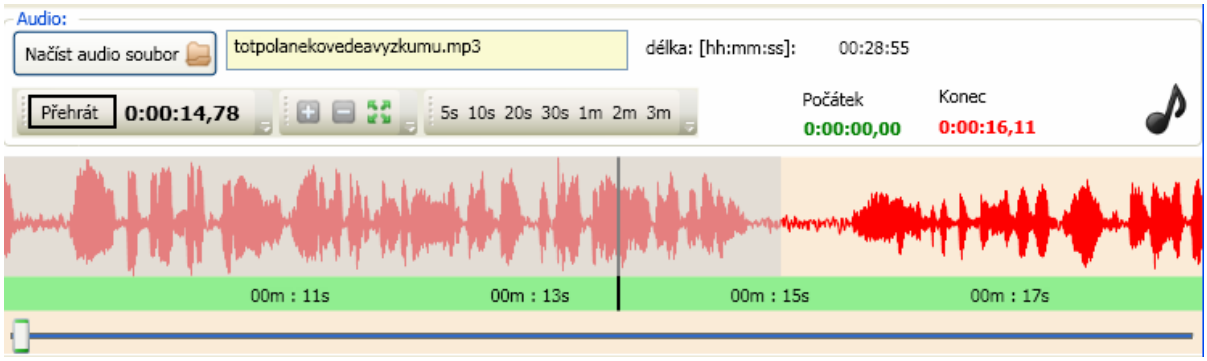
Aby byly úpravy a vyhledávání v delších dokumentech přehlednější, je zpřístupněna pouze možnost vertikálního posunu textového dokumentu.

Pro efektivní vytváření a editaci textového přepisu řečového dokumentu obsahuje popisovaná část aplikace následující nástroje:

Jednotlivé úrovně textového přepisu jsou barevně odlišeny. Kapitoly růžovou barvou, sekce zelenou a odstavce bílou barvou. V levé části se nachází tlačítko, které udává mluvčího elementu. Při stisknutí tohoto tlačítka je zpřístupněn formulář pro práci s mluvčími (viz. kapitola 6). Při stisknutí tlačítka vedle celé sekce přepisu, lze nastavit mluvčího pro celou tuto sekci. V případě vybrání mluvčího v úrovni odstavce, je nastaven mluvčí pouze pro daný odstavec.

Pokud je vedle libovolné úrovně textového přepisu zobrazen zelený kruh, má tato přiřazen časový index svého počátku v řečovém dokumentu (multimediálním souboru). Nad každým elementem lze vyvolat pomocí pravého tlačítka myši kontextové menu, které umožní vybrat mluvčího, vytvořit novou kapitolu, novou sekci, novou sekci, která je vložena mezi stávající odstavce sekce. Dále je v menu přístupná položka pro smazání libovolné úrovně textového přepisu (kapitoly, sekce, odstavce). Dalšími položkami jsou funkce pro smazání časových indexů elementů (počátku a konce).

(B – zvukový soubor)



Obrázek 4.3: Detail oblasti pro práci s audio souborem

Druhou částí hlavního ovládacího formuláře programu, je část umožňující práci s audio soubory (B). Zde je možno načíst multimediální soubor (pomocí příslušného tlačítka), jehož zvuková část je po převedení do podporovaného formátu (viz. kapitola 7) graficky zobrazena (viz. kapitola 8). Dále je zde zobrazováno jméno načteného souboru, doba trvání souboru, aktuální pozice přehrávání a také časové indexy počátku a konce textových elementů (popsané v (A)). Pokud nemá element jeden nebo oba časové indexy, je místo času zobrazeno „N/A“. Pro práci se zobrazenou

částí audio souboru je určen panel nástrojů, který obsahuje všechna potřebná tlačítka: *přehrát/zastavit*, které slouží k ovládání přehrávání, tlačítka *zoomování,* pro úpravu velikosti audio vlny ve svislém směru (+, -, automaticky) a tlačítka, která umožňují zobrazení požadovaného úseku vlny (od 5s až po 3 minuty).

Pokud je načten audio soubor, je graficky vykreslen v podobě vlny na *Image* (komponenta umožňující zobrazit obrázky a ostatní grafiku). Společně s grafickým zobrazením audio souboru, je zobrazena také časová osa aktuálně načtené části multimediálního souboru. Ve spodní části oblasti se nachází posuvník - *Slider* (komponenta, která graficky zobrazuje pozici v právě přehrávaném souboru), pomocí něhož lze se lze rychle posouvat v načteném multimediálním souboru. Pokud dochází k převodu multimediálního souboru, zobrazuje *Slider* také aktuálně přístupnou část audio souboru, kterou je možno přehrávat a zobrazovat.

Aktuální pozice v audio souboru je ve vlně zvýrazněna pomocí svislého kurzoru. Pokud má aktuální textový element z části (A) přiřazeny časové indexy počátku a konce, je zobrazen ve vlně výběr pomocí šedého poloprůhledného obdélníku. Při kliknutí myší do oblasti vlny, je nastavena pozice kurzoru, audio a video souboru a přehrávání zastaveno. Při stisknuté klávese *Ctrl* lze pomocí myši vybrat libovolný úsek vlny. Pomocí kontextového menu lze pak časy tohoto úseku přiřadit elementu z části (A). Pokud se časy po sobě následujících elementů překrývají, je o tom uživatel informován pomocí informačního okna. Jestliže se překrývají časy následujících odstavců pro různé mluvčí, lze povolit jejich překrývání. V kontextovém menu se dále nacházejí volby pro přidání synchronizační značky pro element odstavce (viz. kapitola 5.1). Důležitou položkou tohoto menu je možnost automatického rozpoznání vybrané části audio souboru. Jedná se o využití technologie *v2t* (viz. kapitola 9 a část (D)).

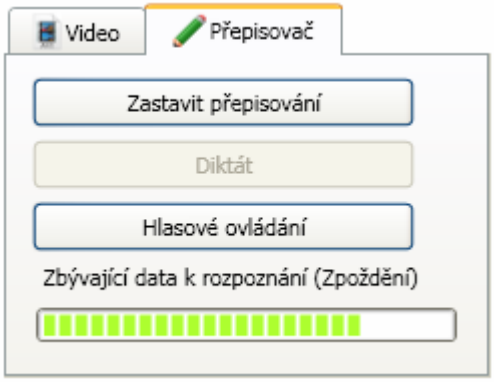
(C – video)

Obrázek 4.4: Detail oblasti pro přehrávání videa

Tato část hlavního formuláře programu slouží k otevření souboru s videem a jeho zobrazení. Způsob přehrávání videa je popsána v kapitole 7.4.3. Je zde zobrazeno jméno načteného souboru a vedle se nachází příslušné tlačítko pro otevření video souboru. Vlevo se nachází posuvník (*Splitter*), kterým je možno video zvětšovat na úkor části (A). Video lze zavřít kliknutím na tlačítko „X“.

Oblast pro přehrávání videa má vlastní kontextovou nabídku přístupnou pomocí pravého tlačítka myši. V nabídce se nachází možnost pořídit snímek videa (screenshot) a tento obrázek předat správci mluvčích (viz. kapitola 4.2.2)

(D – automatické přepisování)



Obrázek 4.5: Detail oblasti pro automatické přepisování

Tato část ovládacího formuláře aplikace je společná s částí videa a nachází se na další záložce komponenty *TabControl* (komponenta, která umožňuje zobrazení více stránek pomocí záložek). Popisovaná část slouží k ovládání a přehledu implementované technologie *v2t* (viz. kapitola 9). Nachází se zde tři tlačítka, která slouží ke spuštění různých módů automatického rozpoznávače.

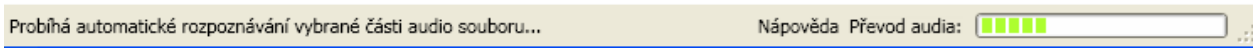
První tlačítko *Automatický přepis* spustí offline rozpoznávání audio souboru, pokud je načten multimediální soubor a je vybrán příslušný element textového přepisu, který má nastaveny synchronizační značky počátku a konce. Automatické rozpoznávání běží nezávisle ve vlastním vlákně (thread) a s programem je možno dále běžně pracovat – kromě úpravy přepisované části textového dokumentu. Další informace o offline přepisování jsou uvedeny v kapitole 9 o technologii *v2t.* Stav automatického přepisu (zbývající data k rozpoznávání) je znázorněn pomocí komponenty *ProgressBar* (komponenta, která umožňuje vizuálně zobrazit stav nějaké činnosti), umístěné pod tlačítky na stejné záložce. Probíhající automatický přepis je možno kdykoliv zastavit stisknutím stejného tlačítka.

Druhé tlačítko *Diktát* umožňuje za pomoci mikrofonu automatický přepis řeči uživatele do textové podoby. Diktovaný text je ukládán do elementu textového přepisu, který byl vybrán před začátkem diktování. Diktování je umožněno až po inicializaci rozpoznávače. O stavu inicializace a možnosti diktování je uživatel informován na stavovém řádku (viz. část (E)). Pokud počítač na kterém běží diktování nestíhá přepisovat v reálném čase, je o aktuálním zpoždění uživatel vizuálně informován pomocí komponenty *ProgressBar,* umístěné ve spodní části záložky.

Jelikož je automatické rozpoznávání a diktování velice náročné na výkon počítače, je zpřístupněna vždy pouze jedna volba (tzn. Nelze současně diktovat a offline rozpoznávat audio soubor)

Třetím tlačítkem *Hlasové ovládání* na popisované části formuláře, lze spustit hlasové ovládání některých funkcí programu. Hlasově lze ovládat některé základní funkce hlavního formuláře programu, které usnadňují ruční přepis (například nastavení mluvčího apod. – seznam všech podporovaných příkazů je popsán v příloze B). Hlasové ovládání je oproti diktování méně náročné na rychlost počítače (viz. kapitola 9.2) a proto ho lze používat samostatně i společně s automatickým offline rozpoznáváním řečového dokumentu.

(E – stavový řádek)



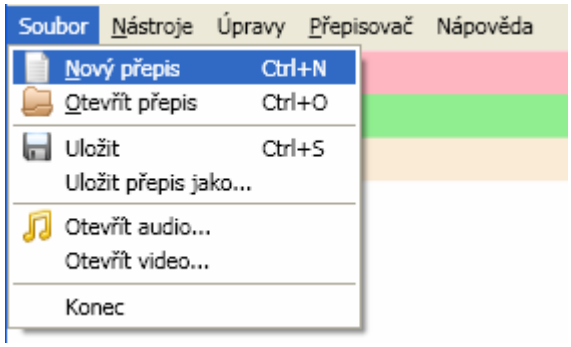
Obrázek 4.6: Detail stavového řádku a příklad jeho použití.

Ve spodní části formuláře se nachází stavový řádek (*StatusBar*). Tato komponenta umožňuje zobrazovat informace o stavu aplikace. Pro efektivní práci s aplikací je důležité, aby byl uživatel informován o jejím aktuálním stavu a chování. Proto je stavový řádek schopen zobrazit textově a vizuálně následující:

Zobrazení informací o průběhu převodu multimediálního souboru včetně grafické interpretace zbývajícího množství dat k převedení pomocí komponenty *ProgressBar*. Informace o dokončeném převodu a možnosti zobrazovat a přehrávat celý audio soubor.

Zobrazení informací při použití automatického offline rozpoznávače, diktování a hlasovém ovládání aplikace (technologie *v2t*). Uživatel je informován o inicializaci rozpoznávače, průběhu rozpoznávání a v případě diktování a hlasového ovládání o nahrávání z mikrofonu a rozpoznaném hlasovém povelu. V případě přerušení nebo dokončení přepisu je také uživatel informován o ukončení rozpoznávání.

(F – hlavní menu aplikace)



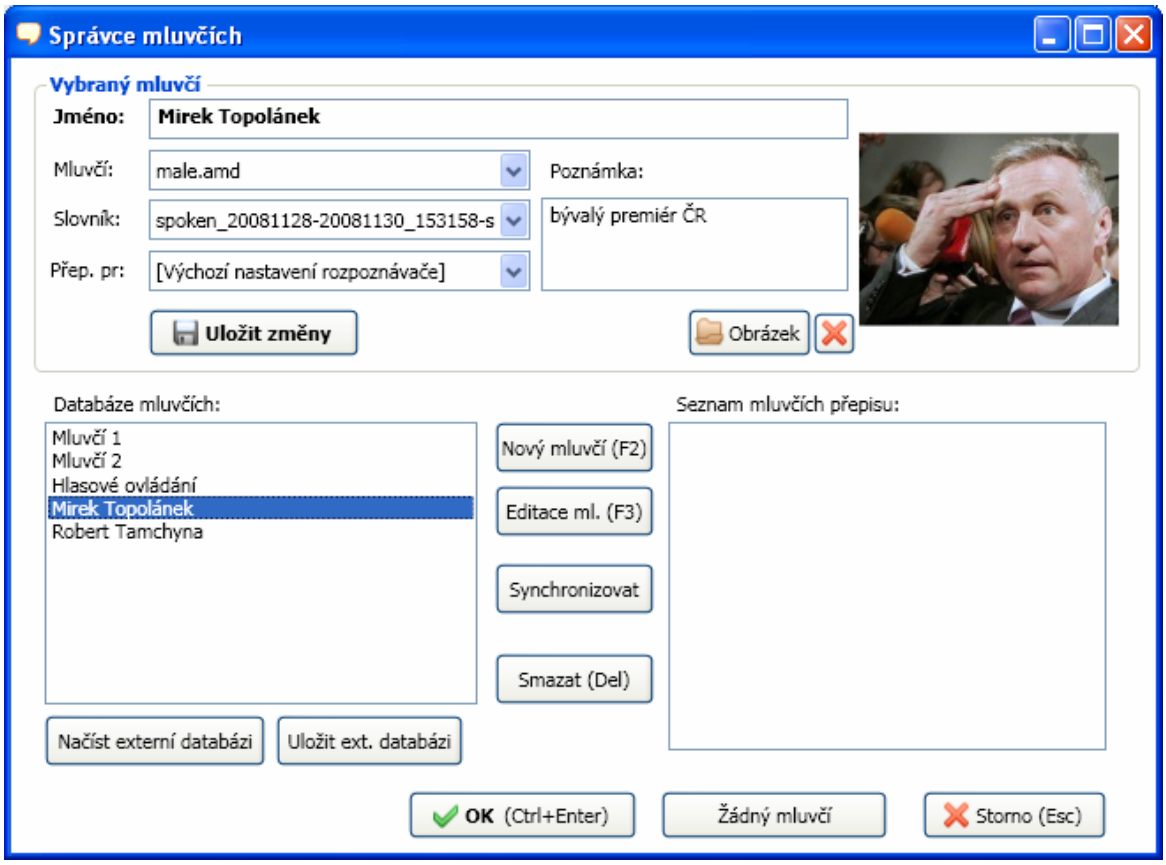
Obrázek 4.7: Hlavní menu aplikace

V horní části aplikace se nachází hlavní menu celé aplikace. V jednotlivých nabídkách se nacházejí všechny důležité volby pro ovládání a nastavení programu. Seznam a popis všech funkcí v nabídkách hlavního menu programu je umístěn v příloze C. Pokud existuje klávesová zkratka pro daný příkaz v nabídce, je u příslušné položky zobrazena.

