

Univerzita Palackého v Olomouci

Přírodovědecká fakulta

Katedra informatiky

BAKALÁŘSKÁ PRÁCE

Lukáš Otáhal

Kalendář pro nevidomé



Olomouc 2013

vedoucí práce: RNDr. Arnošt Večerka

Anotace

Tato práce popisuje, jakým způsobem nevidomý pracuje s počítačem či s jakými problémy se může setkat. Následně pojednává o implementaci kalendáře, který je pro nevidomého uživatele takřka nezbytným doplňkem počítače. Kalendář má za úkol zpřístupnit základní funkce, a to co nejrychleji a pokud možno uživatelsky přívětivě. Program je možno ovládat pomocí globálních a standardních klávesových zkratk.

This bachelor thesis describes the way the blind user works with the computer and problems that are related to this topic. Furthermore, this thesis provides information about implementation of a calendar application what calendar is necessary computer accessory for each and every blind user. Calendar makes all its functions easily available for the blind user, while keeping certain level of comfort. Calendar can be managed via global keyboard shortcuts.

Prohlášení

Prohlašuji, že tato práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje včetně webových, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Poděkování

Děkuji vedoucím této práce *Mgr. Martinu Dostálovi, Ph.D.* a *RNDr. Arnoštu Večerkovi* (převzal vedení práce po odchodu dr. Dostála), kteří mi umožnili vypracovat pro mě zajímavé téma.

Dále děkuji svým asistentům *Lukáši Provazníkovi, Lukáši Laštovičkovi, Petru Zlámalovi* a *Františku Mrtvému*, kteří mi pomáhali v průběhu celého studia s výukou programování a vysvětlením skriptů, pomohli mi s přístupností jak programovacího jazyka, tak webových technologií. Velký dík patří asistentovi *Martinu Rotterovi*, který mi propůjčil svůj zrak k úpravě přehlednosti této práce; vážím si jeho pomoci při programování v situacích, kdy mi odečítač obrazovky nečetl dané údaje potřebné pro psaní aplikace.

Můj dík patří rovněž panu Rostislavu Sáčkovi, který mi poskytl konzultaci právě k problematice hlasových výstupů, kterými se v této práci zabývám.

Klíčová slova

Kalendář, odečítač obrazovky, nevidomý uživatel, hlasový výstup, C#, NVDA, JAWS, WinMonitor, Wintalker, hlasový výstup Hlas, hlas eSpeak.

Obsah

Úvod	6
1. Nevidomý a práce s počítačem	8
1.1 Odečítače obrazovky	10
1.1.1 Svobodné odečítače obrazovky	11
1.1.2 Komerční odečítače obrazovky	12
1.2. Přístupnost Mac OS X.....	14
1.3 Přístupnost systému Linux	15
1.4 Braillovský řádek	16
2. Programové vybavení nevidomého uživatele	18
3. Testování kalendáře uživateli, metoda ovládnání a cíle	19
3.1 Návrh ovladatelnosti aplikace	19
3.2 Ověření aplikace uživateli a testování.....	19
3.3 Problémy s přístupností při psaní aplikace.....	21
3.4 Microsoft Speech API	22
3.5 Odečítače obrazovky a technologie hlasů	24
3.6 Technologie hlasů	25
3.7 Rozdíly mezi SAPI.....	26
3.8 Skriptování kalendáře.....	27
3.8.1 Skripty odečítače JAWS.....	27
3.8.2 Skripty odečítače WinMonitor a NVDA	36

4. Přístupnost mobilních operačních systémů.....	38
4.1. Symbian.....	38
4.2. IOS	39
5. Kalendář.....	41
5.1. Použité technologie	43
5.2. Základní programové moduly	43
5.3. Vztah mezi programovými moduly.....	44
5.3.1 Implementace klávesových zkratk.....	44
5.3.2. Implementace načítání svátků.....	45
5.3.3. Implementace vyskakovacího okna	47
5.3.4. Implementace adresáře	48
5.3.5. Nastavení	53
5.4. Ovládání a rozhraní kalendáře	55
5.4.1. Seznam klávesových zkratk a funkcí.....	56
5.5. Uživatelská příručka.....	57
5.5.1. Hlavní menu.....	57
5.5.2 Nastavení	59
Slovník pojmů	61
Literatura	64

Úvod

Informační a komunikační technologie jsou všude kolem nás. Ve školách, na pracovištích, v obchodech. Počítač vlastní již téměř každý a rozvoj SW a HW je na takové úrovni, že i nevidomý uživatel může pracovat s počítačem zcela sám a plnohodnotně. Při výstupu nepoznáte, že dokument zpracoval nevidomý uživatel počítače. S tímto rozvojem výpočetní techniky se rozvíjí i kompenzační pomůcky pro nevidomé. Ty jsou jak na softwarové úrovni, tak na hardwarové úrovni upraveny tak, aby je bylo možné ovládat zcela bez zraku. Díky těmto technologiím a softwarovým úpravám může nevidomý pracovat jak v zaměstnání a během studia, tak ve volném čase. Jsou zde však jistá omezení, a proto je třeba při práci s těmito nástroji uživateli bez zraku občas pomoci.

Počítač s hlasovým výstupem či daný program, který má zpřístupnit dané vybrané informace, se nazývá kompenzační pomůcka.

Informace uživatel dostává s pomocí hlasového syntetizátoru a hmatově s pomocí Braillova řádku. Velmi mě zajímá přístupnost různých aplikací, a proto jsem se rozhodl naprogramovat tuto aplikaci, aby byla pro nevidomé dostupná v celém operačním systému. Aplikace je skrytá v liště a uživatel ji může ovládat, ať se nachází kdekoli, pomocí klávesových zkratk. Aplikace pro nevidomé musí být upravena tak, aby odečítače obrazovky byly schopné danou informaci vyčíst. Neměla by obsahovat grafické či jinak barevně odlišené prvky. Popisky by měly být výhradně textové, neboť uživatel pracuje pouze s tím objektem, na kterém je kurzor, zbytek obrazovky je pro něj v tu chvíli nepoužitelný. Při vstupu do aplikace či webové stránky si ji musí uživatel projít celou, aby věděl, jaká menu, dialogy či prvky dané rozhraní obsahuje. Počítač je ovládán pouze z klávesnice. Pro představu: při návštěvě internetové stránky ji musí daná osoba projít celou, aby se seznámila s odkazy, texty, nadpisy a jinými prvky na daném webu. Tak to platí i u ostatních aplikací. Je třeba si projít celé rozhraní včetně menu, editačních polí, kde můžeme vkládat text, seznamů a zaškrtačích tlačítek. Pro programování jsem zvolil jazyk C#, který se mi jeví jako nejvíce dostupný a ovladatelný, včetně pochopení jeho syntaxe a sémantiky. Pro psaní jsem dále použil

Microsoft Visual Studio 2010 s odečítačem NVDA, který částečně zpřístupnil menu a funkce programu.

Vzhledem k tomu, že mnoho aplikací nejsou uživatelsky přístupné nevidomým, což jsem si potvrdil při komunikaci s nimi, rozhodl jsem se naprogramovat kalendář pro nevidomé a v něm funkce, které jsou denně používány, a uplatnit tak samostatnost aplikace, aby běžela nezávisle bez ohledu na to, jaký hlas či odečítač je právě používán.

Zvolil jsem ovládání pomocí klávesových zkratk, neboť tato alternativa se mi jeví jako uživatelsky přístupná a rychlá. Díky klávesové zkratce se daná funkce vyvolá odkudkoli, a to je dle mého názoru největší výhoda programu. Odpadá nutnost vstupovat do nabídky Start, do Programů, vybrat daný Program a spustit jej, následně vyhledat danou funkci a klávesou Enter ji aktivovat. Některé programy se tak někdy stávají nepřehledné právě díky menu a spouštění prvků z jeho podjednotek.

1. Nevidomý a práce s počítačem

Nejrozšířenější platformou pro osobní počítače pro nevidomé je platforma Windows. Jelikož někteří nevidomí uživatelé pracují i v Linuxu, je zde problém s kompatibilitou ovladačů. Jedná se výhradně o podporu Braillových řádků a Braillových tiskáren. Čím dál víc se začíná prosazovat práce s počítači typu Mac. Uživatel je odkázán pouze na to, co mu přečte hlasový výstup, či co mu zobrazí Braillový zobrazovač. Počítač je ovladatelný jen tehdy, naběhne-li operační systém, ovladač zvukové karty a základní rozhraní systému. Velkým problémem je nemožnost čtení informace po okamžitém spuštění počítače, prakticky nemožná je práce v nouzovém režimu. Pokud z jakéhokoliv důvodu není spuštěn správně operační systém a po startu odečítač obrazovky, počítač se stává nepoužitelným.

Ovládání počítače je, jak již bylo zmíněno, výhradně z klávesnice. Je třeba, aby se uživatel seznámil s rozhraním aplikace, prošel si všechna menu, rozbalovací seznamy, submenu, veškerá nastavení, aby vůbec věděl, co daný program umí. Vidící osoba uvidí vlevo nahoře menu a v něm položky, avšak bez zraku je třeba si všechna menu projít přes klávesnici. Pro pohyb v aplikacích se výhradně používá tabulátor, kurzorové šipky, funkční klávesy, klávesové zkratky, které jsou velmi důležité a v práci s aplikacemi velmi pomáhají. Není třeba jít do menu, šipkou si dohledat danou položku a pak ji teprve zvolit. Klávesová zkratka provede úkon okamžitě bez hledání dané funkce.

Uživatelům dále pomáhají různé komerční balíky, které sdružují různé aplikace právě uzpůsobené pro začínajícího uživatele počítače, např. Winmenu od firmy Galop, Asistent od firmy ACE Design, Guide od společnosti Dolphin. Tyto balíky v sobě sdružují e-mailové klienty, správce souborů, textové editory, aplikace pro pohodlnější práci se skenerem, telefonní seznamy apod. Pro spuštění těchto zjednodušených nástrojů je však třeba použít rozhraní celého balíku aplikací, ve kterém byly dodány. Jako mluvicí počítač může být použito jakékoliv PC, které je opatřené zvukovou kartou a operačním systémem. Mezi nevidomými uživateli je velmi oblíbený notebook. Pokud se jedná o malý notebook či netbook, je možné toto zařízení použít i na cestách, jako zápisník, avšak se všemi počítačovými funkcemi. Pokud je v notebooku integrován 3G modem, je práce neocenitelná a lze data zpracovávat opravdu kdekoli. Právě možnost

získání informací kdekoli je někdy pro nevidomého nejdůležitější (např. možnost vyhledání v jízdním řádu apod.). Počítač dále musí obsahovat screen reader (odečítač obrazovky) a hlasový syntetizér, tedy hlas, který čte informace z obrazovky počítače. Jestliže k počítači připojíme skener a použijeme příslušný SW, stává se z počítače též čtecí zařízení.

Dále je třeba daný počítač upravit, a to volbou rozlišení právě kvůli odečítacím programům, je nutné vypnout veškeré postranní panely, různé barevné efekty, různá rozhraní systémů Aero. Díky těmto úpravám získáme na pracovní ploše více místa k vkládání nejpoužívanějších aplikací formou zástupců. Práce se tak stává spolehlivější a přesnější. Odpadá nutnost jít do nabídky Start a z volby Všechny programy se dostat k dané aplikaci. Proto se nejpoužívanější aplikace umísťují na plochu formou zástupců, a stiskem daného písmena systém sám nabízí aplikace začínající právě stisknutým písmenem.

