



**FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI**

**KATEDRA
KYBERNETIKY**

Bakalářská práce

Hlasový terminál pro jízdní řády městské hromadné dopravy

Jan Šimek



FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI

KATEDRA
KYBERNETIKY

Bakalářská práce

Hlasový terminál pro jízdní řády městské hromadné dopravy

Jan Šimek

Vedoucí práce

Ing. Luboš Šmídl, Ph.D.

© Jan Šimek, 2023.

Všechna práva vyhrazena. Žádná část tohoto dokumentu nesmí být reprodukována ani rozšiřována jakoukoli formou, elektronicky či mechanicky, fotokopírováním, nahráváním nebo jiným způsobem, nebo uložena v systému pro ukládání a vyhledávání informací bez písemného souhlasu držitelů autorských práv.

Citace v seznamu literatury:

ŠIMEK, Jan. *Hlasový terminál pro jízdni řády městské hromadné dopravy*. Plzeň, 2023. Bakalářská práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra kybernetiky. Vedoucí práce Ing. Luboš Šmídl, Ph.D.

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd
Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Jan ŠIMEK**
Osobní číslo: **A19B0387P**
Studijní program: **B0714A150005 Kybernetika a řídicí technika**
Specializace: **Umělá inteligence a automatizace**
Téma práce: **Hlasový terminál pro jízdní řády městské hromadné dopravy**
Zadávací katedra: **Katedra kybernetiky**

Zásady pro vypracování

- i) Prostudujte problematiku komunikace s počítačem mluvenou řečí a hlasových dialogových systémů, seznamte se s hlasovou platformou SpeechCloud.
- ii) Navrhněte multimodální hlasový dialog pro odjezdy vlaků či spojů městské dopravy, který poskytne základní informace cestujícímu – čas odjezdu a případné informace o přestupu.
- iii) Proveďte rešerši dostupných datových zdrojů poskytujících informace o dopravních spojeních.
- iv) Dialog realizujte a otestujte.

Rozsah bakalářské práce: **30-40 stránek A4**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

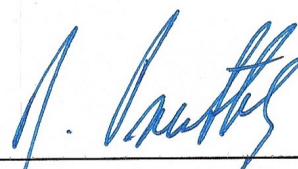
Dodá vedoucí práce

Vedoucí bakalářské práce: **Ing. Luboš Šmídl, Ph.D.**
Katedra kybernetiky

Datum zadání bakalářské práce: **17. října 2022**
Termín odevzdání bakalářské práce: **22. května 2023**



Doc. Ing. Miloš Železný, Ph.D.
děkan



Prof. Ing. Josef Psutka, CSc.
vedoucí katedry

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného akademického titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Západočeská univerzita v Plzni má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V Blovicích dne 21. května 2023

.....
Jan Šimek

V textu jsou použity názvy produktů, technologií, služeb, aplikací, společností apod., které mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Abstrakt

Dokument představuje bakalářskou práci, jejímž cílem je navrhnout hlasový dialogový systém, jenž uživatelům poskytne informace o časech odjezdů spojů městské hromadné dopravy a vyhledá vhodné spojení pro konkrétní zastávku. Dialogový systém je postavený na platformě SpeechCloud, která je vyvíjena na Fakultě aplikovaných věd Západočeské univerzity v Plzni ve spolupráci s její spin-off firmou Spechtech, s.r.o. a zatím je lokalizován převážně pro prostředí města Plzně.

V práci je kladen důraz na návrh a základní funkcionalitu, kterou by měl dialogový systém disponovat. Základní dialogový systém však bude realizován i prakticky – pro demonstraci, co a jak je možné v dialogovém systému technicky implementovat. Vizí do budoucnosti je pak zabývat se detaily uživatelského rozhraní a funkcemi dialogového systému tak, aby jej mohla využít široká veřejnost.

Abstract

The document presents a bachelor thesis, which aims to design a voice dialogue system that provides information about the departure times of public transport connections and searches for a suitable connection for a specific stop to users. The dialogue system is based on the SpeechCloud platform, which is being developed at the Faculty of Applied Sciences of the University of West Bohemia in Pilsen in cooperation with its spin-off company Spechtech, s.r.o., and so far it has been localized mainly for the environment of the city of Pilsen.

The emphasis in the work is on design and basic functionality that the dialogue system should have. However, the basic dialogue system will also be implemented practically – to demonstrate what and how can be technically implemented in the dialogue system. The vision for the future is then to deal with the details of the user interface and advanced features of the dialogue system so that it can be used by the general public.