Pro efektivní práci s aplikací, je nutné, aby často používané funkce byly přístupné přímo z klávesnice. Časté vyhledávání v nabídkách za pomoci myši je totiž příčinou zpomalení při ručním vytváření textového přepisu řečového dokumentu. Proto je většina funkcí programu (mimo nabídek, panelů nástrojů a tlačítek) taktéž přístupná pomocí klávesových zkratek. Seznam klávesových zkratek a jejich význam je popsán v příloze A.

4.2.2 Formulář správce mluvčích

Druhým důležitým formulářem uživatelského rozhraní aplikace je okno správce mluvčích. Návrh tohoto formuláře vychází z potřeby ukládat různé informace o jednotlivých mluvčích přepisu (viz. kapitola 6).



Obrázek 4.8: Formulář správce mluvčích.

Okno správce mluvčích je zobrazeno při stisknuté klávesové zkratce (Ctrl+M), při zvolení položky kontextového menu nebo položky hlavního menu (Nástroje – Nastav mluvčího). Pomocí tohoto formuláře lze vytvořit nového mluvčího, který se v přepisu bude vyskytovat a přiřadit libovolného mluvčího elementu hlavního formuláře, ze kterého byl správce mluvčích spuštěn. Lze také libovolného mluvčího smazat, potom je odstraněn ze všech datových struktur programu.

V horní části formuláře jsou zobrazeny všechny informace o vybraném mluvčím, které se dají upravovat: jméno, typ mluvčího, slovník, přepisovací pravidla, poznámka, obrázek. Pokud je vytvářen nový mluvčí (klávesa F2), jsou zde tyto informace přímo zadávány.

Ve střední části jsou zobrazeny dva seznamy s mluvčími. V levém sloupci jsou zobrazena jména všech mluvčích, kteří se vyskytují v právě načtené interní databázi programu. Příslušným tlačítkem lze databázi uložit a případně načíst. V pravém sloupci je zobrazen seznam mluvčích, kteří jsou nebo byli použiti ve vytvářeném textovém přepisu. Vybrané mluvčí lze také mazat jejich vybráním v příslušném sloupci a stisknutím tlačítka (nebo klávesou Delete). Tlačítko *Synchronizovat* přidá do interní

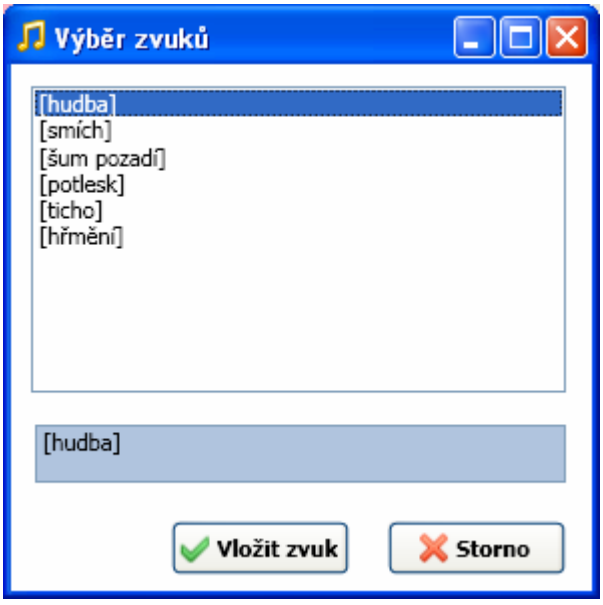
databáze všechny mluvčí, kteří se vyskytují v právě načteném přepisu, ale nejsou přítomni v interní databázi.

Ve spodní části formuláře se nachází trojice tlačítek, která slouží pro potvrzení a nastavení mluvčího do textového přepisu. Pokud je vybrán mluvčí, tlačítko *OK* ho přiřadí elementu textového přepisu na hlavním formuláři. Tlačítko *Žádný mluvčí* vymaže mluvčího z příslušného elementu textového přepisu. Tlačítko *Storno* pak ponechá původního mluvčího, který byl elementu doposavad přiřazen

Vytvářená aplikace je vyvíjena s ohledem na možnost ovládat ji převážně pomocí klávesnice. Formulář spravující mluvčí je proto možno intuitivně ovládat právě pomocí klávesnice tak, aby nejpoužívanější funkce jako je volba mluvčího, případně vytvoření nového bylo možno provést bez nutnosti použít myš. Složitější funkce jako je načtení a uložení seznamu mluvčích z externích souborů je pak ponecháno na použití myši, protože tyto funkce nejsou tak často používané.

4.2.3 Formulář neřečových zvuků

V řečových dokumentech, které je nutné přepisovat, se mimo běžné řeči objevují různé neřečové zvuky, které je potřeba přepisu zaznamenat. Pro zrychlení přepisu, obsahuje program seznam několika základních zvuků, které lze vkládat do textu a není je tak nutné ručně zapisovat. Okno ve kterém je možno tyto zvuky vybrat, je zobrazeno po stisku klávesové zkratky (Ctrl+R) nebo po zvolení položky hlavního menu (Nástroje – Vložit Ruch).



Obrázek 4.9: Formulář neřečových zvuků

Formulář je velice jednoduchý a obsahuje seznam nejpoužívanějších neřečových zvuků, jako je například (hudba, smích, ticho, apod.). Seznam těchto zvuků je uložen v souboru formátu XML (viz. kapitola 5.3). Jelikož má soubor jednoduchou strukturu, lze případné další neřečové zvuky přidat jeho ruční editací (umístění souboru viz. kapitola 10.2).

```
<?xml version="1.0" encoding="utf-8"?>
<MySounds xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ruchy>
    <string>[hudba]</string>
    <string>[smích]</string>
    <string>[šum pozadí]</string>
    <string>[potlesk]</string>
    <string>[ticho]</string>
  </ruchy>
</MySounds>
```

Obrázek 4.10: Příklad souboru neřečových zvuků

Vybraný zvuk je vložen na pozici kurzoru aktuálního elementu hlavního formuláře, ze kterého byl správce zvuků vyvolán.

4.2.4 Formulář nastavení programu

Ve vytvářené aplikaci lze měnit mnoho parametrů. Všechny tyto parametry lze nastavovat na speciálním formuláři *Nastavení*. Formulář obsahuje komponentu *TabControl*, pomocí které je možno se přepínat mezi čtyřmi záložkami s nastavením.

První záložkou je *Audio*, kde lze nastavit výstupní zařízení pro přehrávání zvuku aplikací a také vstupní zařízení pro záznam zvuku aplikací pomocí mikrofonu.

Druhá záložka *Rozpoznávač* je vyhrazena technologii *v2t* (viz. kapitola 9). Zde lze nastavit výchozí parametry (mluvčí – akustický model, jazykový model, přepisovací pravidla, licenční server, soubor s licencí, velikost vyrovnávací paměti)

Na třetí záložce *Hlasové ovládání / diktát* lze nastavit parametry mluvčího (uživatele aplikace) pro technologii *v2t* pro módy hlasového ovládání a diktování.

Čtvrtá záložka *Ostatní* umožňuje nastavit cestu k souboru s interní databází mluvčích přepisu (viz. kapitola 6.2). Další položkou je možnost změnit vzhled textového přepisu na hlavní části formuláře. Lze změnit velikost písma, velikost a přítomnost fotografie mluvčího.

Zvolené nastavení je uchováno i po ukončení aplikace. Všechny parametry jsou ukládány (a později načítány) do souboru formátu XML (viz. kapitola 5.3).

4.2.5 Formulář nápovědy

Jelikož je v programu použito více klávesových zkratek (viz. Příloha A), pro jejichž zapamatování je nutná delší práce s programem, je seznam zkratek s popisem co každá z nich znamená zobrazen na formuláři *Nápověda*. Tato jednoduchá nápověda je přístupná z položky hlavního menu (Nápověda – Popis Programu).

Tento formulář je možno zobrazit společně s hlavním formulářem programu, aby byl seznam klávesových zkratek k dispozici i při běžné práci (lze dále pracovat s hlavním formulářem).

5 Reprezentace textového přepisu v programu

5.1 Struktura textového přepisu

Protože přepisované řečové dokumenty mohou být poměrně komplikované, tak je vhodné vytvářený textový přepis rozčlenit do struktury. Členění umožňuje zpřehlednit vytvářený přepis, což je důležité při přepisování řečových dokumentů, které obsahují více mluvčích a kde se střídá mnoho tématických celků. Tím je zajištěna přehlednost a variabilnost vytvářeného přepisu.

Aby byl vytvářený textový přepis audio souboru dostatečně přehledný, bylo zvoleno jeho členění, které je popsáno dále:

a) První a zároveň nejvyšší úrovní celého přepisu může být jeho název (*transcript*). Jsou zde také uloženi všichni mluvčí, kteří v přepisu vystupují. K této úrovni nejsou přiřazeny žádné synchronizační značky, protože popisuje celý dokument.

b) Druhou úrovní textového přepisu je členění do kapitol (*chapters*). Každá kapitola vyjadřuje v přepisu větší tématickou část. Dokument musí obsahovat alespoň jednu kapitolu a jejich celkový počet není nijak omezen. Kapitola obsahuje název a dále je jí možné přiřadit časové synchronizační značky začátku a konce. Začátky a konce dvou různých kapitol se nesmějí překrývat.

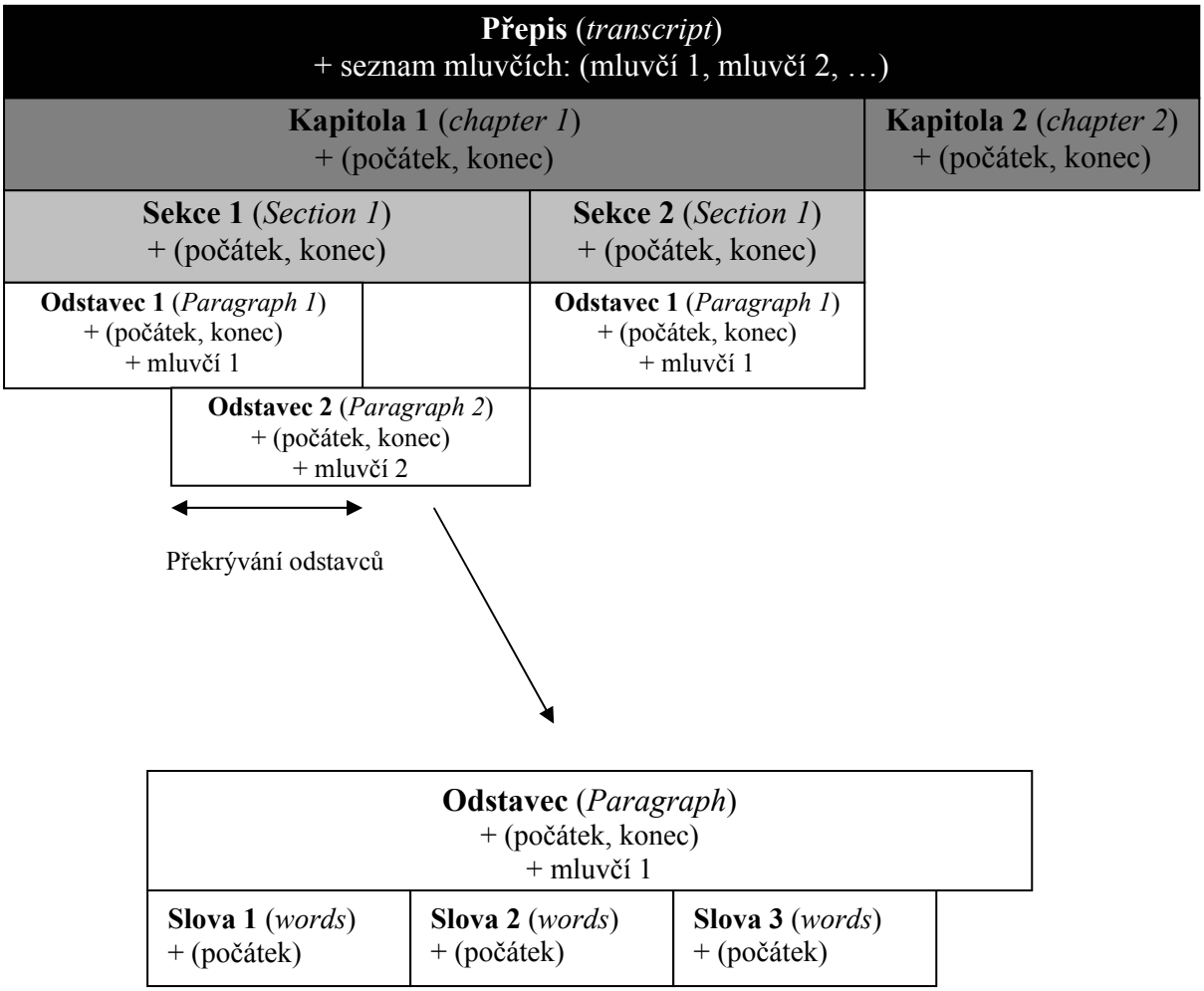
c) Třetí úrovní přepisu je členění do sekcí (*sections*). Každá kapitola je tak rozdělena do libovolného počtu sekcí. Sekce vyjadřuje v přepisu menší tématickou část než kapitola a slouží k upřesnění tématu. Každá sekce obsahuje vlastní název a stejně jako v případě kapitoly jí je možné přiřadit časové synchronizační značky začátku a konce. Začátky a konce dvou různých sekcí se opět nesmějí překrývat. Každá sekce může obsahovat výchozího mluvčího (viz. bod (f)), jeho použití je však nepovinné a primárně se nastavují mluvčí ve čtvrté úrovni přepisu.

d) Čtvrtou úrovní textového přepisu je členění do odstavců (*paragraphs*). Každá sekce je rozdělena do libovolného počtu odstavců. Odstavec byl zvolen jako nejnižší úroveň, která je v programu zobrazena samostatným objektem (pomocí komponenty *TextBox*). Odstavci je možno přiřadit synchronizační značky počátku a konce a má rovněž přiřazeného svého mluvčího (viz. bod (f)). U dvou různých navazujících odstavců (každý s různým mluvčím) je umožněno, aby se překrývaly konec prvního se začátkem druhého. Tato vlastnost umožňuje, aby se

dva mluvčí mohli překrývat v případech, kdy před skončením promluvy jednoho mluvčího začne již mluvit další.

e) Pátou úrovní textového přepisu je skutečnost, že odstavec je dále vnitřně rozčleněn do ještě menších textových úseků (*words*), které mohou obsahovat libovolný počet slov a také synchronizační značku počátku. Pokud odstavec obsahuje nějaký text, musí se vnitřně skládat alespoň z jednoho textového úseku. Takovéto členění umožňuje zaznamenat přesné časové indexy u vět případně samotných slov. Tím je zajištěna přesná provázanost mezi řečovým dokumentem a textovým přepisem. Této vlastnosti je převážně využito ve spojením s technologií *v2t* (viz. kapitola 9), protože při ruční transkripci by bylo velice zdlouhavé určovat přesné počátky jednotlivých slov. Avšak pokud je text odstavce příliš dlouhý, je samozřejmě možno vkládat časové synchronizační značky na libovolné místo ručně.