Stejná metoda se používá i při hledání souborů v daném adresáři. Od Windows 7 a výše a u počítačů Mac je pak efektivní možnost vyhledávat daný program či nastavení psaní do vyhledávače systému, systém sám nabízí dané možnosti. Napíšeme-li např. „uži“, systém nám ukáže všechny aplikace a nastavení, kde se skrývá slovo „uživatel“, a volby uživatelských nastavení. Odečítač obrazovky dále dokáže spolupracovat s Braillskými řádky, které jsou neocenitelné právě pro programátory, překladatele a pro psaní not. Hlasový výstup čte informaci hlasem. Takže tímto způsobem je nereálné psát kód. Počítač by přečetl šestkrát závorku a některé znaky v interpunkci by třeba nečetl vůbec. A právě Braillský řádek je pro tyhle účely neocenitelnou pomůckou. Kde se nachází aktuálně kurzor na obrazovce, je v Braillově písmu zobrazeno na řádku. Jedině tak je možné řádek prohlédnout celý, aniž bychom pohybovali kurzorem a spočítali si počet závorek při ukončení seznamu. Jsou tak viditelné všechny znaky včetně mezer. Práce bez zrakové kontroly je ale vždy pomalejší, než práce vidící osoby právě kvůli tomu, že uživatel musí vše projít ručně, když se chce k nějaké volbě dostat. Je zde důležité si pamatovat, kde se jaká funkce nachází, a k tomu je třeba správně popisovat menu, aby bylo zřejmé, čeho se týkají funkce právě v daném menu. Právě představitost, kde je co na obrazovce, je asi nejtěžší, protože uživatel neví, zda je údaj nahoře, dole, vlevo či vpravo. Tenhle údaj je nepodstatný, když nepočítáme začátek

a konec textu či webové stránky. Proto informace, že vpravo je tabulka a požadovaná informace uvedená v ní, je pro nevidomého nepoužitelná. Nevidomý musí dostat informaci, kde se nachází prvek dané položky. Příklad: volbu Kopírovat najdete v Hlavním menu a submenu Úpravy. Hodně využívanou vlastností je i již zmíněná volba menu klávesou F10 či klávesou Tab či volba kontextového menu. Správná terminologie je též podstatná, jak v samotných programech, tak i příručkách. Je důležité, aby už z názvu bylo patrné, co se pod danou volbou nachází.

1.1 Odečítače obrazovky

Odečítač obrazovky je základ osobního počítače či notebooku. Jedná se o program, který odesílá právě aktuální textovou informaci (položka menu, zapisovaný text, prvek dialogového okna atd.) hlasové syntéze k vyslovení nebo Braillovskému řádku k zobrazení ve slepeckém písmu.

Odečítač obrazovky zpravidla tvoří jak samotný odečítač, tak hlasová syntéza, pomocí níž je vyčítán obsah obrazovky. Dnešní odečítače již obsahují přímo v instalátoru český hlas nebo je možné jej doinstalovat dodatečně. Odečítač má pro nejpoužívanější aplikace vytvořeny skripty, které se vyvolají při spouštění daného programu. Skripty obsahují prvky, informace o barvě rozlišení pozadí programu a textové popisky. Dále mohou obsahovat pojmenované grafické prvky, které byly ozvučeny. Skripty obsahují nabídky, kde uživatel může vyčíst konkrétní údaje, chat v programu Skype či v odečítači WinMonitor, díky nabídce v menu navigátoru pro Word je možné si nechat zobrazit slova dokumentu, která obsahují pravopisné chyby. Odečítače obrazovky obsahují zvukovou myš či virtuální kurzor. Díky těmto vlastnostem uživatel může dostat kurzor tam, kam mu to nedovolí kurzorové šipky, a možnost kliknout na daný objekt. Je možné též nastavit rozsah tohoto kurzoru. Odečítače můžeme rozdělit do dvou kategorií: komerční a nekomerční (svobodné).

K nejpoužívanějším odečítačům v České republice patří odečítací programy: NVDA – open source a komerční odečítače JAWS a WinMonitor. Rozdíl je v podpoře daného odečítače. U nekomerčního NVDA, který je financován pouze z příspěvků, není

jisté, kdy se vývoj zastaví. Produkt je stále financován vývojovým týmem komunity Mozilla Foundations. WinMonitor a JAWS poskytují po zakoupení podporu. Podpora spočívá v nových úpravách odečítacího programu či daného skriptu aplikací. Při problémech s odečítači je poskytována podpora formou konzultací s pracovníky společností v dané zemi, příp. existuje možnost servisního zásahu pomocí vzdálené správy.

1.1.1 Svobodné odečítače obrazovky

NVDA – open source

NVDA je volně šiřitelný odečítač obrazovky. Lze jej instalovat do jakéhokoli 32 i 64 bitového operačního systému Microsoft. Existuje i portable verze, neboť odečítač nepotřebuje ovladač k zachytávání obrazu. Tuto verzi je možné mít na flash disku a použít ji na jakémkoli PC, který je opatřený zvukovou kartou. Je podporována několika desítkami světových jazyků. Čeština využívá hlasový výstup eSpeak. ESpeak pracuje v rozhraní SAPI 4. Odečítač je neustále vyvíjen a spolupracuje se všemi aplikacemi od Microsoftu. Dále je zde podpora internetových prohlížečů Mozilla Firefox a Microsoft Internet Explorer. Významná podpora je i ve Visual Studiu, zde je velmi dobře vyčítán text kódu, ovšem vyčítání některých položek z menu nebo ze seznamu voleb není v některých případech přesné nebo nedochází k vyčtení vůbec, případně je vyčtení jen částečné. NVDA pracuje s většinou Braillových řádků. Za hlavní jeho klad považuji přenositelnost, dobrou práci v internetových prohlížečích, oznamování pravopisných chyb v Microsoft Word. Práce s korekturou je zde celkem náročná, chyby jsou vyčítány při plynulém čtení textu, či čtení textu po řádcích či slovech a nelze zobrazit jen seznam chybných slov.

Je zde velice dobře propracována nabídka nastavení, která je přehledná. Dobrou vlastností je vrácení všech nastavení do továrního. NVDA obsahuje možnost práce s virtuálním kurzorem a nastavení a možnost kliknutí myši pomocí klávesnice. Práce s myší ale je možná jen v některých případech.

Práce v internetových prohlížečích je vynikající. Webová stránka je načtena do okna odečítače a pomocí písmen klávesnice je možné se pohybovat po odkazech nadpisů editačních polí, seznamech apod. Není třeba pomoci některé klávesové zkratky otevřít seznam vyčtení odkazů či nadpisů. Jednotlivá písmena umožňují pohyb po určitých prvcích internetové stránky, např. E=editační pole, N=nadpisy. Odečítač lze rozšířit o pomocné doplňky, aby uměl rozeznat text z obrázku v pdf dokumentu. Dále je zde možné využít důležité možnosti nastavení výslovnosti interpunkce, možnosti úprav řečových slovníků. Jak už bylo zmíněno, možná je instalace či vytvoření přenosné kopie SW. Čtecí kurzor se pohybuje po celé obrazovce, je tak možné získat informace o tlačítkách, která nejsou ozvučena, nebo se nenacházejí ve standardním menu aplikací. Produkt je financován z dobrovolných příspěvků, proto je zde větší riziko ukončení vývoje.

1.1.2 Komerční odečítače obrazovky

WinMonitor

WinMonitor je odečítač obrazovky českého autora Libora Douška z brněnské společnosti ACE Design, která tento odečítač dodává. Podporuje jak Braillovské řádky, tak Braillovské klávesnice díky Sam Acces Manageru od společnosti Dolphin. Sam Acces Manager je manager ovladačů pro Braillovské řádky a Braillovské klávesnice. Je zde možnost tvorby jak vlastních skriptů, tak je možné dozvučení vlastní aplikace, aby byly korektně sdělovány informace, na jakém editačním poli se nacházíme, či tvorba popisků těchto polí. Dále jsou ozvučeny aplikace sady Microsoft Office a běžně používané aplikace jako Skype, Miranda či aplikace pro čtení textu ze skeneru. Odečítač též obsahuje zvukovou myš, kterou se lze pohybovat po obrazovce, a tak kliknout na tlačítka či prvky, které nejsou standardně z menu klávesnice dostupné. Dobrou vlastností je klávesová zkratka Ctrl+Shift +q, která zobrazí celé okno obrazovky a je možné kliknout Enterem i na nepopsané tlačítko. SW obsahuje i vlastní hlas, ale není tak kvalitní jako dostupné hlasy, které se dodávají, proto uživatelé k odečítači instalují ještě zvlášť hlasy. V Microsoft Word považují za skvělou možnost zvolit si seznamy pravopisných chyb, a tak se pohybovat pouze v tomto seznamu a chyby opravovat.

V internetových prohlížečích je práce obdobná jako s NVDA, možnost práce s nadpisy, s odkazy, s texty. Existuje zde navíc možnost si nechat zobrazit jen odkazy a ty si prohlížet, to samé platí i u nadpisů, ostatní elementy z okna jsou odstraněny, a tak je práce v některých případech rychlejší. Není zde ale přesná práce s formuláři. Občas se stává, že se přesuneme na editační pole a při psaní textu se kurzor přesune na jiné místo stránky.

Hodně využívanou vlastností je ozvučení různými zvuky situace, zda jsme na tlačítku zatrhacím seznam při mazání písmena, nebo zvukové oznámení velkého písmena. Důležitou vlastností je dále ohlašování postupu. Když probíhá stahování či kopírování dat či jiná operace, u které je třeba čekat, odečítač produkuje tón, který se zvyšuje v závislosti na průběhu dané akce. Nejvyšší tón signalizuje blížící se konec kopírování či jiné akce.

JAWS (výrobce Freedom Scientific)

JAWS je profesionální odečítač obrazovky. Do České republiky jej dodává společnost Galop. Pracuje se všemi Braillovými řádky. Je zde přesná práce na internetových stránkách, zejména zaměření kurzoru na prvky stránky je přesné. Obsahuje mnoho hlasových syntéz v mnoha jazycích. Propracovaná je i práce s kancelářskými aplikacemi Microsoft Office a práce s tabulkami. Opět zde existuje možnost korektury, ale jen za předpokladu, pokud je v dokumentu více mezer či není uzavřená závorka. Neobsahuje však možnost nechat si zobrazit seznam pravopisných chyb, a tak lépe zpřístupnit jejich editaci. Umožňuje ale vytvářet své vlastní profily nastavení. V nastavení je možné změnit i detaily, zda u internetového odkazu má číst, že jde o odkaz, zda má přečíst i jeho adresu; odečítač může, když je kurzor na odkaze, produkovat zvuk daného prvku nebo může přečíst jen popisek odkazu. I zde je dobře přístupná možnost práce s Visual Studiem. Při práci s těmito aplikacemi je stále mnoho situací, kdy je pomoc vidící osoby nezbytná. Je možné vytvářet schémata výřečnosti. Jako jediný odečítač obrazovky umožňuje přístup ke vzdálené ploše jiného počítače. Funkce JAWS Tandem zajistí uživateli ovládat jiný PC na dálku bez zrakové kontroly. Z ovládaného počítače je přenášen nejen hlas, ale i zvuky systému a zvuky odečítače.

To považuji za velkou výhodu, konečně může i nevidomý provádět vzdálenou správu počítače. Právě tato možnost nebyla prozatím nabízena žádným dostupným způsobem. Díky tomuto odečítači je možné poskytnout např. technickou podporu na dálku. Za poplatek je možné funkci rozšířit na přímý tandem, což umožňuje správu i v zabezpečené firemní síti.

Velmi dobře je zde propracovaná tvorba skriptů pro aplikace, které nemají buď standardní prvky, nebo u kterých JAWS nepozná, o který prvek se jedná. Oknům můžeme přiřadit třídu, a tak určit, jak má JAWS s daným oknem či prvkem pracovat. Díky tomu můžeme u vícepoložkového seznamu vytvořit rozbalovací nabídku, kde si pak můžeme šipkami přesně vybírat hodnotu dané funkce. Příklad bude uveden ve skriptování kalendáře. Pomocí svého skriptu můžeme vyvolat skript jiný, a tím provést např. kliknutí myši při stisku definované klávesové zkratky.

1.2. Přístupnost Mac OS X

Dalším zajímavým systémem z oblasti přístupnosti je operační systém Mac OS X. Ten jako jediný obsahuje odečítač obrazovky přímo v systému. V praxi to znamená, že pokud zakoupíme jakékoli zařízení od Apple, ať se jedná o notebook, iPad nebo iPhone, zařízení obsahují odečítač Voice over. Ten je již přímo v systému a ozvučuje většinu aplikací daného zařízení výrobce. Stačí jen v nastavení zařízení Voice over povolit a nevidomý může pracovat. Odpadá tedy nákup drahého SW, jak je tomu u odečítačů ve Windows. Není nutná žádná instalace, veškeré komponenty včetně skriptů aplikací jsou v systému již od výrobce. S odečítačem je dále možné používat mnoho aplikací třetích stran. Zpřístupněné je jak dotekové ovládání, tak ovládání pomocí klávesnice. Velká výhoda je v pořizovací ceně, stačí jen koupit zařízení a vše je kompletně k dispozici. Voice over dále obsahuje i ovladače Braillových řádků. Řádek tak stačí připojit a zařízení samo detekuje správný ovladač; řádek je tak během okamžiku připraven k práci. Pro některé uživatele může být překážka, že se jedná o méně rozšířený operační systém. Další překážkou může být, že pro některé příkazy je třeba stisknout až 4 klávesy najednou, což může některým lidem činit problémy. Dalším nedostatkem je, že není jednoduchá aplikace pro čtení textu ze skeneru, neexistuje zatím

aplikace, která by byla schopná číst texty pohodlně jako ve Windows. Těmito aplikacemi je myšlen SW, který umí číst plynulý txt soubor, knihu či jiný dlouhý text, vytvářet v něm záložky a později se tak v textu lépe orientovat. Velice propracovaná je však práce na internetu. Webová stránka je načtena dvakrát rychleji, než ji načítají odečítače ve Windows.