Klíčová slova

Hlasový dialogový systém • městská hromadná doprava • jízdní řády • rozpoznávání řeči • syntéza řeči

Poděkování

Na tomto místě bych rád poděkoval svému vedoucímu bakalářské práce, panu Ing. Luboši Šmídlovi, Ph.D. za umožnění práce na vlastním zadání, za pomoc při seznamování se s platformu SpeechCloud, za trpělivost, cenné rady, logicky strukturovanou supervizi mé práce i vzbuzení zájmu v tomto výzkumném odvětví.

Obsah

1	Úvod	3
2	Hlasová komunikace a řešení úloh	5
2.1	Komunikace lidí v přirozeném jazyce	5
2.2	Řešení úloh s využitím výpočetní techniky	5
2.3	Proč chtít mluvit s počítačem	6
2.3.1	Přínos pro osoby zrakově postižené	7
2.4	Hlasový dialogový systém	7
2.4.1	Úskalí rozpoznávání a porozumění řeči	8
2.4.2	Historie a úskalí syntézy řeči	9
2.4.3	Řízení dialogu	9
2.4.4	Výhody multimodálního dialogu	10
3	Městská hromadná doprava v Plzni	11
3.1	Informační zdroje pro širokou veřejnost	12
3.1.1	Zastávkové jízdní řády	12
3.1.2	Elektronické informační tabule	13
3.1.3	Web PMDP	14
3.1.4	Aplikace Moje PMDP	15
3.1.5	Další komerčně dostupné informační zdroje	16
3.2	Otevřená data pro vývojáře	17
3.2.1	Plzeň a Plzeňský kraj	17
3.2.2	Praha a Středočeský kraj jako vzor	18
3.2.3	Situace na celostátní úrovni	18
4	Vývoj hlasového dialogu	19
4.1	Návrh hlasového dialogu	19
4.1.1	Vývojový diagram z pohledu uživatele	19
4.1.2	Rozpoznávání řeči	19
4.1.3	Syntéza řeči	21
4.1.4	Funkce související s jízdními řády	21

4.2	Představení frameworku SpeechCloud	22
4.3	Vývoj backendu	23
4.3.1	Výchozí struktura skriptu	23
4.3.2	Vlastní funkce pro konkrétní úlohu	24
4.3.3	Gramatiky	25
4.4	Vývoj frontendu	26
4.4.1	Struktura HTML stránky	26
4.4.2	Struktura JavaScriptu	28
5	Výsledná aplikace	29
5.1	Uživatelské rozhraní	29
5.2	Příklady použití	30
5.3	Testování na uživateli	32
5.4	Uvažovaná vylepšení do budoucna	33
5.4.1	Robustnější vyhledávač spojení	33
5.4.2	Vylepšení gramatik	34
5.4.3	Vylepšení struktury dialogu	34
5.4.4	Rozdělení na zastávkovou a počítačovou (mobilní) část	35
5.4.5	Nápověda a návod k použití	35
5.4.6	Alternativní možnosti zahájení dialogu	35
5.4.7	Rozšíření funkcionality	35
5.4.8	Vlastní hlas pro syntézu řeči	36
6	Závěr	37
A	Přílohy	39
A.1	Formulář pro uživatelské testování	39
	Bibliografie	43
	Seznam obrázků	45

V rámci této práce jsem se rozhodl navrhnout a implementovat **multimodální hlasový dialog**, pomocí kterého bude moci uživatel vyhledávat **informace o spojeních a odjezdech** spojů v rámci městské hromadné dopravy Plzni. K tomu mě vedly následující skutečnosti.

Jsem velkým fanouškem městské hromadné dopravy v Plzni, a hromadné dopravy obecně. Mimo účel transportu mě baví vyhledávat i netradiční či nepřímocará spojení na různá místa, porovnávat výsledky vyhledávání mezi různými aplikacemi a předávat své vlastní zkušenosti dalším osobám. Tato práce je pro mě cestou, jak se podílet na rozvoji a popularizaci tohoto odvětví.

Mimo umělou inteligenci jsem našel zalíbení i v českém jazyce. Během svých studií na gymnáziu jsem se pravidelně účastnil olympiád z češtiny, a to se slušnými výsledky z krajských i celostátních kol. Jakožto hrdý Čech beru češtinu jako součást našeho kulturního dědictví a – i když nejsem odpůrce neologismů přejímaných hlavně z angličtiny – myslím, že pro udržení českého jazyka naživu je nezbytné, aby si udržel pozici i na poli nových technologií, kde je ve stále větší míře nahrazován hlavně angličtinou, zejména kvůli úspoře financí zahraničních vývojářů.