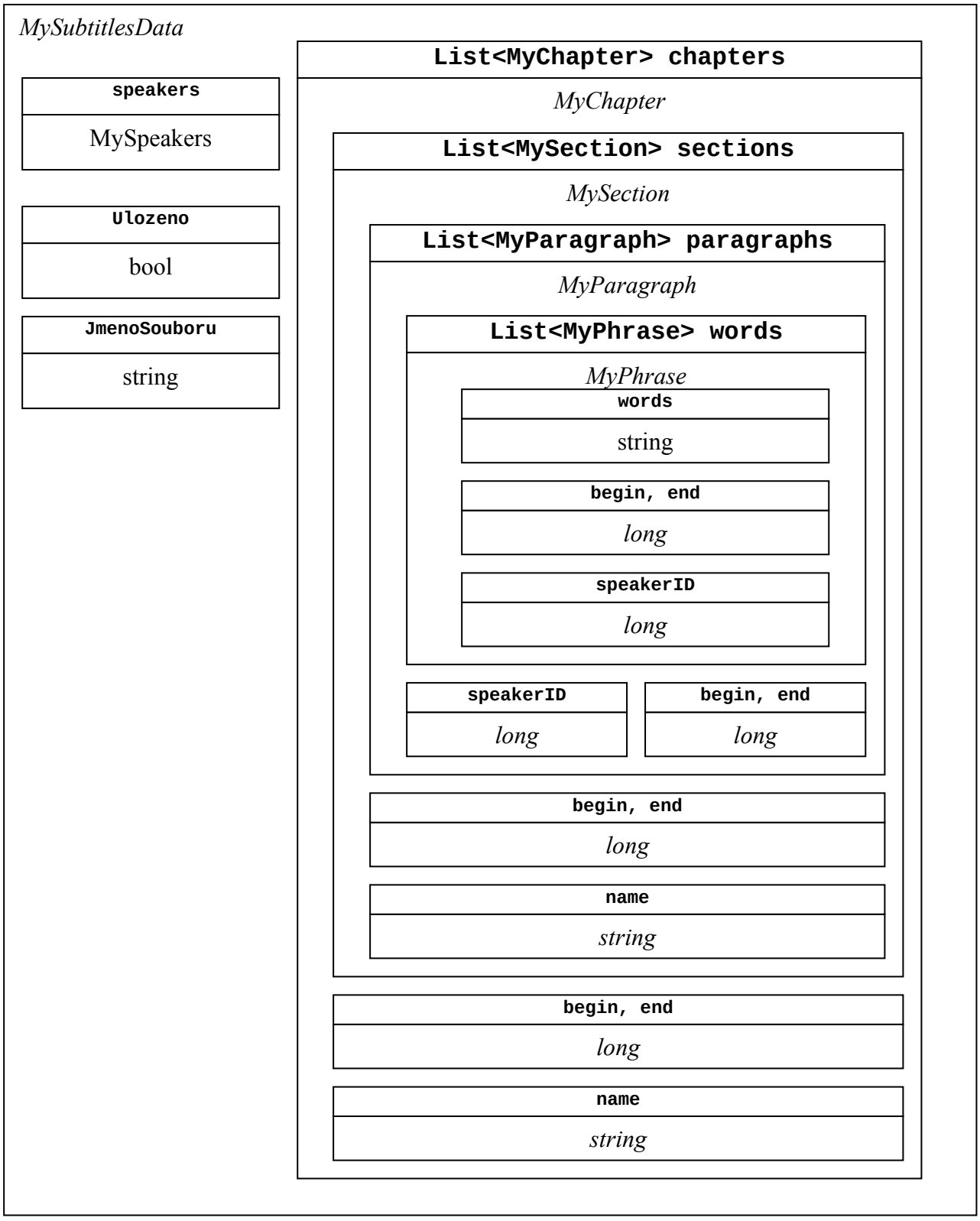
f) Mluvčí: Každému odstavci lze přiřadit mluvčího (osobu, která v přepisovaném audio souboru vyslovuje danou část textu). Všichni mluvčí, kteří se v přepisu vyskytují, jsou uloženi v nejvyšší úrovni v seznamu mluvčích. To zajišťuje přehled nad všemi mluvčími přepisu. Každý mluvčí je v přepisu identifikován unikátním číslem, které je přiřazováno jednotlivým odstavcům a sekcím. Toto řešení bylo zvoleno z důvodu, že při editaci mluvčího stačí pouze změnit jeho vlastnosti na jednom místě a nemusí se měnit v ostatních částech přepisu (odstavce a sekce). Mluvčí má svoji vlastní datovou strukturu a mohou se o něm ukládat různé informace jako je jeho jméno, obrázek a charakteristika (věk, pohlaví, použité jazykové modely a přepisovací pravidla pro technologii *v2t*) Další informace o mluvčích jsou popsány v kapitole 6.



Obrázek 5.1: Grafická podoba struktury textového přepisu

5.2 Datová struktura textového přepisu

Textový přepis vytvářený v aplikaci, musí být reprezentován pomocí datové struktury, se kterou dokáže aplikace dále pracovat. Datová struktura byla navržena podle modelu struktury textového přepisu (viz. kapitola 5.1). Celou datovou strukturu textového přepisu obsahuje třída *MySubtitlesData*. Tato třída obsahuje všechny potřebné objekty, které jsou reprezentací úrovní modelu textového přepisu. Schéma třídy *MySubtitlesData* je vidět na následujícím obrázku (Obrázek 5.2).



Obrázek 5.2: Datová struktura reprezentující textový přepis – třída *MySubtitlesData*

Kromě datové struktury samotného textového přepisu obsahuje třída *MySubtitlesData* také funkce, pomocí kterých jsou přidávány, editovány a vraceny položky datové struktury. Seznam a popis vybraných funkcí, pomocí kterých je datová struktura upravována, je uveden v příloze D.1.

5.3 Ukládání a načítání textového přepisu do (ze) souboru

Aplikace musí být schopna ukládat a načítat vytvořené přepisy řečových dokumentů. Základní myšlenkou bylo umožnit použít textové přepisy i v jiných aplikacích bez nutnosti znát přesnou binární strukturu přepisu. Proto byl pro uložení zvolen textový formát souboru v podobě XML (eXtensible Markup Language). XML má výhodu, že ho lze v případě potřeby jednoduše transformovat do jiné podoby, kterou bude možno využít v jiných aplikacích.

Pro vytvoření souboru formátu XML byl využit nástroj *XMLSerializer*, který je obsažen v platformě .NET Framework a v programovacím jazyce C#. Tento nástroj umožňuje transformovat datovou třídu do podoby XML a poté ji opět z XML načíst do paměti. Aby bylo možno třídu serializovat do požadované podoby, musí být u jednotlivých proměnných třídy nastaveno jakým způsobem budou uloženy v XML souboru (např. zda se jedná o XML element nebo atribut). Výsledné XML má stejný tvar, jako navržená struktura textového přepisu (viz. kapitola 5.1). Příklad XML souboru je uveden na obrázku 5.3.

O uložení datové struktury do souboru a jeho opětovné načtení se stará třída *MySubtitlesData*. Třída obsahuje dvě funkce, které toto zajišťují (*Serializovat(), Deserializovat()*). Jejich hlavičky a popis jsou uvedeny v příloze D.1. Při ukládání textového přepisu, je možno specifikovat, zda bude společně se seznamem mluvčích uložen do souboru i obrázek v textovém formátu (viz. kapitola 6.1). Uložení obrázku je nepovinné kvůli snížení velikosti souboru s textovým přepisem.

```
<?xml version="1.0" encoding="utf-8"?>
<MySubtitlesData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd=…
  <chapters>
    <MyChapter begin="-1" end="-1">
      <name>Kapitola 0</name>
      <sections>
        <MySection begin="-1" end="-1">
          <name>Sekce 0</name>
          <paragraphs>
            <MyParagraph begin="0" end="-1">
              <phrases>
                <MyPhrase begin="-1" end="-1">
                  <words>text1</words>
                </MyPhrase>
              </phrases>
              <speaker>1</speaker>
            </MyParagraph>
          </paragraphs>
          <speaker>0</speaker>
        </MySection>
      </sections>
    </MyChapter>
  </chapters>
  <SeznamMluvcich>
    <speakers>
      <MySpeaker>
        <ID>1</ID>
        <Name>Mluvčí 1</Name>
        <RozpoznavacMluvci>male.amd</RozpoznavacMluvci>
        <Poznamka />
      </MySpeaker>
      <MySpeaker>
        <ID>2</ID>
        <Name>Mluvčí 2</Name>
        <RozpoznavacMluvci>female.amd</RozpoznavacMluvci>
        <Poznamka />
      </MySpeaker>
    </speakers>
  </SeznamMluvcich>
</MySubtitlesData>
```

Obrázek 5.3: Podoba souboru s textovým přepisem ve formátu XML

6 Mluvčí

Při zpracování řečových dokumentů, je problematika mluvčích velice důležitá. Proto byla mluvčím vyhrazena samostatná kapitola.

Mluvčí je osoba, která v přepisovaném řečovém dokumentu vyslovuje danou část textu. Každý řečový dokument musí mít alespoň jednoho mluvčího.

6.1 Potřebné informace o mluvčích

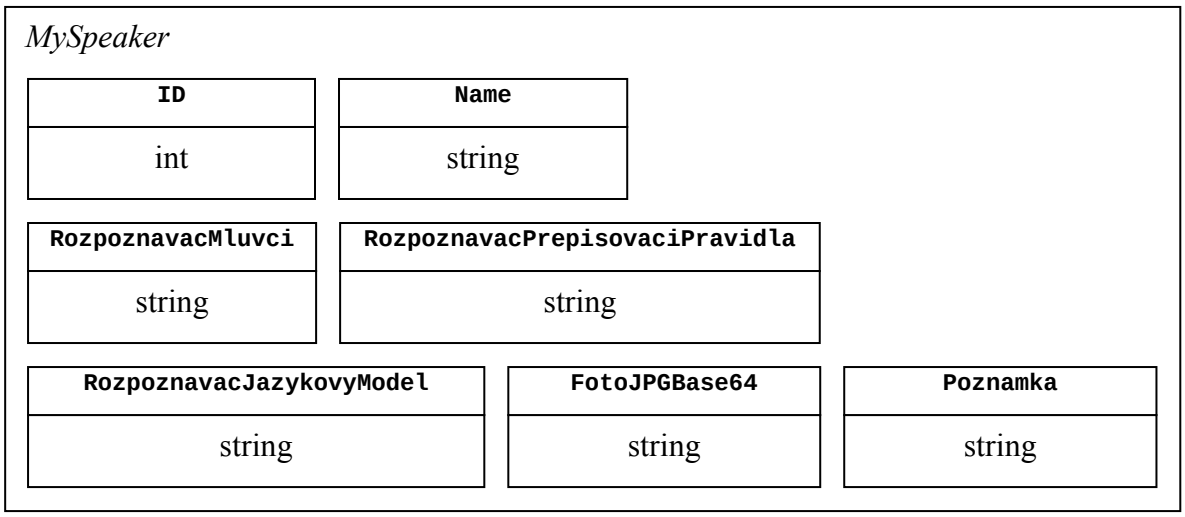
Aby bylo možno zefektivnit vytváření textových přepisů řečových dokumentů, je vhodné mít o každém mluvčím, který se v dokumentu vyskytuje, některé vhodné informace. Při vytváření textového přepisu pak již není nutné pokaždé specifikovat, kdo kterou část textu vyslovuje. Základní charakteristickou informací o mluvčím je jeho *jméno*. Jméno musí být unikátní kvůli jednoznačné identifikaci mluvčího.

Při vytváření přepisů složitějších řečových dokumentů, kde se vyskytuje mnoho mluvčích současně, je vhodné ukládat i další informace, které usnadní jeho identifikaci. Jsou jimi: *obrázek* (vhodný zejména při přítomnosti videa) a jeho charakteristika: *pohlaví* a *poznámka* (umožňuje uložit libovolné doplňkové informace o mluvčím – např. věk). Z důvodu implementace technologie *v2t* (viz. kapitola 9) je vhodné o mluvčích uchovávat příslušný *jazykový model, akustický model* a *přepisovací pravidla*.

Pro potřeby aplikace bylo proto navrženo o každém mluvčím uchovávat následující informace v příslušné datové struktuře (*MySpeaker* – popsaná níže):

Jméno; Poznámka – podrobnější popis mluvčího; Akustický model ~ pohlaví; Jazykový model (příslušný slovník technologie *v2t*); Přepisovací pravidla (pro formátování textového výstupu při automatickém přepisu při použití technologie *v2t*); Obrázek (fotografie, příslušného mluvčího, která usnadní jeho identifikaci při přehrávání video souboru).

Uchovávání informací o mluvčím zajišťuje v programu datová třída *MySpeaker*. Obsahuje potřebné proměnné pro uložení požadovaných informací. Její struktura je zobrazena na obrázku 6.1.



Obrázek 6.1 Datová třída – *MySpeaker* (Informace o mluvčím)

6.2 Interní databáze mluvčích

O jednotlivých mluvčích, kteří se v přepisu mohou vyskytovat se ukládá větší množství informací (viz. kapitola 6.1). Pokud jsou k dispozici podrobnější informace o mluvčím, je to většinou z toho důvodu, že se konkrétní mluvčí vyskytuje ve více řečových dokumentech (Typickým příkladem je například zpravodajská relace). V takovém případě je zbytečné, aby se při každém novém přepisování řečového dokumentu musely znovu zadávat informace o stejném mluvčím. Proto je v programu zavedena podpora interního seznamu mluvčích, kteří se v přepisech mohou vyskytovat.

Pro ukládání informací o mluvčích je využita (stejně jako v případě ukládání textového přepisu) serializace do formátu XML (viz. kapitola 5.3). Formát XML byl opět zvolen z důvodu použitelnosti v případných dalších aplikacích. Vytvářená aplikace umožňuje načítat a ukládat interní databázi do jednotlivých souborů. Tím je zajištěna možnost pracovat pouze s daným okruhem mluvčích, kteří jsou pro daný přepis potřeba.

Protože je pro uložení informací o mluvčích využito textového formátu XML, není do něho možno ukládat přímo binární data. Vytvářená aplikace proto pro uložení obrázku mluvčího (který je ve formátu JPG) využívá překódování binárních dat do kódování *Base64* – jedná se kódování binárních dat do textového formátu ASCII. Výsledkem překódování je textový řetězec, který již lze uložit do formátu XML.

```
<?xml version="1.0" encoding="utf-8"?>
<MySpeakers xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <speakers>
    <MySpeaker>
      <ID>1</ID>
      <Name>Mluvčí 1</Name>
      <RozpoznavacMluvci>male.amd</RozpoznavacMluvci>
      <RozpoznavacJazykovyModel>spoken.bin</RozpoznavacJazykovyModel>
      <RozpoznavacPrepisovaciPravidla>…ppp</RozpoznavacPrepisovaciPravidla>
      <FotoJPGBase64>/9j/4AAQSJRgABAQEBwYIDAoMDAsKCw…</FotoJPGBase64>
      <Poznamka />
    </MySpeaker>
    <MySpeaker>
      <ID>2</ID>
      <Name>Mluvčí 2</Name>
      <RozpoznavacMluvci>female.amd</RozpoznavacMluvci>
      <Poznamka />
    </MySpeaker>
  </speakers>
</MySpeakers>
```

Obrázek 6.2: Struktura souboru se seznamem mluvčích

O seznam mluvčích se v aplikaci stará datová třída *MySpeakers*. Obsahuje všechny potřebné funkce pro správu (přidání, editaci, vyhledávání) seznamu mluvčích a jeho ukládání a načítání do (ze) souboru. Popis vybraných funkcí je uveden v příloze D.2.