1.3 Přístupnost Linux

V systému Linux je nejpřístupnější distribuce Ubuntu s grafickým prostředím Gnome. Ovládání je podobné jako ve Windows právě díky grafickému rozhraní. Jako odečítač obrazovky je zde použit odečítač Orca. Je zde méně hlasových syntetizátorů, nejrychlejší na odezvu je zatím hlasový výstup eSpeak. Z hlediska přístupnosti je nejlepší se zabývat pouze konkrétní distribucí Linuxu. Byl by totiž problém napsat knihovnu, která by byla pro 40 distribucí. I kdyby se knihovny pro všechny distribuce podařily napsat, byly by v odečítači nepoužitelné, protože by byla velice špatná odezva hlasového výstupu a Orca by musela načítat mnoho knihoven.

Další distribucí pro nevidomé je Vinux, který již v instalátoru obsahuje hlasovou syntézu. Na slušné úrovni je ozvučen internetový prohlížeč Firefox, kde je už možná i klávesová navigace po nadpisech, rámcích, odkazech, seznámech atd. Zatím chybí například ozvučení komunikačních programů jako Skype či kvalitní OCR program. Linux je sice pro nevidomé použitelný, ale v odečítači jsou stále chyby, díky kterým by málokdo použil tento systém jako primární.

V nových přenosných distribucích Ubuntu s odečítačem jsou problémy s odezvou některých hlasových syntéz. Z části už je ozvučen i program PDF Reader, který bere pdf jako objekt a chová se k pdf dokumentu, jako kdyby to byla webová stránka. Díky tomu si nevidomý může daný dokument prohlédnout. Chybí však velmi důležitá možnost čtení souvislého textu, knihy či jiného dlouhého dokumentu. Myslím tím speciální aplikaci, která by si pamatovala pomocí záložek, kde jste v čtení právě skončili. Co se týče přesnější navigace v textovém dokumentu, už je možné se pohybovat po slovech či písmenech v řádcích. Využití Linuxu je tak možné použít např.

při situaci, kdy nám přestane fungovat náš primární operační systém. Distribuci můžeme bootovat z USB a po stisku Ctrl +S Linux začne mluvit a my se tak můžeme podívat na strukturu disku apod. Odečítač je stále ve vývoji a je třeba zapracovat hlavně na přístupnosti aplikací třetích stran, které zatím téměř ozvučeny nejsou.

1.4 Braillovský řádek

U Braillovského řádku se jedná o zařízení ve tvaru kvádrů, jeho horní strana je osazena řadou piezoelektrických elementů k zobrazení jednotlivých znaků Braillovské abecedy. Podle počtu elementů rozlišujeme řádky 20, 40 nebo 80 znakové. Kromě řady elementů vytvářejících znaky Braillova písma řádek obsahuje nad každým znakem naváděcí klávesu. Tyto klávesy slouží k nastavení kurzoru na obrazovce právě na daný znak. Tedy kurzor řádku a obrazovky je na stejném místě.

Braillovské řádky dále nabízí navigační klávesy, které slouží k pohybu v dokumentech a v tabulkách webových stránek. Již zmíněné elementy jsou právě na řádcích nejdražší, a tak se ceny řádků pohybují od 80 000 Kč. Braillovské řádky jsou určeny studentům, programátorům a lidem, kteří pracují s velkým množstvím textů. Právě u programování a psaní html kódu je Braillovský řádek nepostradatelný. Při psaní můžeme spočítat závorky nebo ověřit správnost znaků, které běžnou interpunkcí odečítačů vyčítány nejsou. Při přednesu hlasovým výstupem by byla práce zpomalená, kdyby odečítač četl u každého slova znaménka mezery apod. Díky zobrazení v Braillově písmu je přesnější opravovat gramatické chyby v textech přímo při psaní. Hlasový výstup slovo přečte, ale z reprezentace hlasem v hodně případech není poznat gramatická chyba. U zápisu matematiky, chemie a fyziky je Braillovský řádek nezastupitelný, protože např. výrazy matematiky není možné v počítači hlasovým výstupem zpřístupnit.

Braillovské řádky dále využívají překladatelé. Odečítač může číst anglickým hlasem, jak se text čte, ale tímto přednesem se uživatel nedozví, jak se text píše. A právě Braillovský řádek umí zobrazit text i v psaném jazyce, a tak je práce s jazykem kompletně přístupná.

V poslední době jsou velmi moderní přenosné Braillovské řádky. Ty jsou napájeny jak přes USB rozhraní, tak mohou disponovat vlastním napájecím adaptérem. K počítači, telefonu či PDA se připojují pomocí USB či technologie Bluetooth. Braillovský řádek můžeme mít pod klávesnicí, a tak mít k řádku okamžitý přístup. Aby byl Braillovský řádek ovladatelný, musí být v systému nainstalován odečítač obrazovky, který již obsahuje ovladače pro většinu Braillovských řádků.

Zajímavý je Braillovský řádek Alva BC640 od holandského výrobce Optelec. Tento řádek je možné díky akumulátoru využít i na cestách a lze k němu připojit audio braillovský modul. Modul obsahuje jak Braillovskou klávesnici, tak klávesy Esc, Control, Windows a Enter. V řádku se nachází flash paměť, která se chová jako externí disk. Na tomto úložišti je pak možné mít nakopírovanou například přenosnou verzi NVDA. V modulu jsou integrovány reproduktory a vstup pro sluchátka. Uvnitř je též integrovaná zvuková karta, takže připojený PC může těchto reproduktorů využít. Považuji to za velký klad Braillovského řádku, protože tak lze pracovat i s počítačem, ve kterém zvuková karta není či je ztlumený zvuk, a právě díky této variantě lze i takové PC použít.

Řádek v sobě obsahuje software, který se spustí po zapnutí zobrazovače. Obsahuje přímou možnost nastavení řádku, intenzitu Braillovských bodů, nastavení času, data, upřednostněného připojeného zařízení, upgrade Firmware. Při připojeném modulu je dále SW řádku doplněn o zápisník, kde lze vpisovat text přímo z Braillovské klávesnice, bez použití počítače. Braillovskou klávesnicí, je-li připojena s řádkem k PC, je možné použít i jako klávesnici počítače. Modul s řádkem je spojen dvěma šroubky, takže tvoří kompaktní celek.

2. Programové vybavení nevidomého uživatele

Mezi další softwary, které ulehčují práci s počítačem, můžeme zařadit takzvané asistenční nástroje. Jedná se o balíky aplikací, které jsou propojeny s knihovнами standardních aplikací, přebírají od nich data uživateli reprezentovaná v přehledných aplikacích. Jako příklad můžeme uvést Winmenu či Asistent. Jedná se o nejznámější balíky aplikací, které spolupracují jak se skenerem, tak i aplikacemi jako Skype. V standardním programu stačí pouze povolit přístup aplikace k datům a daná aplikace data zobrazí v jednoduchém seznamu, např. kontakty, emailové zprávy, RSS kanály apod. Tyto nástroje jsou určeny pro začátečníky i pokročilé uživatele, kteří chtějí rychlý přístup k datům. Výrazný problém se objevuje v ozvučení standardních aplikací.

V mnoha případech při vydání aktualizace či nové verze SW přestane skript odečítače obrazovky s danou aplikací pracovat. Program se pak stává špatně použitelným, v některých případech zcela nepoužitelným. Stačí změny pozadí, rozlišení či grafické nepopsané prvky a daný odečítač už nedokáže vyčíst dané informace. Screen readersy mají podporu, která zajišťuje i aktualizaci skriptů nejčastěji používaných programů, proto jsou použitelné i v nových verzích či změnách SW. Pokud ale uživateli podpora placeného odečítače skončí, může se dostat do potíží s programy, které používal a na které je zvyklý. Nárok na kompenzační pomůcku je jednou za 5 let. Za tu dobu většina aplikací dosáhne takových změn, že starší verze ozvučení si již nemusí poradit s předchozími úpravami skriptů.

Další vlastností je, že počítač se může chovat jako čtecí zařízení, připojíme-li k němu skener. Dokument snímáme pomocí skeneru, kde jej hlasový výstup pak vyčítá uživateli hlasem či zobrazuje na Braillovém řádku. Pro nevidomého je zde neocenitelná možnost převodu pdf dokumentů, které jsou někdy špatně přístupné do doc či jiného textového formátu. Další neméně důležitou vlastností je získat text, třeba z jpg obrázku.

3. Testování kalendáře uživateli, metoda ovládání a cíle

3.1 Návrh ovladatelnosti aplikace

Před psaním této aplikace jsem dlouho uvažoval nad tím, jaký způsob ovládání zvolit. Nabízela se možnost klasického okna s možností výběru funkcí z menu. Tuto možnost jsem nakonec nevyužil, protože jsem chtěl implementovat nový způsob ovládání aplikace z jakékoli aplikace Windows. Proto byl zvolen systém zobrazení okna, ze kterého je informace vyčtena, a okno se opět skryje. Tak pomocí klávesových zkratk může uživatel program ovládat, ať je v jakékoli aplikaci operačního systému. Aplikace přinesla nový typ ovládání, který není tak často používán pro tuto aplikaci a je dle mého názoru nejvhodnější. Jak je popsáno níže, tento způsob ovládání uživatelé velice dobře přijali, což považuji za velmi pozitivní. Právě způsob ovládání uživatele nejvíc potěšil, protože brzy s aplikací bez problémů pracovali.

3.2 Ověření aplikace uživateli a testování

Aplikace byla poskytnuta 20 uživatelům, kteří ji začali používat, a tak bylo možné reagovat na jejich podněty a připomínky. První uživatelé ocenili právě zmíněný způsob ovládání. Většina uživatelů to odůvodnila tím, že nemusí mít otevřené další okno s aplikací. Dále bylo upozorněno na to, že má-li uživatel otevřených mnoho aplikací najednou, orientace není tak rychlá. Pokud uživatel chtěl do jiné aplikace, tak buď ji musel spustit z nabídky Start, či přejít do okna, kde měl aplikaci otevřenou. Dále uživatelé ocenili fakt, že aplikace neobtěžuje žádným spuštěným oknem a že při práci mohou kalendář využít; on se pak skryje a jeho přítomnost je neobtěžuje, přičemž aplikace je i nadále připravena k použití. Další pozitivní informace je, že vše se zobrazuje i na Braillovém řádku. Jeden návrh byl do kalendáře integrovat adresář, který by umožňoval základní práci s kontakty. Tuto možnost jsem do aplikace implementoval.

Při psaní aplikace bylo zajímavé sledovat reakci odcítačů. Proto byly pro zlepšení aplikace vytvořeny skripty pro odcítač JAWS, které oznámí i informaci, zda je člověk v rozbalovacím seznamu či jsou popsána editační pole. Tak uživatel přesně ví,

co do pole má zadat. Aplikace přinesla jednoduchý nástroj dostupný všem, který umožňuje základní funkce s kalendářovými funkcemi. Aplikaci je dále možné rozšiřovat, avšak pouze dokud nebudou obsazené klávesové zkratky. Bylo by pak možné vytvářet i různá podmenu, ale práce už by nebyla tak rychlá a přehledná.

Cílem aplikace bylo vytvořit nástroj, který kdekoli se uživatel na obrazovce nachází, stiskne klávesovou zkratku, dozví se aktuální informaci či napíše poznámku a aplikace se pak skryje. Menu je možné kdykoli vyvolat. Aplikace však není vhodná pro ty, kteří pracují s myší, protože vždycky musí kvůli menu stisknout klávesovou zkratku.

Dalším cílem bylo zpřístupnit jednoduše a kdykoli bez použití internetu kalendářové funkce, aniž by se musel instalovat nový hlas či použít konkrétní odečítač. Seznam funkcí jsem zvolil na základě užívání daných funkcí. Mnoho uživatelů mi dále sdělilo, že v PC vůbec nevyužijí stopky.