Nejsem si vědom toho, že by v České republice v současné době fungovala komerčně dostupná aplikace podobná té, jež je výsledkem této práce. Myslím, že aplikace by se v případě vyladění technických nedostatků a zpřístupnění veřejnosti stala přínosem pro velkou část cestujících.

Hlasové vyhledávání spojení by využili hlavně cestující, kteří se v plzeňské MHD neorientují dobře – např. turisté nebo ti, kteří opakovaně cestují pouze na několik málo míst. Oproti využití konvenční mobilní aplikace by jim to mohlo ušetřit i několik cenných minut, takže by např. nemuseli promarnit spoj, který by jim kvůli zdlouhavému vyhledávání v aplikaci ujel. Hlasové ovládání terminálu je na místě, neboť v případě umístění na zastávce není integrace klávesnice vhodná a použití dotykové klávesnice by nebylo pohodlné.

Obrovským benefitem by se aplikace stala pro nevidomé, kteří by tak nebyli odkázáni pouze na čtečky nebo rady kolemjdoucích. V současné době je předpokládáno, že aplikace bude umístěna přímo na některé ze zastávek MHD, do budoucna se však uvažuje o tom přenést hlasový dialog i do mobilní aplikace tak, aby jej mohli mít nevidomí (i další lidé) neustále u sebe.

Kromě samotného vývoje aplikace bude úkolem opatřit si pro fungování aplikace potřebná data – zejména jízdní řády a prostředky pro vyhledávání spojení. V této práci nebudeme řešit algoritmus vlastního vyhledávače spojení, ale spokojíme se s veřejně dostupnými službami, ze kterých budeme data čerpat.

Během seznamování se s platformou SpeechCloud neočekávám žádné zásadnější problémy, neboť moduly pro rozpoznávání a syntézu řeči jsou již v rámci frameworku vyřešené a základní funkce frameworku nejsou složité. Navíc je tato platforma využívána v rámci výzkumu na Katedře kybernetiky, kde studuji a tvořím tuto práci, takže předpokládám, že nebude problém se v případě potíží obrátit na svého vedoucího práce, případně jiného vědeckého pracovníka.

Hlasová komunikace a řešení úloh

2

2.1 Komunikace lidí v přirozeném jazyce

Hlasová komunikace v přirozeném jazyce je tím nejpřirozenějším a zpravidla **nejrychlejším způsobem komunikace** mezi dvěma a lidmi nebo mezi člověkem a masou lidí. Svůj mateřský jazyk máme možnost vnímat od narození a první jednoduché věty začínáme zpravidla skládat již kolem třetího roku života. Existují dvě základní teorie snažících se vznik jazykové vybavenosti v takto raném věku – nativistická a empirická [RS90], ty zde však nebudou rozebírány. Přestože můžeme být vystaveni vlivům různých jazykových mutací, tzv. nářečí, která jsou v češtině dána zejména regionálně, vzdělávací systém naší jazykovou výbavu formuje podle jasně daných pravidel tak, abychom byli schopni se bez nedorozumění mluvenou či písemnou formou domluvit s jakýmkoliv jiným člověkem hovořícím stejným jazykem, v našem případě češtinou.

2.2 Řešení úloh s využitím výpočetní techniky

Jedním z prvních komerčně dostupných zařízení výpočetní techniky se stal **osobní počítač**. Ten byl obecně navržen k tomu, aby člověku pomáhal řešit různé typy úloh, nebo některé rutinní dokonce dělal za něj. Při stejném výpočetním výkonu tato zařízení postupem času a vývoje zabírala menší a menší prostor, dnes v kapse u kalhot běžně nosíme chytrý telefon, který je schopen výpočetně zajistit řešení velké části úloh, které jsme před jejich příchodem mohli řešit výhradně na osobním počítači.

Aby však byl stroj schopen řešit a řešit úlohu, kterou od něj očekáváme, musíme stroji poskytnout **správné zadání**. Většina počítačově gramotné společnosti si to již ani neuvědomuje, ale v momentě, kdy zadáváme počítači konkrétní úlohu, využíváme k tomu většinou rozhraní konkrétního programu, který je schopen úlohu řešit. **Uživatelské rozhraní** (běžně se používá zkratka UI, z anglického user inter-

face), většinou grafické, tvoří vývojář daného programu tak, aby bylo pro člověka **co nejintuitivnější** a snadno pochopitelné (s případným poskytnutím nápovědy nebo manuálu), a zároveň pokrylo všechny funkce, které program poskytuje. Pokud chce uživatel danou úlohu řešit pomocí tohoto programu, musí se naučit uživatelské rozhraní používat.