.

7 Podpora multimediálních souborů

Pro vytvoření přepisu multimediálního dokumentu, musí být aplikace schopna načíst a přehrávat audio soubory. Do programu byla zahrnuta podpora většiny multimediálních formátů (viz. kapitola 7.3), aby byla aplikace maximálně flexibilní.

7.1 Interní zpracování audio souboru

Interně pracuje program se zvukovým formátem WAV (viz. kapitola 7.2). Je to z důvodu jeho jednoduché struktury, protože v souborech formátu WAV, jsou audio data uložena v nekomprimované podobě. Toho je využito při přehrávání zvuku (viz. kapitola 7.4.2), při grafickém zobrazení audio dat (viz. kapitola 8) a pro potřeby automatického rozpoznávání řeči z multimediálních souborů – implementovaná technologie *v2t* (viz kapitola 9). Z důvodu požadavků technologie *v2t*, pracuje vytvářená aplikace interně s formátem WAV, který má následující parametry: velikost vzorku – 16 bitů, frekvence vzorkování – 16 kHz, počet zvukových kanálů – 1 (mono).

7.2 Zvukový formát WAV

7.2.1 Základní informace

WAV je zkratka z anglického *Waveform Audio Format*. Tento zvukový formát byl vytvořen firmami Microsoft a IBM pro ukládání zvuku. Do WAV souboru lze ukládat data i v komprimované podobě (např. MP3), ale tato možnost se většinou nevyužívá. Ve WAV souboru je tak zvuk nejčastěji uložen bezztrátově pomocí pulzně kódové modulace (PCM). Proto je WAV nejčastěji používaným formátem při zpracování zvuku.

7.2.2 Hlavička *.wav souboru

Aplikace pracuje se zvukem ve formátu WAV (práce s dočasnými soubory, grafické zobrazení audio dat, přehrávání zvuku, záznam zvuku, rozpoznávání řeči). Proto je nutné znát strukturu hlavičky *.wav souboru, která obsahuje potřebné informace o audio formátu. Struktura hlavičky je zobrazena na obrázku 7.1.

Řetězec „RIFF“				*Počet bytů do konce souboru*				*Řetězec „WAVE“*			
1	2	3	4	5	6	7	8	9	10	11	12
R	I	F	F	A1 L	A1	A1	A1 H	W	A	V	E
Řetězec „fmt “				*Poč. bytů do konc. části format*				*Formát dat*		*Počet kanálů*	
13	14	15	16	17	18	19	20	21	22	23	24
F	m	T		AF L	AF	AF	AF H	K L	K H	CH L	CH H
Vzorkovací frekvence [Hz]				*Počet bytů/s*				*Vel. Vzorku [B]*		*Vel. Vzorku [b]*	
25	26	27	28	29	30	31	32	33	34	35	36
VF L	VF	VF	VF H	PB L	PB	PB	PB H	VB L	VB H	VV L	VV H
Řetězec „data“				*Počet bytů do konce souboru*							
37	38	39	40	41	42	43	44				
D	a	T	a	A2 L	A2	A2	A2 H				

Obrázek 7.1: Hlavička souboru formátu WAV

7.3 Převod multimediálních souborů na podporovaný formát

Aplikace interně podporuje formát WAV (viz. kapitoly 7.1 a 7.2). Podpora ostatních multimediálních formátů je řešena pomocí externího programu *Ffmpeg* (viz. kapitola 7.3.1). Aplikace je tak schopna přehrávat zvuk všech audiovizuálních formátů, které podporuje právě *Ffmpeg*. Seznam podporovaných formátů je dostupný po spuštění utility *ffmpeg.exe* (viz. kapitola 10.2) s parametrem *–formats*.

7.3.1 Ffmpeg

Jedná se o kompletní řešení, které umožňuje nahrávat, konvertovat a streamovat různé digitální audio a video formáty. Celý projekt Ffmpeg se skládá ze tří základních součástí: *ffmpeg* (utilita pro příkazovou řádku určená pro nahrávání a převod multimediálních souborů), *ffserver* (streamování multimediálních souborů) a *ffplay* (přehrávač multimediálních souborů).

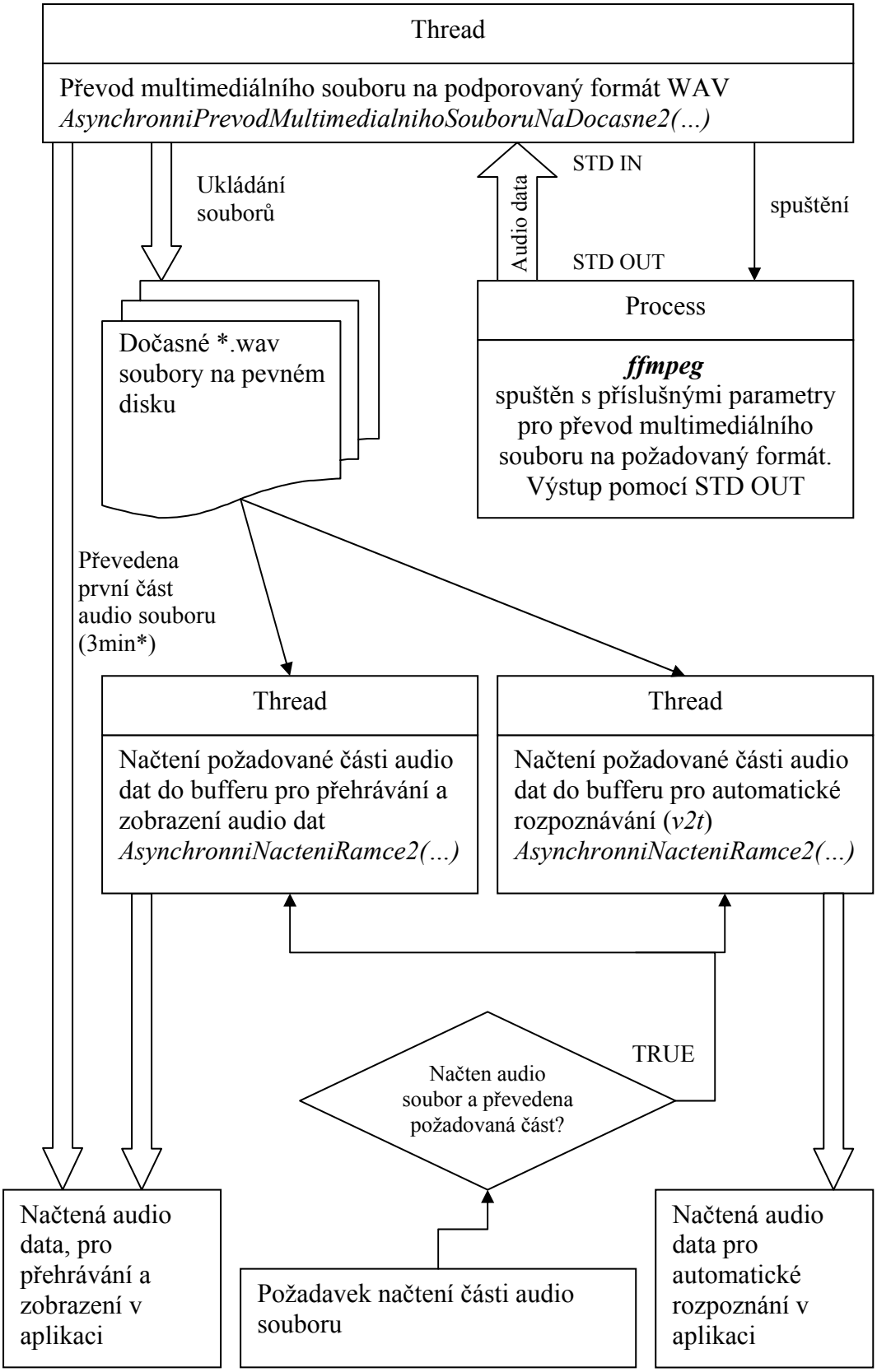
Vytvářená aplikace využívá část *ffmpeg* k převodu do požadovaného formátu. Program *ffmpeg* umožňuje komunikovat s ostatními programy pomocí standardního vstupu a výstupu, což je v programu využito. *Ffmpeg* je spuštěn jako proces s parametry a standardní výstup je přesměrován do aplikace (viz. obrázek 7.2).

Převáděná data jsou poté přímo zpracována aplikací: Původní multimediální soubor je během převodu pomocí *ffmpeg* ukládán v minutových dočasných *.wav souborech na pevný disk počítače. *Ffmpeg* posílá pouze převedená audio data v příslušném formátu, hlavička souboru *.wav (viz.kapitola 7.2.2) je pak vytvořena aplikací při uložení dočasného souboru.

Toto řešení bylo zvoleno proto, aby nebylo nutné uchovávat celý nekomprimovaný WAV v paměti (1 hodina audio záznamu zabere přibližně 100 MB operační paměti). Rozdělení na více souborů pak umožňuje pracovat s již převedenými daty, zatímco na pozadí běží konverze zbylé části multimediálního souboru. Další vysvětlení, jakým způsobem aplikace pracuje s multimediálními soubory je popsán v další kapitole.

7.3.2 Implementace načítání multimediálních souborů v aplikaci

Práci s převodem a dočasnými *.wav soubory má v aplikaci na starost třída *MyWav*. Převod na podporovaný formát (viz. kapitola 7.3) i načítání dočasných souborů je řešeno asynchronně pomocí více threadů a nebrzdí tak aplikaci v jiné činnosti. Způsob jakým je převáděn soubor na dočasné a jak jsou data spravována je znázorněn na obrázku 7.2. Třída *MyWav* obsahuje důležité funkce a property, které jsou popsány v příloze D.3.



Obrázek 7.2 Zpracování multimediálního (audio) souboru

7.4 Způsob přehrávání multimediálních souborů

7.4.1 Přehrávání audio a video souborů

Aby bylo možno vytvářet textový přepis audio souborů, musí aplikace umožňovat načítání a přehrávání přepisovaného multimediálního (audio a video) souboru. Aplikace podporuje přehrávání většiny běžných formátů souborů (popsáno v kapitole: 7.3) Způsob přehrání je popsán v následující kapitole (7.4.2).

Aplikace také umožňuje přehrát společně s audio souborem i video. Velká část materiálů, které je potřeba přepisovat, totiž obsahují kromě zvukové části i video. Video zjednodušuje identifikaci jednotlivých mluvčích, kteří se v přepisovaných materiálech vyskytují. Způsob přehrávání video souboru v aplikaci je popsán v kapitole 7.4.3.

7.4.2 Způsob přehrání audio souboru

Pro přehrávání audio souborů v programu, byly uvažovány dva způsoby, které jsou dále popsány.

a) Přehrávání pomocí komponenty MediaElement

Komponenta *MediaElement*, která je součástí WPF (Windows Presentation Foundation; viz. kapitola 3.2), umožňuje přehrávání většiny audio a video souborů. Přehrávat je možno libovolnou část souboru, přehrávání lze pozastavovat a lze se přesouvat na určenou pozici. Jedná se o jednoduchý způsob, jak přehrávat audiovizuální soubory, který však přináší několik problémů: Při přehrání souborů, které obsahují i video část, je z důvodu komprimace problematické přehrávat určité úseky (například samotné slovo). Posuny v souboru na požadovanou pozici jsou také pomalé a zpomalují práci aplikace. V případě přehrávání zvuku z tohoto souboru by tak mohlo docházet ke vznikům odchylek mezi pozicí kurzoru v graficky zobrazených audio datech a přehrávaným zvukem.

Optimálním řešením je přehrávat nekomprimovaná audio data, která má program k dispozici pro vykreslování vlny (viz. kapitola 8) a technologii *v2t* (viz. kapitola 9). K dispozici jsou minutové soubory formátu WAV. Ovšem při jejich přehrávání pomocí komponenty *MediaElement*, dochází k nežádoucímu praskání ve zvuku při navazování souborů. Dalším nežádoucím jevem je pomalé načítání souborů při rychlých posunech v rámci celého audio souboru.

b) Přehrávání pomocí knihovny winmm.dll

Knihovna *winmm.dll* je součástí operačního systému Microsoft Windows a obsahuje API (aplikační rozhraní – funkce) pro práci s multimédii. Umožňuje například přehrávání a nahrávání nekomprimovaných audio dat.

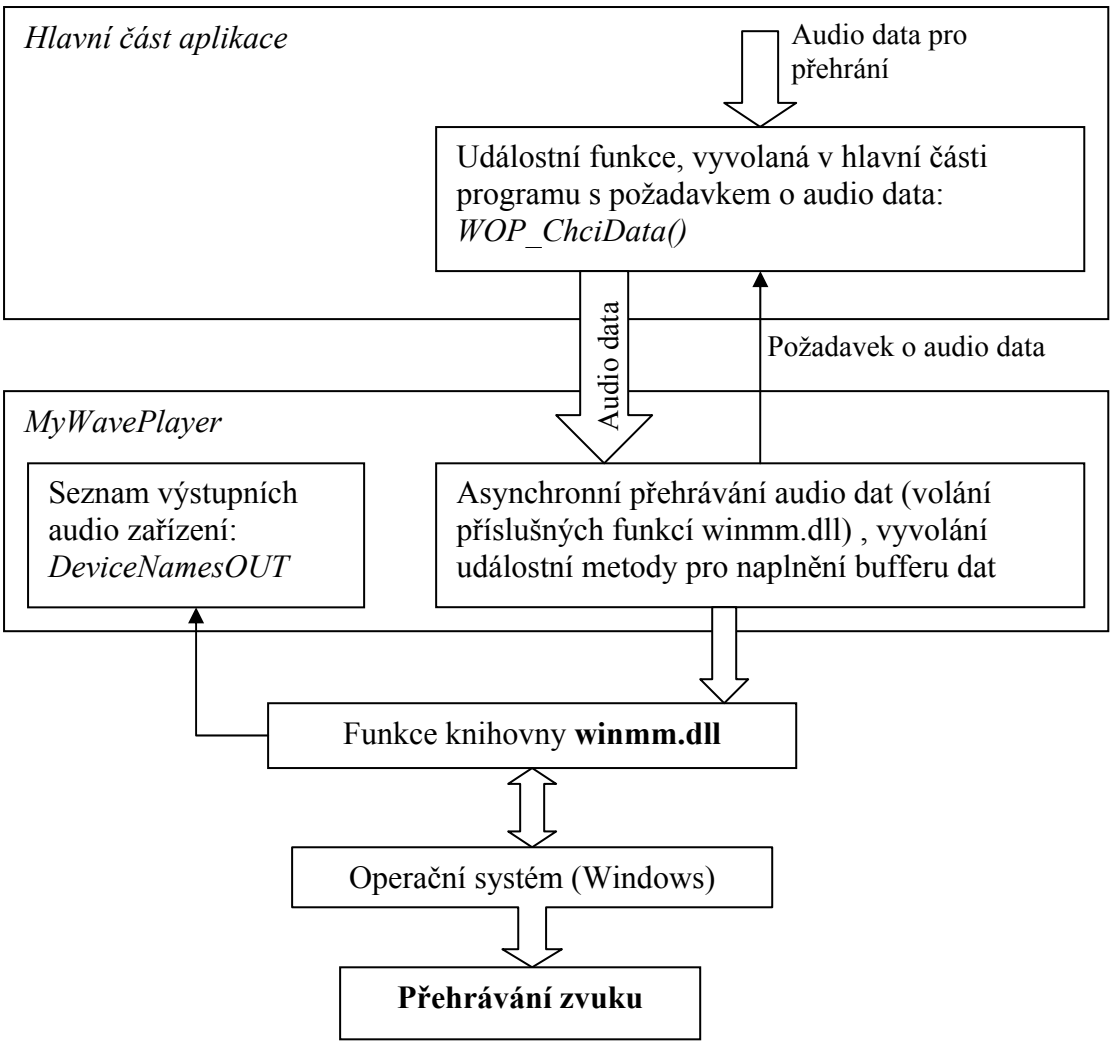
Tento způsob přehrávání byl zvolen, protože přehrávání multimediálních souborů pomocí komponenty *MediaElement*, popsané v předchozí části (a) nepřineslo potřebné výsledky (viz. problémy s přehráváním).