Aplikaci jsem představil na přednášce pobočky Sons v Prostějově, kde si uživatelé mohli aplikaci vyzkoušet. I ze strany těchto uživatelů jsem získal pozitivní reakce, právě na ovladatelnost a jednoduchost aplikace. Dále mi byla sdělena informace, že aplikace neobsahuje zbytečné funkce; u mnoha aplikací uživatelé přiznali, že polovinu funkcí nakonec nevyužijí.

Další inspirací k vytvoření této aplikace byly kalendáře, které jsou napsané pro mobilní platformy, konkrétně Symbian. Na těchto platformách jsou právě dané aplikace velmi přívětivé pro ovládání, to byl impuls vytvořit jednoduchou aplikaci i pro Windows. V textu rozepisují klasické ovládání PC nevidomou osobou proto, aby bylo vidět rozdíl mezi klasickým ovládáním a ovládáním této aplikace. Výběr klávesových zkratk se rovněž ukázal jako vhodný. Ze strany uživatelů mi bylo sděleno, že je jednoduché si zkratky zapamatovat ve zvoleném tvaru Ctrl + funkční klávesy. To bylo přijato velmi pozitivně, protože uživatel si nemusí pamatovat jiné klávesové kombinace.

Při testování aplikace byl kladen důraz na detaily, jak nejrychleji zobrazit údaje např. v adresáři či textu poznámky. Jako příklad uvádím, že máme-li v adresáři či

upozornění záznam, seznam záznamů můžeme procházet kurzorovými šipkami. Pokud ale chceme přečíst všechny údaje z adresy či upozornění, nebo pokud chceme daný údaj zkopírovat do schránky, není to možné. Proto byla přidána možnost, že pokud jsme na kontaktu či upozornění a stiskneme mezerník, objeví se nám možnost zobrazit záznam, jako bychom aktivovali tlačítko Zobrazit záznam. To práci výrazně zrychluje, uživatel na kontaktu či jiné informaci nemusí tlačítkem tab hledat tlačítko Zobrazit a volbu aktivuje rovnou mezerníkem.

V adresáři je též možné hledat kontakty dle stisknutého písmene. Při stisku l je přečten první kontakt se jménem začínajícím písmenem l. Pokud písmena doplňujeme, zužuje se seznam výsledků. Tato možnost opět zrychlí práci s aplikací, uživatel nemusí šipkami procházet celý adresář. Díky funkci Zobrazit, která je jak na tlačítku zobrazit, tak pod mezerníkem při vybraném záznamu, můžeme daný řádek adresy zkopírovat do schránky pomocí Ctrl + C. Do schránky se pak vloží celý obsah řádku, na kterém je kurzor, např. telefonní číslo pro pozdější použití.

3.3 Problémy s přístupností při psaní aplikace

Jedním z problémů je zobrazení okna, některé hlasy nepřechetly informaci celou nebo ji nepřechetly vůbec. Řešením bylo vytvoření skriptů pro JAWS. Další otázka byla, zda při zadávání data zvolit výběr dne měsíce a roku pomocí kurzorových šipek či vyzvat uživatele, aby dané údaje do daného řádku napsal. Vepsání údaje se mi zdálo rychlejší a dále byla stoprocentní jistota, že editační řádek s popiskem vyčte každý odečítač. Informace se při začátku zobrazovala jen v nadpisu okna. Díky odečítači JAWS, který tuto situaci považoval za nestandardní, byl text, který má být oznámen, zobrazen jak v nadpisu okna, tak v okně jako text. Ostatní problémy jako nečtení přesné informace při výběru šipek v seznamech hodnot vyřešily skripty k odečítačům.

V aplikaci byl kladen důraz, aby při ovládání nebylo třeba používat virtuální kurzory či jiné podobné funkce jako zvukovou myš. Vedly mě k tomu dva důvody. Prvním důvodem je, že ne všechny odečítače obrazovky podporují virtuální kurzory a podobné vlastnosti. Jako příklad uvádím odečítač Pocket Reader, který slouží

k základnímu ovládní funkcí operačního systému Windows. Dál nebylo možné okno nechat zobrazené na takovou dobu, aby uživatel prošel nějakou funkcí a odečítač tak vyčetl daný text. Práce by se tím zpomalila a nebyla tak komfortní. Aby bylo možné vyčíst text kurzorem manuálně, muselo by být okno, které reprezentuje dané informace, zobrazeno tak dlouho, dokud by jej uživatel sám neukončil. V tomto případě to považuji za zbytečné, když odečítače obrazovky automaticky při vyvolání funkce klávesovou zkratkou vyčtou dané údaje.

Dále bylo třeba zvolit správný systém klávesových zkratk, který jsem zvolil dle následujících aspektů. Kombinace zkratk nesmí mít velké nároky na zručnost při stisku kláves. Pokud by uživatel měl stisknout například 3 klávesy najednou, mohlo to být pro některé uživatele obtížné. Dále bylo třeba vybrat zkratky, které mají nějakou strukturu, své pořadí. Proto byla zvolena kombinace Ctrl + funkční klávesy. Nejvíce pozornosti pak muselo být věnováno klávesovým zkratkám samotných odečítačů obrazovky. Odečítače se ovládají právě pomocí klávesových zkratk a práce na nich je přímo závislá. Bylo třeba projít všechny používané odečítače obrazovky a najít takové zkratky, které žádný odečítač nemá implementované. Proto se nemůže stát, že by klávesová zkratka kolidovala s klávesovou zkratkou některého z odečítačů obrazovky.

3.4 Microsoft Speech API

Rozhraní API je distribuováno buď v rozhraní Sdk řečových syntéz, nebo jako součást operačního systému Windows. SAPI je obsaženo v sadě Microsoft Office, Microsoft Agent i v Microsoft Speech Server. Slouží k používání hlasové syntézy a rozpoznávání řeči. API je možné implementovat do různých programovacích jazyků. Speech API lze implementovat do aplikací a rozhraní. API můžeme definovat jako rozhraní, které zprostředkovává komunikaci mezi aplikací a řečovou syntézou, či pluginy pro rozpoznávání řeči. Někdy enginem může být myšlen konkrétní hlas včetně jeho nastavení, který produkuje čtení informace na výstupu. Microsoft Speech API verze 1 až 4 může komunikovat přímo s aplikacemi pomocí pluginů. V SAPI 5 nekomunikuje rozhraní přímo s aplikací, ale pomocí runtime součástí sapi.dll, která komunikuje se syntetizátorem. V SAPI 5 je dbáno na to, aby aplikace pracovaly s řečí

právě pomocí runtime sapi.dll, a tak nedochází k ovlivňování aplikací. Speech API již nemusí zasahovat do konkrétní aplikace. Hlavní vlastnosti rozpoznávání řeči běží v samostatném procesu aplikace (sapisvr.exe). Aplikace, které využívají rozpoznávání řeči, tak mohou komunikovat pouze s touto jednou instancí.

Dále Speech API obsahuje možnost diktování textu, která může být ovlivněna vytvořenými slovníky s nahranými vzorky hlasu či jednotlivými slovy nebo slabikami. Hlasový výstup dokáže syntézou řeči vyčítat daný text z obrazovky. Audio rozhraní Runtime obsahuje možnosti přenesení zvuku z mikrofону či hlasového výstupu do reproduktorů, a tak vytvoření zvukové reprezentace například čtením textů. Výslovnost řeči se pak může odkazovat buď na přidané slovníky, které jsou v aplikacích, nebo na přidání nových slov do slovníků uživatelem. Základní hlasový výstup, který je dodáván již s Windows XP v SAPI 5, je Microsoft Sam. Windows Vista obsahuje hlas Microsoft Anna nahrazující hlas Microsoft Sam. Anna zní přirozeněji lidské řeči. Některé starší knihovny SAPI trpí nesprávnou prací s pamětí, které způsobují fragmentaci paměti. Nejpoužívanější rozhraní v současnosti je SAPI 5 - je rychlejší a je lépe optimalizováno pro výkon.

Hlasy pracující v rozhraní 4 mají rychlou odezvu při práci. To se nejvíce projevuje při procházení webových stránek, psaní textů nebo jejich prohlížení. Nejvýraznější odezva je znatelná při psaní. Stisk kláves ihned ohlásí údaj a nedochází k žádným prodlevám mezi stiskem klávesy a přečtení znaku či informace z obrazovky. Mezi české hlasy SAPI 4 patří například eSpeak či hlasový výstup Hlas. Tyto hlasy mají vynikající odezvu při práci, hlasy však nejsou přirozené lidské řeči. Dalo by se říci, že zní jako robot. Nové verze těchto hlasů již dokážou lépe intonovat při čtení, ale hlas stále není přirozený. SAPI 4 hlasy též mohou mít problémy při připojení některých zvukových karet, např. hlas může v některých případech narušit konzistenci odečítače a ten je pak potřeba znovu spustit. Tímto problémem s externími zvukovými kartami či sluchátky, které se jako zvuková karta chovají, trpí starší verze odečítačů včetně zmíněných hlasů. Toto lze řešit umožněním podpory odečítače přepnutím zvukové karty, a tím zamezit těmto problémům. Přepínání zvukových karet podporuje odečítač JAWS. Pro nevidomého je výhodou možná realizace Skype hovoru přes sluchátka

a z reproduktorů poslouchání hlasového výstupu a současně možnost pracovat; hlas tak nemluví do komunikace mezi uživateli.

SAPI 5 hlasy jsou mnohem kvalitnější, některé se blíží lidské řeči. Nejblíže lidské řeči je v dnešní době hlasový výstup Infovox Eliška od švédské firmy Acapella. SAPI 5 hlasy nemají tak rychlou odezvu při práci, při denním užívání lze rozdíly v rychlosti rozpoznat. Ideální hlasový výstup by měl být rychlý na odezvu a srozumitelný. Čím je hlas kvalitnější, tím potřebuje více paměti a odezva není tak rychlá jako u hlasů SAPI 4. Tyto hlasy nemají problém v komunikaci s externími zvukovými kartami, ale i zde platí, že je lepší, aby odečítač podporoval na externí kartu.

3.5 Odečítače obrazovky a technologie hlasů

Dnes se používá odečítač obrazovky Microsoft Active Accessibility. Bohužel systém není jednotný, rozhraní je používáno na různých místech. Vývojář programu má k dispozici Accessibility a může či nemusí tento odečítač implementovat do své aplikace.

Většinou se používá kombinace buď s přístupem přes standardní API, nebo přes Microsoft Active Accessibility, což je problematické. Někdy může být získán text či kontext. Pokud je Accessibility špatně použito, jsou získány informace, které uživatel nepotřebuje. Jméno a popis někdy nejsou hodnoty na daném místě. Například místo jména okna může být přečten jeho obsah. V základě se vycházelo ze standardního API rozhraní. Pokud byl na obrazovce seznam, vyvolala se funkce, která vrátila konkrétní umístění položky daného seznamu. V editačním poli byl vrácen text. V minulosti existovala nabídka Start dostupná pouze pomocí Microsoft Active Accessibility. To bylo pomalejší a při chybě v serveru Accessibility mohlo dojít k problémům. Modernější způsob práce odečítače je pomocí video driveru. Tuto metodu používá například odečítač JAWS, nemusí se tak spoléhat na Microsoft Active Accessibility. Video driver pracuje dvěma metodami. První se nazývá zřetězení. Informace je posílána grafické kartě právě přes video driver odečítače.

Druhá metoda se nazývá zrcadlení. Pokud je v systému více obrazovek, je přidán ještě jeden video driver, kterému se data posílají. Video driver odečítače zachytává pouze text, nepozná tlačítka či jiné prvky. Zda se jedná o tlačítka a prvky, může být tato informace získávána buď přes Microsoft Active Accessibility, nebo v lepším případě vytvořením skriptů k dané aplikaci. Skript driveru sdělí, že konkrétní prvek je např. editační pole či zaškrťávací tlačítko. Tento prvek na dané pozici lze opatřit textovým popiskem. Po spuštění aplikace, když video driver na tuto pozici narazí, odečítač jí nebude považovat za nestandardní a uživateli sdělí konkrétní informaci. U některých aplikací může dojít k překreslení textu, takže uživateli může být zpřístupněna pouze část informace. Tyto situace právě dokážou vyřešit skripty, které při zachytávání obrazovky dají přesnou informaci. Tyto drivery odečítačů mají implementovány rozeznávání tlačítek či prvků dle určitých barev a pozadí, záleží na aplikaci. Pokud se barevné schéma změní, informace nemusí být vyčtena korektně nebo vůbec. Například v editačním poli aplikace Microsoft Word se informace může získávat pomocí com objektu. Com objekt se používá ke komunikaci mezi aplikacemi. Accessibility generuje za vteřinu stovky údajů, které se musí přebírat. Tak může dojít například k situaci, kdy se uživatel nachází v seznamu písem, fokus je na daném typu písma, ale odečítač vyčte jeho velikost. Není zde přesná pozice zaměření, fokus může být na jiném místě než kurzor odečítače.