Přehrávána jsou interní nekomprimovaná data programu ve formátu WAV (viz. kapitola 7.2). Přehrávání má v aplikaci na starost třída *MyWavePlayer*. Obsahuje potřebné funkce, které využívá aplikace pro přehrávání dat a pro konfiguraci výstupního audio zařízení. Důležité funkce a property jsou popsány níže. Komunikace mezi jednotlivými částmi aplikace při přehrávání je zobrazena na obrázku 7.3.

`public static string[] DeviceNamesOUT` – property, která vrací seznam všech výstupních zařízení daného počítače, která jsou schopna přehrávat nekomprimovaný WAV formát. Toto property je použito pro zobrazení a konfiguraci výstupního audio zařízení.

`public MyWavePlayer(int device, WaveFormat format, int bufferSize, int bufferCount, BufferFillEventHandler fillProc)` – Constructor, pomocí proměnných dojde ke konfiguraci a inicializaci přehrávání: Zařízení pro přehrávání; formát WAV (frekvence, velikost vzorků, mono/stereo); velikost bufferu dat, která jsou přehrávána; počet bufferů; Událostní funkce, která je volána pokud jsou požadována audio data pro přehrávání.

`void WOP_ChciData(IntPtr data, int size)` – Funkce je deklarována v hlavní části programu (*Window1*). Je automaticky volána (událostní funkce) při požadavku objektu přehrávače (*MyWavePlayer*) o nová audio data. Pokud dochází k přehrávání je v intervalu 150 ms naplněn buffer a jeho obsah přehrán. Interval 150 ms byl zvolen, aby bylo možno přehrávání přerušit. Přerušení přehrávání je řešeno posíláním 0 do bufferu.



Obrázek 7.3: Způsob přehrávání zvuku v aplikaci

7.4.3 Způsob přehrání a zobrazení video souboru

Pro přehrání videa souboru byla využita komponenta *MediaElement* (viz.kapitola 7.4.2). Video je přehráváno nezávisle na audio souboru z důvodů popsaných v téže kapitole. Přehráváno je pouze video se ztlumeným zvukem a pozice ve video souboru je synchronizována s pozicí přehrávaných audio dat. Na pomalejších počítačích může někdy při přehrávání docházet k rozcházení zvuku a obrazu, proto pokud se obraz a zvuk rozejdou o určitý časový úsek (200 ms), je video znovu synchronizováno vzhledem ke zvuku.

7.5 Způsob záznamu zvuku v aplikaci

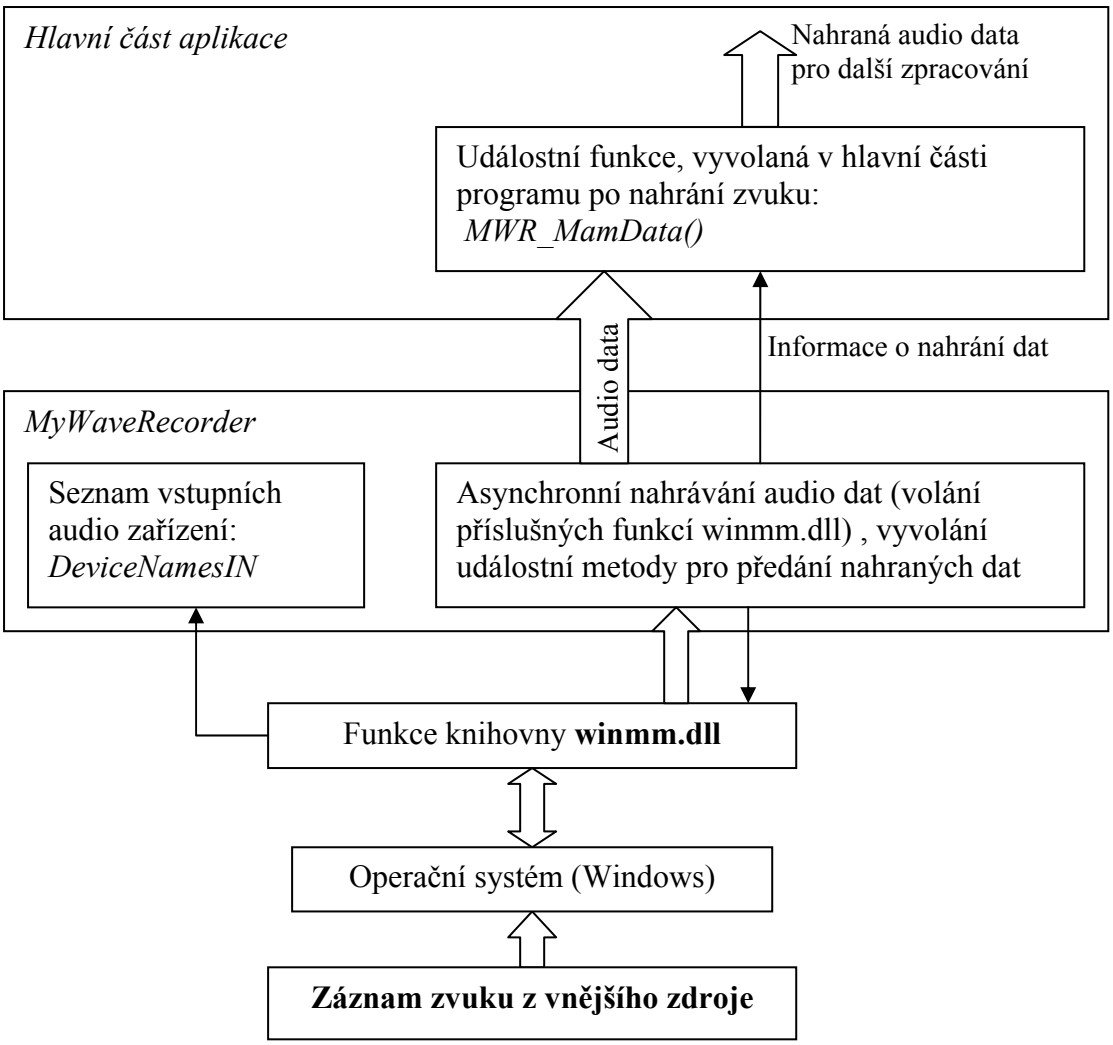
Vytvářená aplikace umožňuje použití technologie *v2t* (viz. kapitola 9) – automatický přepis diktované řeči do textové podoby a také podporuje hlasové ovládání některých funkcí programu. Proto je potřeba zaznamenávat hlas uživatele aplikace pomocí mikrofonu. Zvuk je nutné zaznamenávat v nekomprimované podobě v požadovaném formátu, který aplikace kvůli technologii *v2t* využívá (16 kHz, mono, 16 bitů).

Nahrávání zvuku je řešeno stejně jako v případě přehrávání pomocí knihovny *winmm.dll* (viz.kapitola 7.4.2, část (b)), která obsahuje všechny potřebné nástroje pro záznam. Záznam zvuku má v aplikaci na starost třída *MyWaveRecorder*. Obsahuje potřebné funkce, které využívá aplikace pro nahrávání zvukových dat a pro konfiguraci vstupního audio zařízení počítače. Důležité funkce a property jsou popsány níže. Způsob komunikace mezi jednotlivými částmi (objekty) aplikace je zobrazen na obrázku 7.4.

`public static string[] DeviceNamesIN` – property, která vrací seznam všech vstupních zařízení daného počítače, která jsou schopna nahrávat nekomprimovaný WAV formát. Toto property je použito pro zobrazení a konfiguraci vstupního audio zařízení.

`public MyWaveRecorder(int device, WaveFormat format, int bufferSize, int bufferCount, BufferDoneEventHandler doneProc)` – Constructor, pomocí proměnných dojde ke konfiguraci a inicializaci nahrávání: Zařízení pro nahrávání; formát WAV (frekvence, velikost vzorků, mono/stereo); velikost bufferu dat, která jsou nahrána; počet bufferů; Událostní funkce, která je volána pokud jsou k dispozici nahrána audio data.

`void MWR_MamData(IntPtr data, int size)` – Tato funkce je deklarována v hlavní části programu (*Window1*) a je automaticky volána (událostní funkce) po nahrání požadovaných audio dat. Dodává audio data přímo automatickému rozpoznávači spojité řeči (viz. kapitola 9) pomocí jeho rozhraní. Audio data jsou posílána buď při diktování nebo při hlasovém ovládání aplikace.



Obrázek 7.4: Způsob záznamu zvuku aplikací

8 Grafické zobrazení zvukových dat

Pro usnadnění práce při vytváření přepisu audio souboru, je vhodné zobrazit grafickou reprezentaci audio dat, která jsou přehrávána. Grafické zobrazení zjednodušuje orientaci v audio souborech a umožňuje přesně určit začátky a konce jednotlivých slov, případně celých vět.

Pro grafické zobrazení audio souboru v podobě vlny bylo využito vektorové grafiky, kterou WPF (Windows Presentation Foundation, viz kapitola 3.2) nabízí. Ve WPF, je grafika zobrazována v tzv. *retained módu*. To znamená, že data, která je potřeba vykreslit, se jednou definují a pokud je potřeba část nebo celou scénu překreslit, stará se o toto samotné WPF bez nutnosti manuálně překreslovat požadovanou oblast v programu.

V aplikaci je vždy vykreslena pouze viditelná část vlny, protože vykreslení celých audio souborů by bylo zbytečně zdlouhavé a paměťově náročné.

8.1 Způsoby grafického zobrazení audio dat

Pro grafické zobrazení audio dat v podobě vlny, byly uvažovány 2 možnosti, které budou dále popsány.

8.1.1 Princip vynechávání vzorků

První způsob využívá toho, že při kreslení vlny nejsou kresleny všechny audio vzorky, které jsou k dispozici, ale zobrazen vždy každý *N*-tý vzorek. *N* je určeno podle následujícího vztahu (1):

$$N = round\left(\frac{sekundy}{4.5} \times \frac{frekvence}{1000000}\right) \quad (1)$$

sekundy – délka zobrazené části audio souboru v sekundách;

frekvence – počet audio vzorků za 1 s;

Konstanty byly určeny kompromisem tak, aby vykreslená vlna měla dostatečnou kvalitu a přitom její kreslení bylo dostatečně rychlé.

Pokud je potřeba vlnu rychle překreslovat – například při rychlých posunech pozice v audio souboru, je *N* dále dočasně zvětšeno, aby mohla být vlna vykreslena ještě rychleji. V tomto případě je kreslen každý *Nd*-tý vzorek. *Nd* je vypočítáno podle vztahu (2)

$$Nd = round(N \times 2.5) \quad (2)$$

N – kolikátý vzorek je vždy kreslen; *Konstanta* je určena tak, aby překreslování vlny probíhalo dostatečnou rychlostí.

Výhodou tohoto řešení je kvalitní zobrazení vlny krátkých časových úseků. Vlnu je možno věrohodně zobrazit s přesností jediného vzorku.

Tento způsob zobrazení má však několik nevýhod: Hlavním problémem je rychlost zobrazení, protože i přes to, že je počet vzorků k vykreslení snížen, dochází stále k zobrazování velkého množství bodů. Pro zobrazení je tak nutno všechny souřadnice bodů přepočítávat, což je zbytečně časově náročné. Další nevýhodou tohoto způsobu vykreslování vlny, je nedostatečná kvalita při zobrazení delších časových úseků než 1 minuta. Kvůli vynechání příliš mnoha vzorků, dochází k podstatným ztrátám informace o podobě audio souboru.

8.1.2 Princip průměrování

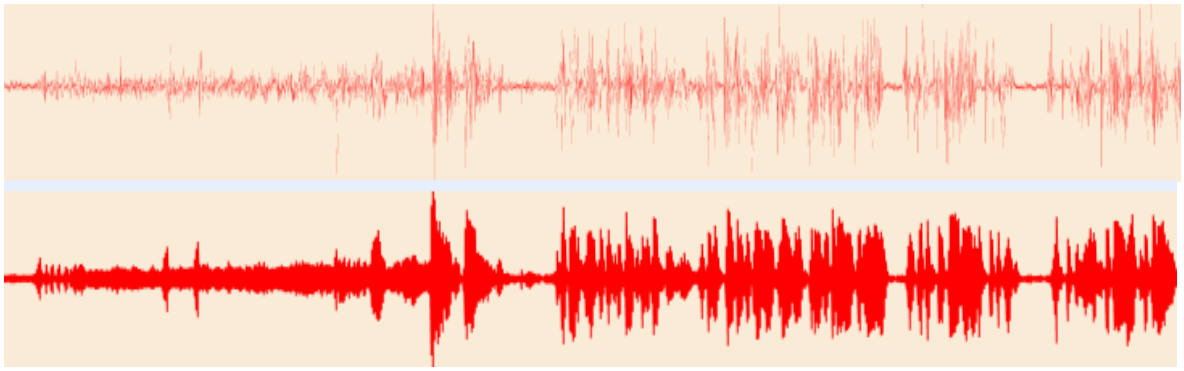
Druhý způsob zobrazení vlny byl naprogramován za účelem zrychlit vykreslování a zvýšit kvalitu. Počet vykreslovaných bodů vlny byl minimalizován podle počtu skutečně možných zobrazovaných bodů na obrazovce. Výsledná vlna se skládá z tolika svislých úseček umístěných těsně vedle sebe, kolik je skutečně obrazových bodů.

Souřadnice úseček jsou spočteny následujícím způsobem: souřadnice ve směru osy x, jsou dány jednotlivými obrazovými body. Souřadnice ve směru osy y jsou vypočteny jako průměry okolních vzorků, které by měly být zobrazeny. Samotné průměrování by nevedlo k požadovanému výsledku, protože navzorkované hodnoty audio signálu oscilují kolem nulové hodnoty (signál má střední hodnotu 0). Proto je třeba zprůměrovat zvlášť kladné a zvlášť záporné hodnoty audio signálu. Výstupem jsou 2 souřadnice y_1 a y_2 pro každou pozici x_i na obrazovce.

8.2 Porovnání řešení zobrazení audio dat

Tab. 8.1: Klady a zápory jednotlivých způsobů grafického zobrazení audio dat

Způsob řešení	Klady	Zápory
a) Princip vynechávání vzorků	- Jednoduchá implementace - Kvalitní zobrazení kratších časových úseků	- Pomalé a náročné překreslování audio dat - Nízká kvalita zobrazení delších časových úseků (>1 minuta) - nutnost snížit kvalitu zobrazení při rychlých posunech v audio souboru
b) Princip průměrování	- rychlost překreslování - výsledná kvalita vlny	- složitější přepočet vzorků - nemožnost velkých zvětšení



Obrázek 8.1: Porovnání způsobů grafického zobrazení audio dat
(nahoře – princip vynechávání vzorků; dole – princip průměrování)

Po zhodnocení výhod a nevýhod jednotlivých způsobů grafického zobrazení audio dat, bylo vybráno řešení spočívající v principu průměrování. Toto řešení přináší dostatečnou kvalitu zobrazení vlny i její rychlé překreslování.