3.6 Technologie hlasů

U SAPI 4 lze nastavit plynulejší zrychlování a zpomalování hlasu. Za prvé jsou to syntetizéry, které generují frekvenci, pravidla určující generování tónu. Zní jako robot, ale je srozumitelný. Za druhé se jedná o technologie hlasů tyfonové. Skládají se z namluvených kusů řeči, kdy jsou nahrány jednoduché slabiky člověka, slabiky se rozkouskují a zápis se vytvoří pomocí vln. Kousky se následně vloží do databáze. Při generaci řeči se slova zpracovávají přes různé slovníky výslovnosti. Dále se zpracovává intonační křivka, zda má hlas klesat či stoupat, a rychlost řeči. V databázi se pracuje se zapsanými vlnami, díky nimž probíhá generování řeči. Pro tyto účely je těžké vybrat mluvčího; není vhodný hlas, který je příliš akcentovaný či zkreslený. Další

technologie je namluvení velkých kusů textů a ty pak pomocí algoritmů skládat. Tyto hlasy jsou nejkvalitnější, avšak zabírají více paměti Ram až několik GB a je zde potřeba výkonný procesor. U těchto hlasů je třeba najít mluvčího s kvalitní výslovností. Namluvené části již nelze upravovat, tyto hlasy neumožňují nastavení intonace. Kusy namluveného textu se vyhodnocují ručně. Mezi tyto hlasy patří třeba hlas Infovox Eliška od Acapelly.

3.7 Rozdíly mezi SAPI

SAPI 4 je zcela odlišný, není kompatibilní se SAPI 5. SAPI 4 používá svůj formát pro zápis tagů a nastavení. SAPI 5 používá standardní xml formát a xml Parser. Je ponechána velká část formátu, který zpracovává systém a část zpracovává engine. U přepínání hlasů zpracovává údaje systém a předá je rovnou enginu. U SAPI 4 jdou data přímo enginu hlasu a ten je teprve zpracovává. Enginem je myšlen konkrétní hlas. SAPI 5 je zcela Unicode a SAPI 4 používá jak Ansi, tak Unicode, aplikace si vybírá, jaké kódování chce využít. Ve vyhledávání hlasů enumeraci provádí engine. Musíme dát požadavek a API vrací hlasy. V SAPI 5 jsou zápisy hlasů v registrech nezávislé na enginech. Systém SAPI si vybere příslušný hlas, aniž by vyvolal engine. Výběr hlasu je oddělený, nezávislý na enginu. U SAPI 4 enumerace prochází přes všechny hlasy, které jsou v systému. Pokud byl nějaký engine špatně implementován, dojde k poškození všech hlasů a pak nelze s hlasy pracovat. V SAPI 5 je striktně definovaná možnost nastavení rychlosti. U SAPI 4 bylo možné nastavit rozsah na maximum a minimum, a tak bylo možné nastavovat jemnější nastavování rychlosti hlasu. Největší překážka SAPI 4 je, že již není implementován v 64 bytových systémech. Naopak v SAPI 5 je implementováno jak 32, tak 64 bytové rozhraní i enginy najednou. V operačním systému jsou tato rozhraní obě. Pokud se u SAPI 4 nepoužije do Bufferu vteřina před vyslovením, tak hlas nemluví. Je-li třeba reprodukovat krátký zvuk, musí se do jedné vteřiny vložit ticho, aby hlas začal mluvit na výstupu. Dále není dobrá podpora slovníku, je těžší je implementovat.

3.8 Skriptování kalendáře

Pro úplné vyčítání informací z kalendáře bylo třeba vytvořit skripty. Kalendář lze používat i bez nich, ale práce není tak příjemná.

3.8.1 Skripty odečítače JAWS

Kalendář se bez skriptů chová následovně. Po stisku klávesové zkratky není automaticky vyčtena informace z okna. Pokud vyskakovací okno něco oznamuje, též není text vyčten. Pro přečtení dané informace při zobrazeném okně je třeba stisknout Ctrl + šipku dolů. Informace není čtena automaticky. Dále u editačních polí nebyla čtena informace, co se do daného pole má zadat. Jako příklad uvádím přidání nového upozornění do kalendáře. Zde jsou tři řádky, kdy uživatel neví, do kterého řádku má zadat nadpis datum a čas a text upozornění. Tuto situaci bylo možné vyřešit vložením textových popisků do formuláře vedle textových polí s danými údaji. Dále při výběru položky ze seznamu není aktuální hodnota čtena přesně, jako příklad uvádím výběr časového intervalu pravidelného oznamování času. Skripty zajišťují automatická vyčítání okna. Pokud máme nastavený pravidelný časový interval každých 30 minut, po zobrazení okna je informace automaticky vyčtena přesně, jak má být.

Rozbalovací seznam považoval JAWS jako nestandardní element. Po vytvoření skriptu odečítač ví, že se jedná o seznam, a čte přesně, jak kterou položku zvolíme, tak položku, na které stojíme kurzorem. Informace je tedy vyčítána přesně; pokud v seznamu stiskneme šipku, je přesně vyčtena hodnota, která je na obrazovce a na kterou jsme nastavili kurzor. U vyskakovacího okna bylo třeba, aby v něm byl zobrazován text s danou informací, text nesměl být jen v popisku. V popisku nebyl JAWS text vyčítán vůbec. Díky skriptům, které jsou v konfiguračních souborech kalendáře, je možné úplné ozvučení aplikace a po jakémkoli stisku klávesy je vyčtena přesná informace, jak ze seznamu prvků, tak z editačních polí. Je velmi důležité, že uživatel má vyčtenou pouze tu informaci, kterou požaduje. Pokud například uživatel vejde do nápovědy, skript mu popiskem sdělí, že v menu nápovědy se má pohybovat šipkami nahoru a dolů.

Skripty fungují pouze tehdy, je-li v operačním systému nainstalován odečítací program JAWS. Při spuštění kalendáře jsou skripty rovnou připraveny k použití. V programové složce aplikace Kalendář se nachází složka s právě těmito skripty. Obsah složky skripty pro JAWS je třeba nakopírovat do následujícího umístění:

```
C:\Users\jmenouzivatele\AppData\Roaming\Freedom Scientific\
JAWS/j (číslo verze JAWSu\Settings\csy
```

Skripty je nutno překopírovat jen jednou. I po reinstalaci či úplné odinstalaci JAWS včetně jeho součástí zůstanou skripty s nastavením pro kalendář v počítači zachovány. Úplná odinstalace JAWS ponechá soubory se skripty v umístění, které je uvedeno výše. Pokud tedy uživatel začne používat novější verzi odečítače a předchozí verzi odstraní, veškeré nastavení odečítače včetně těchto skriptů zůstanou zachovány i v nové verzi odečítače skript pro JAWS. Tento skript je ilustrační příklad, jak by mělo vypadat ozvučení aplikace, kde nejsou popsána editační pole Kalendáře.

```
[datum_upozornění]
```

```
originy=48
```

```
originx=117
```

```
spokenprompt=datum upozornění
```

```
brailledprompt=datum upozornění
```

```
event1=1|0|0|23
```

```
enable=1
```

```
realwindowtop=271
```

```
realwindowright=677
```

```
realwindowbottom=497
```

```
height=20
```

```
realwindowleft=347
```

echo=-1
relativeto=0
windowbased=1
validationrule1=1|0|2|Přidat upozornění
width=197
priority=5
dpi=96
validationrule2=1|1|2|WindowsForms10.Window.8.app.0.378734a
tutormessage=
[nadpis_upozornění]
originy=79
originx=115
spokenprompt=nadpis upozornění
event1=1|0|0|23
brailledprompt=nadpis upozornění
enable=1
realwindowtop=271
realwindowbottom=497
realwindowright=677
height=20
realwindowleft=347
relativeto=0

echo=-1
windowbased=1
width=200
validationrule1=1|0|2|Přidat upozornění
tutormessage=
validationrule2=1|1|2|WindowsForms10.Window.8.app.0.378734a
dpi=96
priority=5
[text_upozornění]
originy=107
originx=113
spokenprompt=text upozornění
brailledprompt=text upozornění
event1=1|0|0|23
enable=1
realwindowtop=271
realwindowright=677
realwindowbottom=497
height=76
realwindowleft=347
echo=-1
relativeto=0

windowbased=1
validationrule1=1|0|2|Přidat upozornění
width=200
priority=5
dpi=96
validationrule2=1|1|2|WindowsForms10.Window.8.app.0.378734a
tutormessage=
[text_poznámky]
originy=35
originx=16
spokenprompt=text poznámky
event1=1|0|0|23
brailledprompt=text poznámky
enable=1
realwindowtop=66
realwindowbottom=247
realwindowright=416
height=130
realwindowleft=66
relativeto=0
echo=-1
windowbased=1

width=318
validationrule1=1|0|2|Přidat poznámku
tutormessage=po vložení poznámky stiskněte enter
validationrule2=1|1|2|WindowsForms10.Window.8.app.0.378734a
dpi=96
priority=5
[text_poznámky2]
originy=35
originx=16
spokenprompt=
brailledprompt=text poznámky
event1=1|0|0|23
enable=1
realwindowtop=23
realwindowright=373
realwindowbottom=204
height=130
realwindowleft=23
echo=-1
relativeto=0
windowbased=1
validationrule1=1|0|2|Přidat poznámku

width=318
priority=5
dpi=96
validationrule2=1|1|2|WindowsForms10.Window.8.app.0.378734a
tutormessage=
[šipkami__zvolte_pocet_fteřin_minutnıku]
originy=41
originx=137
spokenprompt=šipkami zvolte pocet fteřin minutnıku
brailledprompt=šipkami zvolte pocet fteřin minutnıku
event1=1|0|0|23
enable=1
realwindowtop=45
realwindowright=341
realwindowbottom=158
height=12
realwindowleft=45
echo=-1
relativeto=0
windowbased=1
validationrule1=1|0|2|FromTimer
width=96

priority=5
dpi=96
validationrule2=1|1|2|WindowsForms10.Window.8.app.0.378734a
tutormessage=
[OKNO]
originy=0
originx=0
spokenprompt=
event1=1|0|1|*|0|*/|20
enable=1
realwindowtop=336
event3=1|0|1|*|0|*/|20
realwindowbottom=431
realwindowright=714
event2=1|0|2|2|OKNO|PCCursor
height=95
realwindowleft=310
relativeto=0
echo=-1
windowbased=1
width=404
validationrule1=1|0|2|Dnes má svátek Svatopluk.

validationrule2=1|1|2|WindowsForms10.Window.8.app.0.378734a
dpi=96
priority=5
[ŠÍPKAMI_ZAHÁJÍTE_ČTENÍ_NÁPOVĚDY]
originy=35
originx=16
spokenprompt=ŠÍPKAMI ZAHÁJÍTE ČTENÍ NÁPOVĚDY
brailledprompt=ŠÍPKAMI ZAHÁJÍTE ČTENÍ NÁPOVĚDY
event1=1|0|0|23
enable=1
realwindowtop=181
realwindowright=878
realwindowbottom=587
height=327
realwindowleft=146
echo=-1
relativeto=0
windowbased=1
validationrule1=1|0|2|Nápověda
width=701
priority=5
dpi=96

```
validationrule2=1|1|2|WindowsForms10.Window.8.app.0.378734a  
tutormessage=
```

3.8.2 Skripty odečítače WinMonitor a NVDA

U odečítače NVDA nebylo třeba vytvářet žádné skripty, v kalendáři bylo vše čteno korektně včetně rozbalovacích seznamů, editačních polí i informace z vyskakovacího okna.