8.3 Zobrazení vlny v aplikaci

Ve vytvářené aplikaci, je možno vlnu zobrazit v sedmi základních délkách: 5 sekund; 10 sekund; 20 sekund; 30 sekund; 1 minuta; 2 minuty; 3 minuty.

Zobrazení delšího časového úseku, než 3 minuty, již nemá význam, protože pak již nelze rozlišit počátky a konce jednotlivých úseků. Taktéž zobrazení kratšího úseku než je 5 sekund není pro určení počátků a konce slov nutné.

9 Technologie *v2t*

Technologie *v2t* (Voice to Text) byla vyvinuta laboratoří *Speechlab* (Laboratoř počítačového zpracování řeči; Ústav informačních technologií a elektroniky; Fakulta Mechatroniky, informatiky a mezioborových studií; Technická univerzita v Liberci). Umožňuje rozpoznávání spojité řeči (pro český jazyk) do textové podoby. Technologie je reprezentovaná hlavním spustitelným souborem a dalšími podpůrnými soubory (viz. kapitola 9.1). Komunikace s ostatními aplikacemi je řešena pomocí standardního vstupu respektive standardního výstupu. Komunikace funguje na principu zasílání zpráv určitého tvaru (Stručný popis komunikačního rozhraní je v kapitole 9.1).

Technologie *v2t* je schopna rozpoznávat audio data pouze daného formátu. Zvuková data musí být v nekomprimovaném formátu WAV (viz. kapitola 7.2) s následujícími parametry: frekvence vzorkování – 16 kHz; počet zvukových kanálů – 1; velikost vzorku – 16 bitů.

9.1 Spojení v2t – aplikace

Při požadavku na použití technologie *v2t* je aplikací spuštěn hlavní spustitelný soubor *nanocore.dll* jako samostatný proces s příslušnými parametry:

a) licenční soubor – pro každý počítač musí být k dispozici unikátní licenční soubor, který je ověřen na licenčním serveru. Proto je zapotřebí, aby měl počítač, na kterém je rozpoznávací software pouštěn, přístup na internet.
b) akustický model – podpůrný soubor, ve kterém jsou k dispozici charakteristika mluvčího, který má být rozpoznáván (například: jedná-li se o muže nebo ženu, případně může jít o akustický model konkrétního uživatele programu popřípadě jiné osoby).
c) jazykový model – nejdůležitější podpůrný soubor, který obsahuje slovník slov, které je schopen rozpoznávač rozpoznat.
d) přepisovací pravidla – podpůrný soubor, který obsahuje informace jakým způsobem má být prováděn postprocessing výsledného přepisu (například: přepisování číslovek, datumů, apod.)
e) adresa licenčního serveru – parametr, který určuje adresu licenčního serveru.

f) kvalita rozpoznávání – parametr, který určuje kvalitu a výslednou rychlost rozpoznávání. Protože je rozpoznávání velice náročný proces, musí být na pomalých počítačích snížena kvalita rozpoznávání, aby se zvýšila rychlost (například při diktování).
g) velikost vyrovnávací paměti – parametr, určující velikost vnitřní paměti rozpoznávače.

Pokud jsou k dispozici požadované podpůrné soubory (akustický model, jazykový model, přepisovací pravidla, licenční soubor) je ověřena licence pro daný počítač. Pokud je vše v pořádku, je na standardní výstup rozpoznávače poslána zpráva o úspěšné inicializaci. Poté je možno posílat rozpoznávači na standardní vstup následující zprávy:

a) *Spuštění rozpoznávání* – po úspěšném spuštění, vrátí rozpoznávač informaci a je připraven přijímat audio data.
b) *Zpráva s audio daty* – Po přijetí dat je spuštěno jejich rozpoznávání.
c) *Požadavek na rozpoznaný text* – rozpoznávač vrátí rozpoznaný text, pokud je k dispozici. Jsou vraceny 2 druhy zpráv: První s aktuálním odhadem rozpoznávání a druhá s finálně rozpoznaným textem včetně časových indexů v rozpoznávaných audio datech.
d) *Požadavek na aktuální zpoždění rozpoznávání* – rozpoznávač poté vrací informaci o zpoždění rozpoznávání (kolik dat k rozpoznání má ve vnitřním bufferu).

Z předchozího popisu komunikace s rozpoznávačem je patrné, že komunikace musí probíhat asynchronně – čtení i zápis standardního výstupu respektive vstupu. Proto je v programu pro čtení a zápis využito vláken (Threads). Způsob využití je popsán v následujícím popisu datové třídy *MyPrepisovac*.

9.1.1 Datová třída MyPrepisovac

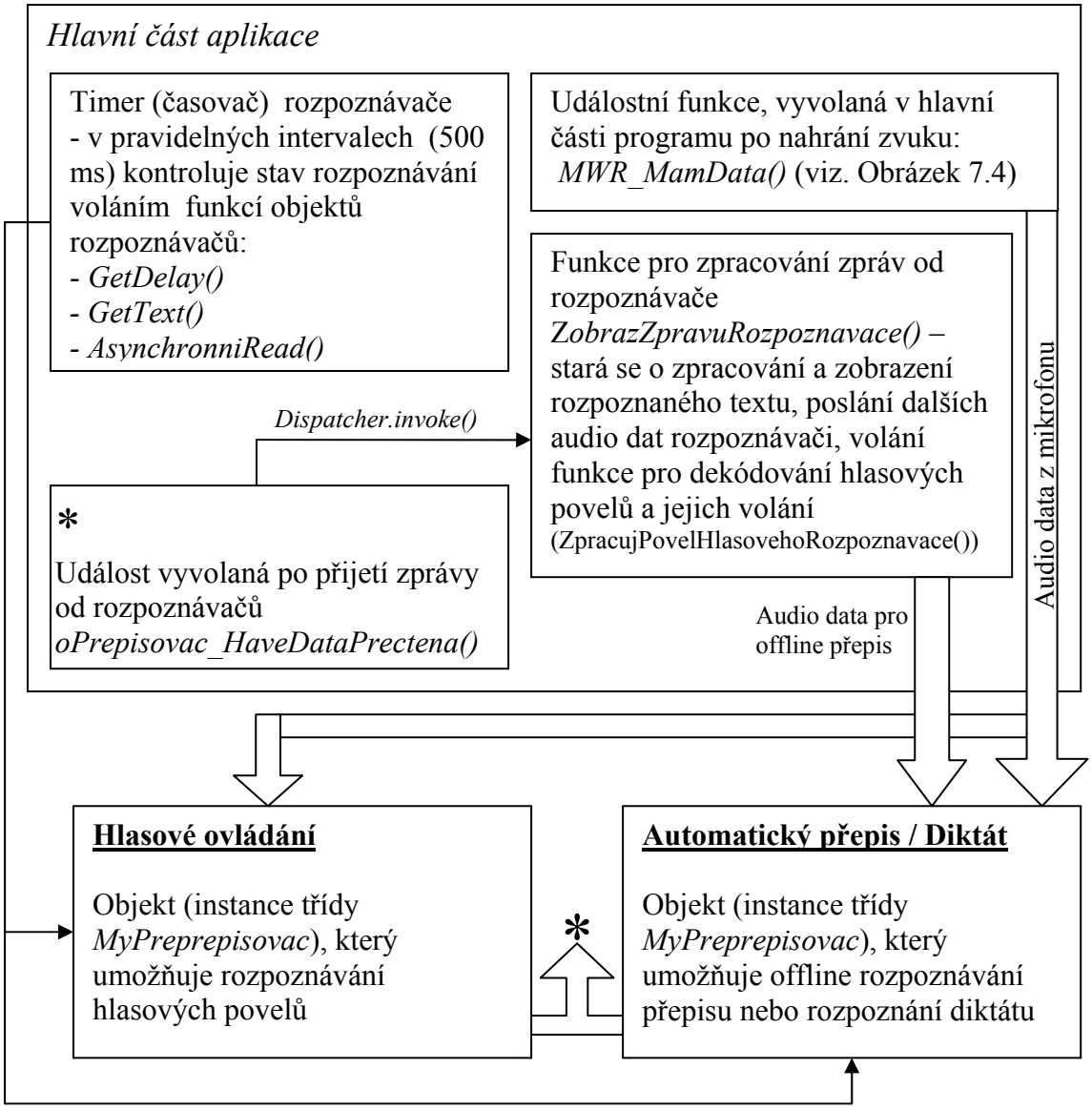
O spouštění a konfiguraci rozpoznávače se v programu stará třída *MyPrepisovac*. Obsahuje potřebné rozhraní pro komunikaci s rozpoznávačem pro zasílání a čtení komunikačních zpráv. Popis hlavních funkcí a důležitých property, které třída *MyPrepisovac* nabízí je uveden v příloze D.4.

9.1.2 Struktura napojení v2t – aplikace

Po vytvoření instance třídy *MyPrepisovac*, musí být zavolána její inicializace (viz. Příloha D.4). Také je spuštěn časovač (*TimerRozpoznavace*), který v pravidelných intervalech (500 ms) volá asynchronní čtení zprávy ze standardního výstupu rozpoznávače (*AsynchronniRead()*) a také zjišťuje aktuální zpoždění rozpoznávání (*GetDelay()*) a rozpoznaný text (*GetText()*). Pokud se na standardním výstupu rozpoznávače objeví data, jsou poslána pomocí delegované událostní metody (*oPrepisovac_HaveDataPrectena()*) do hlavní části aplikace.

Protože však není možno přímo v této funkci, která byla vyvolána jiným vláknem přistupovat k vizuálním částem hlavního formuláře (která běží v jiném vlákně), je využita možnost objektu *Dispatcher*. *Dispatcher* je ve WPF (viz. kapitola 3.2) objekt, který umožňuje komunikaci mezi vlákny. Pomocí metody *Invoke()* tohoto objektu je zavolána funkce *ZobrazZpravuRozpoznavace()*, která již má přístup k vizuálním komponentám hlavního formuláře aplikace a dojde ke zpracování rozpoznaných dat.

Na základě dekódování došlé zprávy je zobrazen rozpoznaný text do příslušné oblasti formuláře, nebo zavolán příslušný příkaz hlasového ovládání, nebo jsou poslána další audio data rozpoznávači (viz. popis funkcí v příloze D.4).



Obrázek 9.1: Struktura toku dat při automatickém rozpoznávání řeči

9.2 Implementace technologie v2t v aplikaci

V aplikaci je technologie *v2t* implementována pro tři režimy, které jsou popsány níže. Způsob použití je popsán v kapitole 4 (Uživatelské rozhraní aplikace). Důležité parametry technologie *v2t* (viz. kapitola 9.1) lze ovlivnit v nastavení aplikace (viz. kapitola 4.2.4).

Technologie v2t je do programu implementována tak, aby fungovala nezávisle na hlavní části aplikace (běží v samostatném procesu a vláknech).

9.2.1 Automatický přepis

Jedná se o nejdůležitější funkci automatického rozpoznávání řeči. Umožňuje automaticky přepisovat řečový dokument nebo jeho části podle předem nastavených parametrů rozpoznávání.

Při automatickém přepisování běží rozpoznávání v nejvyšší možné kvalitě. Manuálně je nutno pouze specifikovat, které části řečového dokumentu mají být přepsány (kapitoly, sekce a odstavce). U všech částí postačí zadat časové indexy počátku a konce v původním řečovém dokumentu. Lze také vybrat jednotlivé mluvčí, kteří mluví v dané části dokumentu. Rozpoznávač je pak vždy znovu inicializován pro každého mluvčího podle zadaných parametrů (jazykový model, typ mluvčího, přepisovací pravidla; viz.kapitola 6). Pokud není specifikován žádný mluvčí pro rozpoznávaný úsek, je rozpoznávač inicializován s výchozími parametry, které je možno změnit v nastavení aplikace (viz.kapitola 4.2.4)

9.2.2 Diktát

Jako doplněk k automatickému přepisu řečového dokumentu byla do aplikace přidána podpora diktování. Například pokud není možno kvůli rušivým zvukům v řečovém dokumentu (hluk, hudba, apod.) automaticky rozpoznat některou jeho část, lze příslušný úsek nadiktovat. Pomocí mikrofonu je tak možné diktovat a aplikace automaticky převádí řeč do textové podoby. Pro diktování je rozpoznávač spuštěn s parametry (kvalita rozpoznávání, jazykový model, akustický model, přepisovací pravidla), které je možno měnit v příslušné části nastavení aplikace (viz. kapitola 4.2.4).

Rozpoznávání v reálném čase je velice náročný proces a využije většinu výkonu počítače (viz. kapitola 10.3), proto je možno spustit pouze samotný *diktát* nebo *automatický přepis* (nikoliv současně).

9.2.3 Hlasové ovládání

Pro zrychlení práce s aplikací při ručním vytváření textového přepisu byla přidána podpora hlasového ovládání aplikace. Pomocí mikrofonu je tak možno ovládat některé funkce programu. Seznam funkcí, které je možno hlasově ovládat je uveden v příloze B. Hlasové povely je možno editovat v příslušném textovém souboru ve formátu XML. Příklad souboru je uveden na obrázku 9.2. Každý příkaz má své unikátní číslo *(indexMakra)*, podle kterého je identifikován. Dále obsahuje fonetický přepis hlasového povelu *(fonetickyPrepis)*, na který má rozpoznávač reagovat. Každý příkaz lze doplnit o text, který je zobrazen uživateli po jeho rozpoznání *(hodnotaVraceni)*. Aplikace reaguje

na unikátní čísla (*indexMakra*), která jsou uvedena v příloze B. Při editaci je možno měnit fonetický tvar příkazu a vytvářet tak nový hlasový povel pro konkrétní funkci. Lze také pro jednu funkci definovat více hlasových povelů (se stejným *indexMakra*).

```
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfMyMakro xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <MyMakro>
    <fonetickyPrepis>novípřepis</fonetickyPrepis>
    <hodnotaVraceni>Nový přepis</hodnotaVraceni>
    <indexMakra>10</indexMakra>
  </MyMakro>
  <MyMakro>
    <fonetickyPrepis>otevřítpřepis</fonetickyPrepis>
    <hodnotaVraceni>Otevřít přepis</hodnotaVraceni>
    <indexMakra>20</indexMakra>
  </MyMakro>
</ArrayOfMyMakro>
```