U odečítače WinMonitor je situace trochu jiná. Skripty je třeba vytvořit ručně v jednoduchém nastavení. Opět zde uvádím příklad, jak ozvučit editační řádek v aplikaci. Bude lepší použít termín, popis vytvoření ozvučení kalendáře pro WinMonitor. Nejprve se přesuneme na editační pole, které chceme ozvučit. Jako příklad uvádím přidání nové poznámky. V systému stiskneme Ctrl + F1, objeví se editační pole pro psaní poznámky. Teď je třeba stisknout Ctrl + F12 a zvolit Obecné. Šipkou doprava menu rozbalujeme, dále rozbalíme volbu nástroje a klávesou Enter aktivujeme možnost Úprava ozvučení. Zde nic neměníme, tlačítkem Tab zvolíme možnost Další a mezerníkem či Enterem volbu aktivujeme. V tomto kroku je popisován titulek Okna. Zde jsou dvě položky na klávesu a na fokus; my zvolíme fokus. Opět mezerníkem přejdeme na Další. Nyní vybereme možnost Oznámit fokus a Komentář. Dále je zde číslo pozice daného pole na obrazovce. Tento údaj nesmí být měněn, odečítač by pak informaci nevyčetl. Nevěděl by přesnou pozici editačního pole, vkládání textové poznámky. Po stisku tlačítka Další se objeví tlačítko Dokončit, které opět zvolíme, a tlačítko Uložit. Můžeme na editačním poli, které chceme popsat, stisknout rovnou Ctrl + Shift F10, touto volbou vyvoláme přímo úpravu ozvučení. Rozbalovací seznamy s výběrem hodnot a vyskakovací okna jsou čteny korektně, není zde třeba žádného ozvučení jako například u JAWS. Na příkladu je vidět, že kalendář je možné použít i bez skriptů; jejich vytvoření usnadní práci a vyčítá údaje přesně, jak má. Skript tak uživateli přesně oznámí, na jakém editačním poli se nachází, a on ví, co do daného pole má zadat.

Za nejdůležitější považuji skript pro JAWS, protože bez něj nemůže být okno automaticky vyčteno. Na tomto příkladu je přesně vidět, že se stejnou aplikací každý odečítač pracuje jinak. V první verzi kalendáře bylo třeba použít i výše uvedené skripty pro čtení popisků editačních polí, ve finální verzi však již editační pole jsou díky umístěným popiskům vedle polí ve formuláři čteny všemi odečítači, aniž by bylo třeba provádět skriptování pro tuhle situaci.

4. Přístupnost mobilních operačních systémů

V oblasti mobilních operačních systémů jsou dobře přístupné systémy Symbian a iOS.

4.1. Symbian

Operační systém Symbian je pro nevidomé uživatele zpřístupněn pomocí odečítacího programu MobileSpeak.

Aplikace jako taková se skládá ze dvou částí, samotného hlasového výstupu a odečítače obrazovky. Po nainstalování tohoto SW se mobilní telefony stávají přístupnými. Jsou ozvučeny takřka kompletně, cca 90 % všech původních aplikací operačního systému. Dále jsou ozvučeny různé aplikace, které si uživatelé mohou doinstalovat. Některé funkce mobilních telefonů jsou lépe přístupné z mobilních zařízení než z osobního počítače, například kalendář v Symbianu. Šipkou nahoru a dolů procházíme jednotlivé záznamy a buď je přečteno datum včetně záznamu, nebo jsou záznamy vyčteny po stisku středního tlačítka na telefonu z daného záznamu. Zadávání údajů je tak velmi rychlé a intuitivní. Kvalitně jsou ozvučeny i internetové prohlížeče. Je tak možné se pomocí číslic pohybovat po nadpisech, odkazech či rámcích. Stránka je okamžitě načtena a uživatel ji tak může prohlížet, jako kdyby měl před sebou textový dokument. I zde je ovládání velmi jednoduché, nemění se zde často verze flash přehrávače, takže stránka se odečítači jeví stále stejně. Velmi dobrá je též přístupnost tzv. mobilních stránek. Je zajímavé upozornit, že i některé weby jdou spouštět v mobilní verzi i na osobních počítačích. Mobilní stránka obsahuje většinou textové informace, což je plně vyhovující. Uživatel je tak zbaven obrázků, vyskakovacích oken či jiných grafických prvků. Pohyb v mobilních stránkách je velmi přívětivý, protože uživatele neobtěžují již zmíněné grafické prvky.

Osobně některé weby raději používám v mobilní verzi. To považuji v oblasti přístupnosti jako velké plus. Velmi zajímavá je i přístupnost map. Telefon nám může říkat názvy ulic včetně čísel. Při cestování to může být též výhodou, pokud osoba trasu

nezná. Telefon mu hlásí, kde se aktuálně nachází. Byly snahy i o zpřístupnění navigace pěšky, ale zde někdy GPS poskytne informaci později a dotyčná osoba se včas nedozví, kdy má změnit směr apod. Tahle část se spíše bere jako orientační, není na takové úrovni, aby se nevidomý podle této navigace někam dopravil. Uplatnění vidím v tom, že nevidomý si může vyčíst u zadaného místa, co je kolem něj - od obchodů až po kulturní památky. Kladně hodnotím aplikaci map od provozovatele webu Seznam.cz. Víím, že když jsem na náměstí a spustím aplikaci map, mohu si nechat zobrazit objekty: například kde je banka či daný obchod. Díky tomu má uživatel představu, co se kolem něj nachází. Dále mapy poskytují informace o rychlosti, o směru, kterým jdeme, což může být užitečné. Je zde též možnost čtení knih v textových souborech, takže se zařízení se systémem Symbian stává i čtečkou knih

4.2. IOS

Zde všechna zařízení obsahují odečítací program Voice over. Většinou se nachází v nastavení zpřístupnění. Po aktivaci tohoto odečítače se upraví i práce dotekového ovládání, aby jej bylo možné používat bez zrakové kontroly.

Pomocí dotekových gest je možné používat celý operační systém včetně instalovaných aplikací. Co zatím možné není, je číst knihy v textových souborech txt. Myslím tím čtení knihy, přidávání záložek apod. Voice over používá pro klepnutí či švihnutí dva nebo tři prsty. Pro zadání gest je možné použít různé techniky. Například můžeme klepnout dvěma prsty jedné ruky. Často používané je ovládání pomocí rozděleného klepnutí. To pracuje tak, že položku vybereme prstem - pohybem po displeji na položce, kterou chceme aktivovat, prst podržíme a aktivujeme ji klepnutím dalšího prstu kdekoli na obrazovce.

Švihnutí prstem doleva nebo doprava vybere následující nebo předchozí položku. Švihnutí dvěma prsty nahoru přečte vše od horního okraje obrazovky. Švihnutí dvěma prsty dolů přečte vše od stávajícího místa. Švihnutí třemi prsty nahoru nebo dolů znamená posun o jednu stranu textu. Poklepání aktivuje vybranou položku. Další zajímavostí v dotekovém ovládání je tzv. používání rotoru. Rotor se aktivuje

zakroužením dvěma prsty po obrazovce. Efekt rotoru záleží na tom, co uživatel právě dělá. Může tak nastavit, jestli se chce v textu pohybovat po slovech, řádcích či větách. Na webové stránce můžeme rotorem vybrat, zda se chceme pohybovat po nadpisech, rámcích, seznamech či odkazech.

Pro psaní na dotekovém displeji Voiceover se užívá standardní psaní a dotekové psaní. Při standardním psaní prstem vybereme znak a napíšeme jej poklepáním na obrazovku. Prst je třeba na daném znaku podržet. Metoda dotykového psaní spočívá v doteku na obrazovku, který vybere znak, a po zvednutí prstu se daný znak automaticky napíše.

5. Kalendář

Hlavním cílem aplikace Kalendář (pro nevidomé) je zpřístupnit kalendářové funkce tak, aby byly použitelné s jakýmkoliv odečítačem obrazovky. To přineslo dva problémy, jak danou informaci zobrazit a jak ji odečítačem sdělit. Situace se vyřešila zobrazením okna s danou informací při vyvolání funkce nebo zobrazením editačního pole pro vstup ze strany uživatele. Odečítač tedy jen vyčte obsah okna, které se mu na daný počet milisekund zobrazí, a okno se následně skryje. Tato metoda práce dovoluje uživateli použít kalendář kdekoli v systému, což byl hlavní cíl této aplikace.

Po komunikaci s uživatelem se okno kalendáře skryje a uživatel se ocitne opět tam, kde předtím pracoval. Tato aplikace je použitelná prakticky kdekoli na obrazovce, ať je uživatel v textovém editoru či na webové stránce. To považuji za velkou výhodu. Pokud při psaní textu si chceme například uložit poznámku, vyvoláme funkci poznámek, zadáme potřebné údaje a po stisku klávesy Enter poznámku uložíme a opět se vrátíme do textového editoru. Tento způsob je velmi rychlý a uživatel nemusí spouštět aplikaci nebo v menu hledat danou funkci.

Aplikace je ovládána pomocí klávesových zkratk. Dále lze ovládat aplikaci i z jejího menu, ale to není to tak rychlé; nejdříve musíme do menu vstoupit a šípkami či tabulátorem danou funkci najít. To je i důvod, proč jsem zvolil jako způsob vyčítání informace zobrazení okna na určitý počet milisekund. Hlasový výstup tak stihne danou informaci vyčíst a uživatel je opět v systému, odkud funkci kalendáře vyvolal.

Délku zobrazení okna je možné v nastavení aplikace měnit. Každý má totiž nastavenou jinou rychlost řeči odečítače, a proto potřebuje své individuální nastavení. Stačí jen jednou upravit v nastavení počet milisekund na vyčtená jednoho písmena, hodnota je pak uložena a při dalším spuštění je použita automaticky.

Další problém nastal, jak danou informaci v okně zobrazovat. Jako nejlepší se ukázala možnost v jednom sloupci (tedy v jednom prvku), protože ve více sloupcích (tedy prvcích) nebyly informace všemi odečítači vyčítány. Musí se počítat i s tím, že uživatel použije třeba i několik let starou verzi odečítače.

Zajímavou možností by bylo i přímé volání hlasového výstupu, ale tuto funkci jsem záměrně nepoužil, protože každý nevidomý používá svůj vlastní odečítač, na který je zvyklý. Aplikace je portable, tedy nevíme, s jakým hlasovým výstupem bude program komunikovat. Pokud by se volal přímo hlasový výstup, u některých PC, kde není nainstalován správně hlas či driver odečítače, by mohlo docházet k problémům. Proto je lepší svěřit komunikaci s knihovnamí hlasů odečítači, jak je tomu u standardně používaných aplikací.

Aplikace hodně cílí na začátečníka v práci s počítačem, proto jsem se snažil vyhnout problémům při nastavování voleb. Aplikaci může uživatel použít i na počítači, kde standardně odečítač používán není. Pokud má i portable odečítač obrazovky, například NVDA, může aplikaci použít kdekoli spolu s tímto odečítačem. Uživatel též nemusí řešit problémy administrátorských práv, odečítač i kalendář spustí bez problémů.

Další důvod uvedeného zobrazení informace je rozlišení obrazovky. Odečítače jsou na tuto změnu dost citlivé. Pokud je nevidomý u PC, kde standardně odečítač instalován není, může mít jiné rozlišení, což pozná jedině tak, že mu odečítač nebude správně nebo vůbec vyčítat informace z obrazovky. Pokud je nevidomý administrátorem systému, může si rozlišení nastavit, ale to vždy možné není. Někdy ani není žádoucí, aby si tento uživatel nastavení rozlišení měnil pro potřebu svého odečítače. Jako příklad lze uvést počítač, který je používán pro práci s grafikou jinými uživateli, a ti by tak museli nastavení pracně pro své potřeby vracet zpět. Počítač může být v doméně a uživatel vždy nemusí mít dostatečná oprávnění pro změnu systémových voleb.

Důvodem špatného vyčítání rovněž může být nastavení nějakého motivu či jiných grafických efektů. Tato aplikace bude informace sdělovat i v těchto případech. Další pomocnou volbou v nastavení kalendáře je oznámení tónu před zobrazením okna s informacemi. Po zvukovém signálu je za 1 vteřinu (či po uplynutí dostatečně krátkého intervalu) okno standardně zobrazeno. Tato možnost se hodí tehdy, pokud máme aplikaci stále spuštěnou v systému. Může se nám stát, že píšeme text a kalendář nám zrovna bude chtít sdělit například upozornění z diáře. Standardně tuto situaci poznáme

právě tak, že nám přestane na danou dobu mluvit stisknutá klávesa. Tomu se lze vyhnout nastavením signalizace před zobrazením okna. Po zvukové signalizaci můžeme přestat psát a informace se nám objeví. Po přečtení zprávy a dalším (volitelném) zvukovém signálu může uživatel pokračovat.