Obrázek 9.2: Příklad souboru s hlasovými makry pro ovládání programu

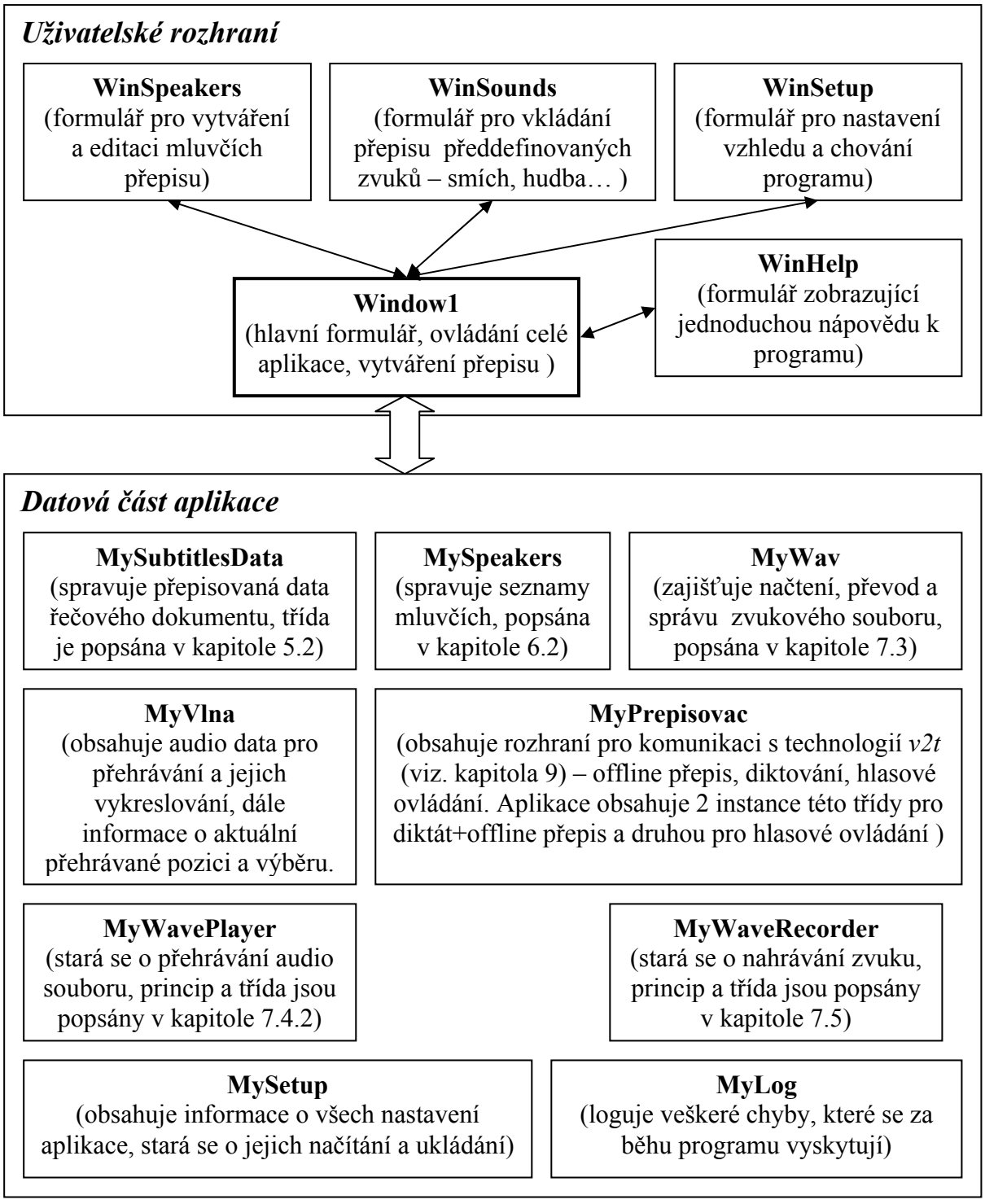
Pro *hlasové ovládání* je rozpoznávač spuštěn s parametry (kvalita rozpoznávání, jazykový model, akustický model, přepisovací pravidla), které je možno změnit v nastavení aplikace (viz.kapitola 4.2.4). Jazykový model (slovník rozpoznávače) by měl být zvolen prázdný, případně velmi malý, protože při hlasovém ovládání je využito pouze hlasových maker, která jsou rozpoznávána. Při použití malého slovníku je pak *hlasové ovládání* málo náročné na výkon počítače (je zapotřebí přibližně 15% procesorového výkonu) a je možné mít spuštěn *automatický přepis* i *hlasové ovládání* současně. Naopak není možno mít spuštěno současně *diktování* a *hlasové ovládání*.

Protože samotný rozpoznávač není přímo určen k hlasovému ovládání v reálném čase, bylo nutné vyvinout způsob jak zpracovat rozpoznaný text hlasových maker. Po vyslovení hlasového povelu a jeho rozpoznání, je nejprve vrácen pouze dočasný text. Až po chvíli je příkaz potvrzen a vrácen jako finální text. Čas mezi vrácení dočasného a potvrzeného textu však může být i několik vteřin. Takovéto zpoždění není vhodné pro ovládání v reálném čase. Proto je v aplikaci zpracováván i dočasný text a ihned po jeho přijetí je zavolána příslušná funkce aplikace. Příkaz je pak uložen do vnitřní fronty a jakmile přijde potvrzen ve finálním textu, je z fronty odstraněn. Toto řešení přináší okamžitou reakci na hlasový povel. Aby však nedocházelo k chybám při rozpoznávání příkazů (záměna povelů), je vhodné aby si nebyly jednotlivé hlasové příkazy podobné.

10 Aplikace Přepisovač 2.0

10.1 Struktura aplikace

Aplikace se skládá z několika objektů, které jsou instancemi tříd, jejichž provázanost je zobrazena na obrázku 10.1.



Obrázek 10.1: Struktura tříd a formulářů aplikace

Základním prvkem celé aplikace je třída *Window1*. Tato třída v sobě obsahuje všechny potřebné funkce, které zajišťují obsluhu událostí hlavního formuláře programu. Funkce třídy *Window1* lze rozdělit do kategorií uvedených níže. Jednotlivé kategorie jsou stručně popsány.

a) Práce s textovým přepisem

- Tato kategorie obsahuje funkce, které zajišťují zobrazování textového přepisu na hlavním formuláři (*ZobrazXMLData(), UpdateXMLData()*), jeho úpravu a synchronizaci s datovou strukturou při editaci textu (Přidávání a mazání úrovní textového přepisu) a ukládání a načítání přepisu do (ze) souborů (*NoveTitulky(), UlozitTitulky(), OtevritTitulky()*).

b) Aktualizace formuláře při přehrávání

- Funkce které jsou volány pravidelně při vyvolání události časovačem. Zajišťují správné zobrazení pozice kurzoru při přehrávání a aktuální zobrazení grafické podoby audio souboru na formuláři.

c) Zobrazení grafické reprezentace audio souboru

- Funkce, které zajišťují správné zobrazení vybrané části grafické reprezentace audio souboru na formuláři (*KresliVlnu(), KresliVyber(), KresliCasovouOsu()*).

d) Načtení a zobrazení audio a video souborů

- Funkce pro výběr a načtení multimediálních souborů pro další zpracování a zobrazení v aplikaci (viz. kapitoly 7 a 8).

e) Obsluha položek z hlavního menu programu a jejich klávesových zkratek

- Funkce, které zajišťují obsluhu položek hlavního menu programu. Jsou také volány po stisku klávesové zkratky (*Window_PreviewKeyDown()*)

f) Obsluha položek z kontextových menu

- Funkce obsluhující příkazy položek kontextových menu z oblasti textového přepisu, grafického zobrazení vlny a video souboru.

g) Obsluha automatického rozpoznávání řeči a hlasového ovládání

- Funkce, které zajišťují posílání zvukových dat rozpoznávači (technologie v2t) a zpracování rozpoznaných dat (viz. kapitola 9.1.1).

10.2 Podpůrné soubory

Pro svůj běh aplikace vyžaduje podpůrné soubory. Všechny tyto soubory se nachází v adresářích u hlavního spustitelného souboru. Některé soubory lze ručně editovat a měnit tak chování programu (viz. kapitola 4.2.3 a kapitola 9.2.3)

Tab. 10.1: Seznam adresářů aplikace, podpůrných souborů a jejich význam

Adresář (relativní cesta)	Seznam souborů
/Data/	*config.xml* – konfigurační soubor aplikace s uchováním nastavení *makra.xml* – soubor s řečovými makry pro hlasové ovládání aplikace (lze ručně editovat) *ruchy.xml* – soubor se seznamem neřečových zvuků. (lze ručně editovat) *DatabazeMluvcich.txt* – výchozí umístění interní databáze mluvčích (umístění lze měnit v nastavení aplikace)
/Nanocore/	**/amodels/** – adresář obsahující soubory akustických modelů (*.amd); **/drules/** – adresář obsahující soubory přepisovacích pravidel (*.ppp); **/lmodels/** – adresář obsahující soubory jazykových modelů (*.bin); *nanocore.dll* – hlavní spustitelný soubor rozpoznávače *locale.pm, perl58.dll, t-perl.dll* – soubory nutné pro běh rozpoznávače *licenční soubor* – soubor, který musí být vygenerován a aktivován pro konkrétní počítač (nutno kontaktovat laboratoř Speechlab [6]); soubor může být umístěn libovolně, ale je doporučeno umístění v daném adresáři
/Prevod/	*ffmpeg.exe* – soubor pro převod multimediálních souborů na požadovaný formát. (viz. kapitola 7.3.1) **/temp**/ – adresář pro dočasné audio soubory

10.3 Softwarové a hardwarové požadavky aplikace

Aplikace je vytvořena pro operační systém Microsoft Windows XP a vyšší. Pro svůj běh potřebuje rozhraní .NET Framework verze alespoň 3.0. Pro správné přehrávání multimediálních souborů je vyžadována instalace příslušných kodeků.

Protože je aplikace vyvinuta pro podsystém WPF, je grafická část uživatelského rozhraní zobrazována pomocí grafické karty. Při vývoji bylo zjištěno, že v některých

případech dochází k chybnému zobrazení částí aplikace (náhodné „rozpadávání obrazu“ na jednotlivých formulářích). Problémy se nejčastěji vyskytovaly na grafických kartách firmy nVidia. Řešení přinesla aktualizace ovladačů grafické karty na novější verzi. Proto je v případě problémů se zobrazením aplikace doporučeno aktualizovat ovladače grafické karty a nainstalovat verzi .NET Framework 3.0, která se při vývoji aplikace ukázala být stabilnější než verze 3.5.

Aplikace je pak schopna běžet na libovolném počítači vybaveným výše zmíněným software. Pro využití funkcí aplikace je dále vyžadována zvuková karta. Při použití technologie *v2t* je doporučeno spouštět aplikaci na dvoujádrovém procesoru s taktem alespoň 2 GHz a minimálně 1GB operační paměti. Dále je při použití technologie *v2t* vyžadován licenční soubor pro konkrétní počítač a připojení k internetu pro ověření této licence.

Tab. 10.2: Doporučená konfigurace počítače pro optimální běh aplikace

Procesor:	2 x 2 GHz (Intel, AMD)
Operační paměť:	2 GB
Zvuková karta:	ANO
Mikrofon:	ANO
Grafická karta:	s podporou DirectX 9.0c nebo novější
Připojení k internetu:	ANO
Licenční soubor pro *v2t:*	ANO [6]
Operační systém:	Microsoft Windows XP
Kodeky:	ANO (pro audiovizuální soubory, které budou přehrávány)

11 Závěr

Výsledkem Diplomové práce je aplikace, která umožňuje vytvářet textový přepis řečových dokumentů (audio a video souborů). Přepis je rozdělen do přehledných úrovní kapitola – sekce – odstavec. U každé úrovně lze nastavit časy jejich počátku a konce. Pro nejnižší úroveň lze dále nastavit mluvčího. Promluvy různých mluvčích se také mohou překrývat.

Aplikace umožňuje pracovat s většinou dnes běžně používaných audio souborů, které je možno přehrávat a zobrazovat graficky v podobě vlny. Program dále umožňuje přehrávat i video soubory. Video pomáhá s identifikací mluvčích, kteří se v přepisovaných materiálech vyskytují a usnadňuje tvorbu textového přepisu.

Textový přepis mluveného dokumentu je ukládán do souboru v textovém formátu XML. Tento formát je dostatečně přehledný a umožňuje případné využití přepisů i v jiných aplikacích.

Software dále podporuje kompletní správu mluvčích, kteří se v dokumentech mohou vyskytovat. Lze vytvářet seznamy mluvčích pro různé textové přepisy. Seznamy jsou uloženy v textovém formátu XML jako v případě textového přepisu. Tím je zajištěna případná budoucí použitelnost těchto seznamů i v jiný aplikacích, které je mohou využívat. O jednotlivých mluvčích umožňuje aplikace kromě jména uchovávat podrobnější informace: například obrázek, který zjednodušuje identifikaci mluvčích při přehrávání video souboru.

Do aplikace byla také přidána podpora technologie *v2t* (rozpoznávání spojité řeči, vyvinuté laboratoří Speechlab na Technické univerzitě v Liberci). Program umožňuje automatické přepisování řečových dokumentů a jejich částí. Také je zde přítomna možnost diktování kratších úseků textu. Poslední funkce, pro kterou je technologie *v2t* využita je hlasové ovládání aplikace. Hlasové ovládání je použito u vybraných funkcí a pomáhá zrychlit práci při ručním přepisu řečových dokumentů.

Při vytváření textových přepisů také dochází ke značnému zdržení při současném použití myši a klávesnice. Aplikace byla proto vytvořena tak, aby všechny často používané funkce byly přístupné pomocí klávesových zkratek (případně hlasových povelů).

Při vývoji aplikace bylo nutné navrhnout strukturu textového přepisu (viz. kapitola 5) pro potřeby ručního zpracování i automatického přepisu, zvolit vhodný způsob přehrávání multimediálních souborů (viz. kapitola 7) a vyřešit problémy

s rychlostí a kvalitou zobrazení grafické podoby audio dat (viz. kapitola 8). Hlavním problém při vývoji uživatelského rozhraní aplikace bylo náhodné „rozpadávání obrazu formulářů“, které bylo nakonec vyřešeno aktualizací ovladačů grafické karty a použitím stabilní verze .NET 3.0 (viz. kapitola 10.3)

Aplikace splňuje všechny požadavky zadání a je v ní možno kompletně zpracovávat mluvené dokumenty. Program je možno konfigurovat a přizpůsobovat tak jeho vzhled a chování požadavku uživatele (vzhled textového přepisu, technologie v2t, tvorba vlastních hlasových povelů, vlastní neřečové zvuky, seznamy mluvčích). V budoucnosti by bylo možné aplikaci dále rozšiřovat vylepšováním stávajících funkcí (např. editace textového přepisu, ovládání programu) i doplněním nových funkcí (podpora exportu a importu textového přepisu do jiných formátů, podpora dalších řečových technologií, apod.).