5.1. Použité technologie

K sepsání aplikace tohoto typu je potřeba určité kompatibility s některými odečítači. Od toho se odvíjí také použitá technologie. Aplikace musí mít grafické rozhraní, aby měl odečítač co číst. Z tohoto důvodu jsem byl omezen na výběr jazyka, který tvorbu uživatelských rozhraní umožňuje. Rozhodoval jsem se mezi C# a C++. Nakonec byl zvolen C#, protože jeho syntax je konzistentnější vzhledem ke schopnosti nevidomého jeho kód přečíst.

Program je napsán s použitím jazyka C# na platformě NET. Měl by tedy fungovat (případně s nutností rekompilace) na všech systémech, které NET podporují.

K vývoji bylo použito Visual Studio ve verzi 2010.

5.2. Základní programové moduly

Kalendář logicky sestává z několika komponent:

1. Úložiště pro poznámky a události uživatele.
2. Úložiště pro nastavení aplikace.
3. Správa globálních klávesových zkratk.
4. Dynamický načítač svátků z XML souborů.
5. Grafická část aplikace, která je tvořena několika okny a dialogy. Ty slouží k nastavení a interakci jednotlivých komponent.

5.3. Vztah mezi programovými moduly

Hlavním programovým modulem kalendáře je třída `Calendar`. Tato třída obsahuje metody, které reprezentují všechny funkce kalendáře, tedy například adresář, vyskakovací okno, oznamování času nebo také zobrazení hlavního menu. Třída kalendář rovněž obstarává načítání dat z XML souborů a také jejich ukládání. Tato třída je centrálním bodem celé aplikace.

Třída `Calendar` obsahuje instance tříd `FormCalendar` a `FormMenu`, které představují uživatelské rozhraní menu aplikace a vyskakovacího okna. `Calendar` rovněž tvoří wrapper pro adresář, poznámky a uživatelské události. Každý z posledních tří prvků je představován prostým seznamem daných elementů.

Třída `Calendar` obsahuje také instanci třídy `SettingsProvider`, která poskytuje základní rozhraní pro ukládání nastavení aplikace.

5.3.1 Implementace klávesových zkratk

Klávesové zkratky je třeba registrovat přes funkce z Windows API, aby fungovaly v celém OS. Jednu klávesovou zkratku představuje třída `Hotkey`. Registrace každé instance klávesové zkratky probíhá přes funkci `RegisterHotkey` z třídy `GlobalHotkey`, kterou právě Windows API poskytuje. Pro naše použití bylo nutné vytvořit wrapper kolem této funkce.

```
public static bool RegisterGlobalHotkey(Hotkey hotkey) {  
  
    Hotkeys.Add(hotkey);  
  
    return RegisterHotKey(  
  
        hotkey.HWnd,  
  
        hotkey.Id,  
  
        hotkey.Modifier,  

```

```

        (int) hotkey.Key);
    }

```

Třída GlobalHotkey představuje kolekci metod, které pracují s globálními zkratkami, zkratky lze tedy registrovat atp. Lze je přiřadit k nějaké funkci kalendáře při jejich vytváření. O vše se postará konstruktor zkratky.

```

public Hotkey(IntPtr hwnd, int modifier, Keys key, CalFunc
tions function) {
    Hwnd = hwnd;
    Modifier = modifier;
    Key = key;
    Id = GetHashCode();
    Function = function;
}

```

5.3.2. Implementace načítání svátků

Svátky jsou pro jednoduchost ukládány v XML souboru, což plně dostačuje potřebám nevidomého. Každý záznam souboru odpovídá jednomu dni v roce, tedy jednomu svátku. Typický záznam vypadá takto:

```

<nameday date="1-10" type="nameday">Břetislav</nameday>

```

O načítání svátků se stará statická třída NamedayLoader, která je schopna XML soubory načíst a svátky z nich uložit ve formě slovníků, kde klíčem je jméno a hodnotami jsou dny v roce. Načtení svátků spočívá v procházení XML a postupném ukládání dat. Následující část metody LoadNamedays prozradí další podstatné detaily.

```

foreach (XmlNode day in year) {

    string key = day.Attributes["date"].Value;

    if (namedays.ContainsKey(key) == false) {

        CalendarDay my_day = new CalendarDay();

        if (day.Attributes["type"].Value == "nameday") {

            my_day.Nameday = day.InnerText;

        }

        else {

            my_day.Eventday = day.InnerText;

        }

        namedays.Add(key, my_day);

    }

    else {

        if (day.Attributes["type"].Value == "nameday") {

            CalendarDay dayy = namedays[key];

            dayy.Nameday = day.InnerText;

        }

        else {

            CalendarDay dayy = namedays[key];

            dayy.Eventday = day.InnerText;

            namedays[key] = dayy;

        }

    }

}

```

```

        }
    }
}

```

5.3.3. Implementace vyskakovacího okna

Vyskakovací okno je důležitá část, protože je to způsob, jak kalendář oznamuje uživateli nastalou událost (například ho takto upozorňuje na aktuální čas atp). Okno funguje tak, že po jeho zobrazení se zjistí délka zobrazovaného textu, a podle něho se určí, jak dlouho bude okno zobrazeno. Po uplynutí doby zobrazení je okno automaticky skryto, takže uživatel může pokračovat v předchozí práci. Z tohoto pohledu si tedy vystačíme s funkcionalitou na zobrazení okna a funkcionalitou na skrytí okna.

Zobrazení okna obstará metoda Show z třídy FormCalendar.

```

public void Show(int milliseconds) {

    if (bool.Parse(Kalendar.SettingsHolder["start_ring"]
)) {

        new SoundPlayer(@"data\sounds\beep.wav").Play();

    }

    System.Threading.Thread.Sleep(500);

    timerKalendar.Interval = milliseconds;

    timerKalendar.Start();

    lblMain.Focus();

    Show();

    BringToFront();

    Focus();
}

```

```
        Activate();  
    }  
}
```

Jak vidíme, tak metoda zajistí přehrání zvuku před zobrazením okna, pokud to má uživatel povoleno v nastavení aplikace. Následně metoda pozdrží zobrazení o 500 ms a poté okno zobrazí a přesune na něj focus, takže jej odečítač začne číst.

Analogicky funguje metoda na skrytí okna, která se automaticky vyvolá po uplynutí doby stanovené časovačem timerKalendar.

```
private void TimerTick(object sender, EventArgs e) {  
  
    timerKalendar.Stop();  
  
    if (bool.Parse(Kalendar.SettingsHolder["ring"])) {  
  
        new SoundPlayer(@"data\sounds\beep.wav").Play();  
  
    }  
  
    Hide();  
  
    Kalendar.Working = false;  
  
}
```

5.3.4. Implementace adresáře

Adresář je důležitou součástí aplikace. Jedna položka adresáře (tedy jeden kontakt) je představován instancí třídy AddressElement. Třída implementuje potřebné vlastnosti jako jméno kontaktu nebo jeho adresu:

```
public string Name {  
  
    get;
```



```

        set;
    }

    public string Address {
        get;
        set;
    }

    public string Phone {
        get;
        set;
    }
}

```

Adresář nemůže obsahovat dvě položky, které mají stejné jméno a stejnou adresu. Porovnávání prvků adresáře zajistí operátor rovnosti:

```

public static bool operator ==(AddressElement lhs,
AddressElement rhs) {

return rhs.Name == lhs.Name && rhs.Address == lhs.Address;

}

public static bool operator !=(AddressElement lhs,
AddressElement rhs) {

return !(lhs == rhs);

}

```

Obsah adresáře se mezi sezeními aplikace ukládá do XML souboru, k tomu slouží funkce SaveBook():

```
public void SaveBook() {  
  
    string directory =  
System.IO.Path.GetDirectoryName(System.Reflection.Assembly.  
GetExecutingAssembly().Location);  
  
    directory =  
Path.Combine(directory, @"data\save\book.xml");  
  
    XmlWriterSettings settings = new XmlWriterSettings();  
  
    settings.Indent = true;  
  
    settings.NewLineOnAttributes = false;  
  
    XmlWriter writer =  
XmlWriter.Create(directory, settings);  
  
    writer.WriteStartDocument();  
  
    writer.WriteStartElement("contacts");  
  
    foreach (AddressElement add_elem in AddressBook) {  
  
        writer.WriteStartElement("contact");  
  
        writer.WriteAttributeString("name", add_elem.Name);  
  
        writer.WriteAttributeString("email",  
            add_elem.Email);  
  
        writer.WriteAttributeString("icq", add_elem.Icq);  
  
        writer.WriteAttributeString("address",  
            add_elem.Address);  
  
        writer.WriteAttributeString("description",  
            add_elem.Description);  
  
    }  
  
}
```

```

        writer.WriteAttributeString("phone",
                                    add_elem.Phone);

        writer.WriteAttributeString("skype",
                                    add_elem.Skype);

        writer.WriteEndElement();
    }

    writer.WriteEndElement();

    writer.WriteEndDocument();

    writer.Close();
}

```

Analogicky je třeba při spuštění aplikace načítat obsah již uloženého XML souboru:

```

public void LoadBook() {

    AddressBook.Clear();

    string directory =
System.IO.Path.GetDirectoryName(System.Reflection.Assembly.
GetExecutingAssembly().Location);

    directory =
Path.Combine(directory, @"data\save\book.xml");

    if (File.Exists(directory) == false) {

        return;

    }

    XmlReaderSettings settings = new XmlReaderSettings();

```

```

settings.IgnoreComments = true;

settings.ProhibitDtd = false;

settings.IgnoreWhitespace = true;

XmlReader xml =
XmlReader.Create(directory, settings);

while (xml.Read()) {

    if (xml.Name == "contact" && xml.NodeType ==
XmlNodeType.Element) {

        string name = xml.GetAttribute("name");

        string phone = xml.GetAttribute("phone");

        string address = xml.GetAttribute("address");

        string email = xml.GetAttribute("email");

        string skype = xml.GetAttribute("skype");

        string icq = xml.GetAttribute("icq");

        string desc = xml.GetAttribute("description");

        AddAddressElement(name, address, phone, email,
                        skype, icq, desc);

    }

}

xml.Close();
}

```

5.3.5. Nastavení

Nastavení aplikace je realizováno dvouúrovňově:

1. Na nižší úrovni se pracuje se systémovým registrem a funkce na této úrovni odstiňují práci s registrem od zbytku aplikace.
2. Na vyšší úrovni se vyskytuje slovníková struktura, která přiřazuje každému klíči nastavení hodnot právě načítanou a ukládanou z/do registru.

Třída `SettingsProvider` představuje nižší vrstvu, která pracuje s registrem.

```
public class SettingsProvider {  
  
    private readonly string path;  
  
    public string ApplicationName { get; set; }  
  
    public string Path {  
  
        get {  
  
            return path;  
  
        }  
  
    }  
  
    public SettingsProvider(string applicationName,  
                            string applicationPath) {  
  
        if (applicationName.Trim() == string.Empty)  
  
            throw new ArgumentException(  
  
                string.Format(  
                    "Nelze vytvořit nastavení s cestou \"{0}\"",  
                    applicationName), "applicationName");  
  
        ApplicationName = applicationName;  
  
    }  
  
}
```

```

        path = string.Format(@"{0}\{1}",
            applicationPath, applicationName);
    }

    public void WriteSetting(string identifier,
        string value) {
        Registry.SetValue(Path, identifier, value);
    }

    public string ReadSettings(string identifier,
        string defaultValue) {
        string value =
            (string)Registry.GetValue(Path,
                identifier, defaultValue);

        if (value == null) {
            return defaultValue;
        }

        else {
            return value;
        }
    }
}

```

Metoda LoadSettings() je součástí třídy Calendar, tato metoda načítá nastavení pomocí metod třídy SettingsProvider do slovníku. Aplikace následně pracuje s daty v tomto slovníku tak, aby program mohl neustále nepřístupoval k registru.

```

public void LoadSettings() {

    SettingsHolder.Add("ring",
        SettingsBackend.ReadSettings("ring",
            false.ToString()));

    SettingsHolder.Add("lang",
        SettingsBackend.ReadSettings("lang", "cs-CZ"));

    SettingsHolder.Add("speed",
        SettingsBackend.ReadSettings("speed", "100"));

    SettingsHolder.Add("start_ring",
        SettingsBackend.ReadSettings("start_ring",
            false.ToString()));

    SettingsHolder.Add("time_interval",
        SettingsBackend.ReadSettings("time_interval", "5"));

    SettingsHolder.Add("time_interval_enabled",
        SettingsBackend.ReadSettings("time_interval_enabled",
            false.ToString()));

}

```

5.4. Ovládání a rozhraní kalendáře

Aplikace Kalendář se ovládá, jak již bylo uvedeno, pomocí klávesových zkratk. Pro klávesové zkratky byla použita kombinace kláves Ctrl + funkční klávesy F1 až F12 (mimo klávesové zkratky F4, která nebyla naprogramována).

5.4.1. Seznam klávesových zkratek a funkcí

1. Ctrl + F1: přidání poznámky
2. Ctrl + F2: přidání upozornění do diáře
3. Ctrl + F3: výpočet dnů do data
4. Ctrl + F5: zobrazení zadaných a uložených poznámek uživatelem
5. Ctrl + F6: zobrazení zadaných a uložených upozornění v diáři uživatelem
6. Ctrl + F7: oznámení svátků k aktuálnímu dni
7. Ctrl + F8: zobrazení nejbližšího upozornění v diáři
8. Ctrl + F9: adresář
9. Ctrl + F10: vyvolání hlavního menu
10. Ctrl + F11: kdo má dnes svátek
11. Ctrl + F12 : den v týdnu

Nastavení kalendáře obsahuje následující položky:

1. Možnost volby českých či slovenských svátků.
2. Volba oznamování času v pravidelném intervalu.
3. Počet milisekund na jedno písmeno.
4. Možnost nastavení zvukové signalizace před a po zobrazení okna aplikace.
5. Spouštění aplikace při startu operačního systému.

V rozhraní aplikace nebyly použity žádné grafické prvky. Aplikace běží na systému Windows od verze XP výše. Pracuje jak v 32, tak 64 bytových systémech. Je nutné, aby v počítači bylo nainstalováno rozhraní Net Framework. Též musí být nainstalován odečítač obrazovky. Nejlepší podporu má odečítač NVDA. Kompletní ovládání a použití je popsáno v nápovědě, která též slouží jako uživatelská příručka. Tato příručka se nachází v hlavním adresáři kalendáře, a sice ve formátu txt, což je pro nevidomého uživatele dostačující.

5.5. Uživatelská příručka

5.5.1. Hlavní menu

Hlavní menu obsahuje jak funkce kalendáře, tak i možnost skrytí menu do systémové lišty. Zrušení právě nabízené volby lze realizovat klávesou Esc či případně klepnutím na tlačítko Zrušit. V menu kalendáře se lze pohybovat klávesou Tab, mezerníkem či klávesou Enter a tím danou volbu aktivovat.

Hlavní menu kalendáře obsahuje následující položky:

1. Moje upozornění - zde je možné uložit upozornění, která nám mají být připomenuta v zadaný den a čas. Nejbližší upozornění je při spuštění kalendáře oznamováno každé dvě hodiny. Je to proto, že uživatel není stále u počítače a pokud přijde po hodině, tak se nemusí bát, že mu událost nebude oznámena. Dvě hodiny zároveň nejsou tak malý interval, že by upozornění působilo rušivě. Upozornění lze zobrazit v řádkovém seznamu, kde se uživatel dozví datum, čas a text upozornění. Zadaná upozornění se nachází v podadresáři data/save, v souboru events.xml.
2. Přidání poznámky - poznámkou je myšlen napsaný text, který si může uživatel nechat kdykoli pohotově zobrazit. Hodí se pro okamžité uložení údajů, které budeme časem zpracovávat a potřebovat (například telefonní čísla či jiný libovolný

text). Není zde nastaven limit vkládaného textu, tak je možné vkládat i text e-mailů, ze kterého budeme časem potřebovat nějakou informaci. Data poznámek jsou uložena v podadresáři data/save, v souboru notes.xml.

3. Počet dnů do data - zde je možné vložit datum, aby nám byla následně sdělena informace o tom, kolik dní do zadaného data zbývá. Nelze vkládat datum, které již uplynulo. Tuto funkci můžeme využít v případě, že dostaneme úkol či termín k určitému datu a můžeme si tak nechat spočítat, kolik dní na vyřešení úkolu máme.
4. Minutník - je možné nastavit výběrem kurzorovými šipkami, za jakou dobu nám má být přehrán zvukový tón. Údaj se nastavuje v 30 vteřinových intervalech. Můžeme nastavit jak 3,5 minuty, ale také 2 hodiny a 32 minut. Při nastavování je oznamován počet vteřin.
5. Nastavení – obsahuje nastavení kalendáře.
6. Konec - ukončí kalendář.
7. Skryt menu - tím se kalendář přemístí do systémové lišty a běží na pozadí. Pak můžeme pracovat v jiné aplikaci a budeme mít dostupné pod kurzorovými šipkami jen okno dané aplikace.
8. Adresář - po otevření tohoto podmenu se objeví přidání, úprava, odstranění a zobrazení adresy. Uživatel může zadat adresu, telefon a tak mít v kalendáři i jednoduchý adresář. Po kontaktech se uživatel pohybuje kurzorovými šipkami, odečítač čte jméno a příjmení. Pro zobrazení celé adresy, je třeba kliknout na tlačítko zobrazit. Je to proto, že ne všechny odečítače by vyčetly celý text a uživatel se může po jednotlivých editačních polích po stisku volby Zobrazit pohybovat klávesou Tab, a tak procházet informace i kurzorovými šipkami, obsah kopírovat do schránky a dále s ním pracovat.
9. Svátky dle jména - uživatel je vyzván napsat jméno, po jeho napsání a stisku klávesy Enter je vyčteno datum svátku zadaného jména.
10. Informace o datu ke dni - po napsání datumu v této funkci je uživateli sdělen název dne zadaného data.

5.5.2 Nastavení

V nastavení najdeme následující možnosti:

1. Spustit při startu systému - po zatrhnutí tohoto tlačítka mezerníkem dojde ke spuštění kalendáře při startu Windows.
2. Volba jazyka, pro který mají být zobrazovány svátky - touto volbou lze nastavit oznamování českých či slovenských svátků. Danou možnost vybereme šipkami nahoru a dolů.
3. Počet milisekund na jedno písmeno - zde individuálně nastavujeme, jak dlouho se má zobrazit okno s danou informací, aby jej odečítač vyčetl z obrazovky. Doporučení je od 180 ms pro hlasy SAPI 5 a 70 ms pro eSpeak či hlasový výstup Hlas. Záleží na rychlosti hlasu uživatele.
4. Možnost nastavení pravidelného oznamování času - zde můžeme nastavovat čas, aby nám byl sdělován například každých 10 minut. Počet minut od minimálních 5 můžeme opět nastavit šipkami nahoru a dolů.
5. Tón před a po přečtení textu - těmito tóny jsme upozorněni před zobrazením informace a také po jejím ukončení, rovněž skrytím okna s danou informací.
6. Tlačítkem OK nastavení uložíme a aktivujeme.

Závěr

Ve své bakalářské práci jsem ukázal, jaké jsou možnosti přístupnosti počítačových technologií pro nevidomé uživatele, a to jak v oblasti hardwaru, tak speciálního softwaru. Přístupností webu se zabývá i Google, který pokud pozná na své stránce využití asistivní technologie, nabídne uživateli například přepnutí webové stránky Gmailu do základního HTML. Jako další úpravu nabízí vyhledávač, který nabídne při aktivním odečítači vypnutí dynamického vyhledávání.

Zjistil jsem, že Google podporuje nejvíce odečítací program JAWS. Google Kalendář nabízí ovládání pomocí klávesových zkratk, tak jej i nevidomý může celkem pohodlně ovládat i ve webovém rozhraní. Podporu pro asistenční technologie nabízí i balík LibreOffice, který při instalaci nabízí podporu screen readerů. Svoji aplikací Kalendář jsem chtěl ukázat, že je dobré programovat i aplikace, které jsou specializované pro danou skupinu uživatelů. Tito uživatelé mají jistotu, že SW jim bude pracovat vždy a nebude záležet na tom, jaký odečítač budou pro svou práci používat. Tyto specializované aplikace mají své specifické ovládání a pro uživatele se práce stává rychlejší a intuitivnější. Do aplikace jsem integroval funkce, které se nejčastěji využijí, tedy funkce pro každodenní potřebu. Právě o tyto funkce jsem zaznamenal zájem, protože nebyly jednoduché a pro všechny v dostupné formě. Jednoduchost dostala přednost před počtem funkcí.

Slovník pojmů

Odečítač obrazovky

SW zajišťující ozvučení počítače či jiného zařízení s operačním systémem.

Hlasový výstup

Konkrétní hlas, který je odečítačem využit pro výčet informací z obrazovky.

Manager slovníků

Seznam pravidel výslovnosti určitých slov, které jsou v hlase špatně definovány.

Sam

Rozhraní obsahující ovladače pro Braillovské řádky a klávesnice, zajišťující komunikaci mezi odečítačem a daným zařízením.

Metodika webu

Soubor důležitých pravidel pro zpřístupnění webové stránky.

Script

Soubor obsahující definice či posloupnost příkazu spuštění jiných skriptů, také akcí odečítače pro daný program či pro situaci v aplikaci.

Braillovský řádek

Zařízení zobrazující text na řádku ve slepeckém písmu.

Braillovská tiskárna

Zařízení, které tiskne na speciální papír text či grafy v bodovém (slepeckém) písmu.

Braillovská klávesnice

Klávesnice umožňující vstup uživatele pomocí vkládání znaků Braillovské abecedy.

6 a 8 bod

Dvě metody čtení a vkládání Braillova (bodového) písma. 6 bod tvoří klasické znaky abecedy. 8 bod umožňuje reprezentaci speciálních znaků ASCII či matematických výrazů.

Asistent, WinMenu, Guide

Balík aplikací zjednodušující práci v jednotlivých aplikacích, např. e-mail, skenování dokumentů, zpracovávání a tisk dokumentů.

Talkman

Aplikace sloužící k pohodlnému čtení dlouhých textů či knih.

Blind friendly

Web seznam metodik pro zpřístupnění webu.

Asistivní technologie

Mezi tyto technologie můžeme zařadit jak odečítač obrazovky, tak SW pro úpravy aplikací či weby pro nevidomé uživatele pro lepší funkčnost odečítače

FineReader SW

Slouží pro snímání dokumentu ze skeneru a jeho zpracování do textové podoby.

Podmenu

Podnabídka hlavního menu, tedy „zanořené“ menu.

Karta

Seznam prvků tlačítek a seznamů.

Zaškrťovací tlačítko

Tlačítko, které lze zatrnout mezerníkem, aby se daná volba aplikovala.

Klávesová zkratka

Kombinace kláves vyvolávající např. spuštění aplikace či provedení dané akce v aplikaci.

Zvuková myš

Simulace myši pomocí virtuálního kurzoru. Editační pole, řádek, do kterého je možné vkládat text.

Seznam

Seznam prvků, které jsou pod sebou a jdou zvolit pomocí kurzorových šipek, např. volba druhu písma.

Virtuální kurzor

Kurzor, se kterým lze pohybovat po celé obrazovce i tam, kde se kurzorovými šípkami není možné dostat. Definice virtuálního kurzoru mohou být obsaženy ve skriptu.

Literatura

NVDA

www.NVDA-project.org

WinMonitor

www.acedesign.cz

Dodatavek JAWS a WinMenu

www.galop.cz

Výrobce JAWS

www.freedomscientific.com

VoiceOver (Mac OS X)

www.apple.com

Orca (Linux)

live.gnome.org/Orca

Stránky o přístupnosti Linuxu a použití odečítačů

www.skolanet.cz/downloads.php?cat_id=7&rowstart=15

Distribuce Vinux

vinuxproject.org

eSpeak (syntetizér)

eSpeak.sourceforge.net

Wintalker

www.rosasoft.cz

Výstup Eliška

www.acapela.com

O přístupnosti webových stránek

blindfriendly.cz

Blog Radka Pavlíčka z brněnského Tyflocentra, který se zabývá dostupností výpočetní techniky pro nevidomé

www.poslepu.blogspot.com

Sam (sada balíčků pro Braillové písmo)

<http://www.yourdolphin.com/>

Microsoft Active Accessibility

[http://msdn.microsoft.com/en-us/library/windows/desktop/dd373592\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/dd373592(v=vs.85).aspx)

Portál o použití OS Android pro nevidomé

www.blind-android.cz