Použitá literatura

[1] TROELSEN, Andrew. *C# a .NET 2.0 profesionálně.* Zoner, 2006. ISBN: 80-86815-42-0

[2] *C sharp*. [online]. URL: (www.csharp-home.com)

[3] SELLS, Chris, GRIFFITHS, Ian. *Programming WPF, Second Edition*. O'REILLY Media, 2007. ISBN 10: 0-596-51037-3

[4] *WPF*. [online]. URL: (www.vyvojar.cz/Articles/445-0-wpf-uvod.aspx)

[5] *Wave File Format*. [online]. URL: (www.sonicspot.com/guide/wavefiles.html)

[6] *Laboratoř Speechlab*. [online]. URL: (www.ite.tul.cz/speechlab/index.php)

Příloha A – Klávesové zkratky aplikace

Tab. A.1: Seznam klávesových zkratek hlavního formuláře

Klávesová zkratka	**Funkce**
Tab	přehrání/zastavení audio (video) souboru
Ctrl+Tab	přehrání vybrané části souboru s opakováním
Alt+Left	posun audio souboru o 1s zpět
Alt+Right	posun audio souboru o 1s vpřed
Ctrl+N	nový přepis
Ctrl+O	otevře soubor s přepisem
Ctrl+S	uloží soubor s přepisem
Ctrl+M	otevře nové okno, kde lze vytvořit nového mluvčího, který se v přepisu vyskytuje a nastavit ho aktuálnímu elementu
Ctrl+R	otevře nové okno, ve kterém lze vybrat některý ze zvuků, které se v přepisovaném audio souboru mohou vyskytovat
F2	nová kapitola přepisu
F3	nová sekce přepisu
Shift+F3	nová sekce přepisu na aktuální pozici v textovém přepisu
Shift+Del	smaže aktuální kapitolu, sekci nebo odstavec
Ctrl+Del	smaže počáteční časový index aktuálního elementu
Ctrl+Home	nastaví počáteční časový index daného elementu podle pozice kurzoru
Ctrl+End	nastaví koncový časový index daného elementu podle pozice kurzoru
Alt+Enter	maximalizuje okno hlavního formuláře na celou obrazovku
F5	spustí automatické rozpoznávání vybraného elementu textového přepisu (musí být načten audio soubor)
F6	spustí režim diktování do aktuálně vybraného elementu textového přepisu
F7	spustí hlasové ovládání funkcí programu
F12	vytvoří obrázek mluvčího z přehrávaného video souboru
F1	okno se seznamem klávesových zkratek
Ctrl+F1	informace o programu

Příloha B – Dostupné příkazy hlasového ovládání

Tab. B.1: Seznam dostupných funkcí pro hlasové ovládání

Hlasový povel	**Index makra**	**Popis funkce**
Nový přepis	10	Nový textový přepis
Otevřít přepis	20	Dialog pro otevření textového přepisu
Otevřít audio	30	Dialog pro otevření audio souboru
Otevřít video	40	Dialog pro otevření video souboru
Uložit	50	Uloží aktuální textový přepis
Uložit jako	60	Uloží aktuální textový přepis jako…
Mluvčí	220	Nastaví mluvčího aktuálního elementu
Vložit ruch	230	Zobrazí správce neřečových zvuků
Nastavení	200	Otevře nastavení aplikace
Nová kapitola	301	Vytvoří novou kapitolu textového přepisu
Nová sekce	302	Vytvoří novou sekci textového přepisu
Nový odstavec	303	Vytvoří nový odstavec textového přepisu
Nápověda	400	Zobrazí okno s nápovědou
O programu	401	Zobrazí okno s informacemi o programu
Ukončit	500	Ukončí aplikaci
Maximalizovat	505	Maximalizuje/obnoví formulář aplikace
Minimalizovat	506	Minimalizuje/obnoví formulář aplikace
Konec hlasového ovládání	550	Ukončí hlasové ovládání
Přehrát	1000	Přehraje audio/video soubor
Zastavit	1001	Zastaví přehrávání audio/video souboru

Příloha C – Hlavní menu aplikace

Tab. C.1: Popis hlavního menu aplikace

Položka/nabídka	**Kláv. zkratka**	**Popis**
Soubor		
Nový přepis	Ctrl+N	Nový textový přepis
Otevřít přepis	Ctrl+O	Dialog pro otevření textového přepisu
Uložit	Ctrl+S	Uloží aktuální textový přepis
Uložit jako		Uloží aktuální textový přepis jako…
Otevřít audio		Dialog pro otevření audio souboru
Otevřít video		Dialog pro otevření video souboru
Konec		Konec programu
Nástroje		
Nastav mluvčího	Ctrl+M	Nastaví mluvčího aktuálního elementu
Vložit ruch	Ctrl+R	Zobrazí správce neřečových zvuků
Vytvořit obrázek mluvčího	F12	vytvoří obrázek mluvčího z přehrávaného video souboru
Nastavení		Otevře nastavení aplikace
Úpravy		
Nová kapitola	F2	Nová kapitola přepisu
Nová sekce	F3	Nová sekce přepisu
Nová sekce na pozici	Shift+F3	nová sekce přepisu na aktuální pozici v textovém přepisu
Smazat položku	Shift+Del	smaže aktuální kapitolu, sekci nebo odstavec
Smazat poč. časový index	Ctrl+Del	smaže počáteční časový index aktuálního elementu
Smazat kon. časový index		smaže koncový časový index aktuálního elementu
Přepisovač		
Automatický přepis	F5	Spustí/zastaví automatický přepis vybraného elementu (musí být načten audio soubor)
Diktát	F6	Spustí/zastaví možnost diktování do vybraného elementu
Hlasové ovládání	F7	Spustí/zastaví hlasové ovládání aplikace
Nápověda		
Popis programu	F1	Zobrazí okno s klávesovými zkratkami
O programu	Ctrl+F1	Informace o verzi programu

Příloha D – Popis vybraných funkcí a proměnných

D.1 Třída MySubtitlesData

Poznámka: Některé dále uvedené funkce jsou v programu přetíženy a obsahují i další parametry. Pokud je ve funkcích použita proměnná udávající časové hodnoty, tak jednotky jsou vždy milisekundy. Proměnná typu MyTag obsahuje index kapitoly, sekce a odstavce – určuje tak každý element a je použita ve většině funkcí programu.

`public int NovaKapitola()` – Do datové struktury přidá novou kapitolu, vrací index udávající pozici kapitoly.

`public int NovaSekce(int kapitola, int index)` – Na danou pozici v kapitole (specifikované indexem) přidává novou sekci a vrací index nové sekce.

`public int NovyOdstavec(int kapitola, int sekce, int index)` – Na daný index v zadané kapitole a sekci přidá nový odstavec. Vrací index odstavce.

`public bool UpravElement(int kapitola,int sekce, int odstavec, string text)` – Edituje text odstavce, název kapitoly nebo sekce podle zadaných indexů.

`public bool UpravCasElementu(MyTag aTag, long timeStart, long timeStop)` – Tato funkce upravuje časové indexy začátku a konce kapitol, sekcí a odstavců.

`public long VratCasElementuPocatek(MyTag aTag)` – Vrací počáteční čas kapitoly, sekce nebo odstavce podle zadané proměnné.

`public long VratCasElementuKonec(MyTag aTag)` – Vrací koncový čas kapitoly, sekce nebo odstavce podle zadané proměnné.

`public int NovySpeaker(MySpeaker aSpeaker)` – Přidává do seznamu mluvčích nového mluvčího a vrací jeho index v seznamu.

`public bool OdstranSpeakera(MySpeaker aSpeaker)` – Odstraní zvoleného mluvčího (pokud existuje) ze seznamu mluvčích. Pokud byl mluvčí jinde v přepisu, odstraní ho z jednotlivých odstavců.

`public bool ZadejSpeakera( MyTag aTag, MySpeaker aSpeaker)` – Přiřadí mluvčího odstavci, určeným v proměnné.

`public MySpeaker VratSpeakera(MyTag aTag)` – Vrací mluvčího ze zadaného odstavce.

`public MySpeaker NajdiSpeakera(string aJmeno)` – Najde mluvčího v celkovém seznamu mluvčích podle jeho jména.

`public bool Serializovat(string jmenoSouboru, MySubtitlesData co, bool aUkladatKompletMluvci)` – Funkce, která převede instanci třídy *MySubtitlesData* a uloží ji do souboru, jehož název je zadán v parametrech. V případě nezdaru vypíše chybovou hlášku a vrací hodnotu *false*. Jinak vrací *true*.

`public MySubtitlesData Deserializovat(String jmenoSouboru)` – Funkce, která převede zadaný soubor na objekt třídy *MySubtitlesData*. V případě neúspěchu vrací hodnotu *null*. Jinak vrací převedený objekt.

D.2 Třída MySpeakers

`public int NovySpeaker(MySpeaker aSpeaker)` – Přidá nového mluvčího do seznamu a vrací jeho příslušný index. V případě neúspěchu (například pokud již mluvčí se stejným názvem existuje) vrací hodnotu -1.

`public MySpeaker VratSpeakera(int aIDSpeakera)` – Podle ID vrátí příslušného mluvčího. Pokud není hledaný mluvčí v seznamu, vrací prázdného mluvčího (nová instance bez jména mluvčího)

`public bool OdstranSpeakera(MySpeaker aSpeaker)` – Odstraní zadaného mluvčího z vnitřního seznamu. V případě neúspěchu vrací *false* jinak *true.*

`public int NajdiSpeakeraID(string aJmeno)` – Podle zadaného jména mluvčího se pokusí vyhledat v seznamu mluvčích jeho ID. Pokud mluvčí neexistuje vrací *-1*.

`public bool UpdatujSpeakera(string aJmeno, MySpeaker aSpeaker)` – Změní informace o příslušném mluvčím, který je hledán podle jména. V případě nenalezení záznamu je vrácena hodnota *False.*

`public bool Serializovat(String jmenoSouboru, MySpeakers co)` – Ukládá seznam mluvčích do XML souboru.

`public MySpeakers Deserializovat(String jmenoSouboru)` – Pokusí se načíst seznam mluvčích z XML souboru. V případě neúspěchu vrací prázdný seznam.

D.3 Třída MyWav

`public MyWav()` – constructor, stará se o inicializaci proměnných a počáteční nastavení

`public void AsynchronniNacteniRamce2(…)` – Spustí Thread, ve kterém se pokusí načíst specifikovaná audio data z dočasných souborů, pokud existují. Zároveň lze mít spuštěné až 2 thready s načítáním různých dat pro zobrazení vlny a pro požadavky technologie *v2t* (viz. kapitola 9) Data jsou předána delegovanou událostní metodou registrovanou vně třídy

`public void AsynchronniPrevodMultimedialnihoSouboruNaDocasne2(…))` - Spustí *Thread*, ve kterém je spuštěn *Process* s *ffmpeg* a převádí multimediální soubor na dočasné *.wav soubory, které ukládá. Pomocí delegované metody předává hlavnímu programu informaci o stavu převodu a také první převedená audio data pro vykreslení vlny.

`public void Dispose()` – Zajistí zrušení všech Threadů načítání a procesu převodu.

`public bool Nacteno` – uchovávající informaci zda je načten kompatibilní multimediální soubor, který je možno převádět

`public bool Prevedeno` – Informace,zda je multimediální soubor převeden

`public long DelkaSouboruMS` – Informace o délce multimediálního souboru v milisekundách

`public bool NacitaniBufferu` – Informace, zda dochází k načítání audio dat z dočasných souborů

Další property *pFrekvence*, *pPocetVzorku, pVelikostVzorku, pPocetKanalu*, popisují formát WAV, ve kterém jsou dočasné soubory ukládány.

D.4 Třída MyPrepisovac

`public MyPrepisovac(…)` - constructor třídy, jeho parametry jsou: cesta k hlavnímu souboru rozpoznávače; soubor akustického modelu; soubor jazykového modelu; soubor přepisovacích pravidel; licenční server; licenční soubor; velikost interního bufferu; kvalita přepisu pro online rozpoznávání; Odkaz na delegovanou událostní funkci, která je vyvolána v případě asynchronního přečtení dat z rozpoznávače.

`public int InicializaceRozpoznavace(…)` - funkce, která inicializuje rozpoznávač – spustí samostatný proces hlavního souboru rozpoznávače (*nanocore.dll*) s příslušnými parametry. Může obsahovat parametry shodné s constructorem, pokud již běží proces, dojde k jeho ukončení a spuštění s novými parametry (využito při rozpoznávání pro různé akustické modely – různé mluvčí).

`public int Start(short aStavRozpoznavace)` – Pošle rozpoznávači zprávu s požadavkem na spuštění rozpoznávání. Parametr funkce udává v jakém módu je rozpoznávač spuštěn (offline rozpoznávání, diktování, hlasové ovládání – viz. kapitola 9.2).

`public void AsynchronniStop()` – Pošle rozpoznávači zprávu o ukončení rozpoznávání, rozpoznávání je ukončeno až po dokončení přepisu všech předaných audio dat.

`public int StopHned()` – Pošle rozpoznávači zprávu s požadavkem o okamžité ukončení rozpoznávání bez čekání na dokončení přepisu dat ve vyrovnávací paměti.

`public bool AsynchronniZapsaniDat(byte[] aData)` – Zapíše audio data pro rozpoznání na standardní vstup rozpoznávače. Zápis je řešen asynchronně pomocí threadu (vlákna), aby nedocházelo k brždění hlavního programu při zápisu. Pokud již dochází k zápisu předchozí zprávy, vrací funkce hodnotu *false.*

`public int GetDelay()` – Pošle zprávu s požadavkem na zjištění zpoždění rozpoznávání (kolik audio dat ještě zbývá k rozpoznání). Tato funkce je cyklicky volána z hlavního vlákna programu (viz. obrázek 9.1).

`public int GetText()` – Pošle zprávu s požadavkem na zjištění rozpoznaného textu. Rozpoznávač pak vrací rozpoznaný text, pokud je k dispozici. Tato funkce je cyklicky volána z hlavního vlákna programu (viz. obrázek 9.1).

`public void AsynchronniRead()` – Spustí nový thread (vlákno), které čeká až se na standardním výstupu rozpoznávače objeví data (např. zpoždění, rozpoznaný text, informace o inicializaci a ukončení rozpoznávání). Funkce musí být cyklicky volána z hlavní části aplikace (nový thread je spuštěn pouze v případě, že již není spuštěno čtení v předchozím threadu).

`public bool ZapisMakra(List<MyMakro> aSeznamMaker)` – Zapíše seznam hlasových maker, která jsou použita pro hlasové ovládání (viz. kapitola 9.2.3 a Příloha B).

`public void Dispose()`– Zajistí zrušení všech běžících threadů, hlavního procesu rozpoznávače.

Hlavní část aplikace (třída *Window1*) obsahuje několik funkcí, které jsou používány pro volání výše popsaných funkcí. Dodávají rozpoznávači nová data k přepisování a pracovávají již přepsaná textová data z rozpoznávače.

`void oPrepisovac_HaveDataPrectena(object sender, EventArgs e)` – Jedná se o funkci, která je vyvolána přepisovačem, pokud jsou k dispozici data pro zpracování (rozpoznaný text, zpoždění rozpoznávání, zprávy o stavu rozpoznávače, apod.). Tato funkce je společná pro zpracování dat z offline rozpoznávání, diktovávání a hlasového ovládání.

`public bool ZpracujRozpoznanyText(long aAbsolutniCasPocatku, int aPredchoziDelkaTextu, ref string aText, ref List<MyCasovaZnacka> aCasoveZnacky)` – Zpracuje rozpoznaný text tak, že vrátí samotný text a k němu synchronizační časové značky počátků jednotlivých úseků textu. Text a časové značky je pak možno uložit do datové struktury textového přepisu.

`private void ZobrazZpravuRozpoznavace(MyEventArgsPrectenaData e)` – Nejdůležitější funkce pro externí zpracování rozpoznaných dat a ovládání rozpoznávače (pro všechny tři módy: Offline rozpoznávání, Diktování, Hlasové ovládání). Podle typu zprávy z rozpoznávače umožňuje zobrazit a uložit finální textová data do datové struktury textového přepisu, volá příslušné funkce pro nahrání dalších zvukových dat při offline rozpoznávání a v případě hlasového ovládání volá funkci na jeho zpracování (popsanou níže).

`private bool ZpracujPovelHlasovehoRozpoznavace(string s)` – Podle textového makra, které je rozpoznáno při hlasovém ovládání zavolá příslušný příkaz programu pro provedení požadované akce.