Přestože naučit se pracovat s uživatelským rozhráním není pro většinu počítačově gramotných lidí takový problém, je třeba mu věnovat nějaký čas. Navíc, práce s ním může být někdy zbytečně zdlouhavá. Pojďme si ukázat modelovou situaci, kdy chceme na počítači řešit úlohu odebrání pozadí z portrétové fotografie. Zevrubný algoritmus této úlohy můžeme z našeho pohledu sepsat do následujících bodů (nyní neuvažujeme využití umělé inteligence, i když bychom mohli):

1. vybrat vhodný software pro řešení této úlohy – grafický editor,
2. načíst fotografii do grafického editoru,
3. vybrat vhodný nástroj pro výběr oblasti,
4. označit oblast fotografie s postavou,
5. odstranit vše mimo vybranou oblast.

Hranice mezi zadáním úlohy a jejím řešením je zde velmi tenká. Někdo by mohl namítat, že úlohu na počítači přímo řešíme, nicméně anotaci a změnu barvy (resp. alfa kanálu) jednotlivých pixelů za nás v tomto případě provádí počítač, zastávám tedy názor, že se stále jedná spíše o zadání, přestože o tom můžeme vést spory. Tak či onak trvá tato činnost několik minut.

2.3 Proč chtít mluvit s počítačem

Pokud bychom chtěli úlohu z předchozí kapitoly zadat jinému člověku, pravděpodobně by nám k zadání stačila jedna věta, např. *Odstraň pozadí této fotky a zachovej jen postavu.* Chtěli-li bychom být konkrétnější: *Učiň na fotografii veškeré pixely oblasti mimo postavy plně průhledné.* V případě, že je nám jedno, jakým způsobem bude onen člověk úlohu řešit (jde nám hlavně o výsledek), úloha je zadána a samotná promluva trvala jen několik sekund, což je oproti předchozímu zadávání úlohy stroji znatelná časová úspora.

Zadáme-li však tuto úlohu jinému člověku, moc času neušetříme, protože onomu člověku bude řešení úlohy na počítači trvat pravděpodobně stejně dlouho, jako by to trvalo nám. Nejlepším řešením by bylo zadat úlohu v přirozeném jazyce stojí, který by ji pochopil a úlohu pak sám vyřešil během několika málo sekund.

2.3.1 Přínos pro osoby zrakově postižené

Jak již bylo naznačeno v kapitole 2.2, běžné úlohy na počítači řešíme téměř výhradně skrz grafické (příp. textové) uživatelské rozhraní operačního systému a jednotlivých programů, k čemuž jako kontroléry používáme např. klávesnici, myš, touchpad nebo dotykovou obrazovku. Jako zpětná vazba nám pak slouží náš zrak, protože na monitoru počítače (nebo telefonu) vidíme odezvu kontrolérů (polohu kurzoru, napsaný text, reakce UI).

Na světě však žije téměř 40 milionů nevidomých lidí [Pet23], jejichž možnosti práce s výpočetní technikou jsou značně omezené¹. Tito lidé se často při práci na počítači neobjedou bez speciálních **kompensačních pomůcek**, jakými jsou např. braillový displej nebo různé asistivní čtečky [Pav23]. Pokud by byli s počítačem schopni plně komunikovat v přirozeném jazyce, pravděpodobně by dalších kompenzačních pomůckách nebyli tolik závislí.

2.4 Hlasový dialogový systém

Hlasový dialogový systém (dále jen HDS) se skládá z uživatele (člověk), hlasového agenta (stroje, který je schopen komunikovat v přirozeném jazyce) a modelu úlohy (po zadání úlohy od hlasového agenta je od něj očekáván výsledek). V této práci předpokládáme, že model úlohy je daný (konkrétně se jedná o vyhledávač v jízdních řádech – rozvedeme později) a naším úkolem je především **navrhnout hlasového agenta** tak, aby pro uživatele byla komunikace s ním co nejpřirozenější a zároveň hlasový agent správně chápal, co po něm člověk chce.

Hlasový agent je sám o sobě velmi komplexní systém, který musí pro celistvou funkčnost implementovat funkce všech následujících modulů [Psu+06]:

- rozpoznávání řeči,
- porozumění řeči,
- řízení dialogu,
- generování odpovědi,
- syntéza řeči (z textu).

Toky informací mezi těmito moduly jsou znázorněny na obrázku 2.1.

¹Nutno podotknout, že velká část této populace žije v rozvojových zemích, kde často není dostupná základní zdravotní péče, ale ani výpočetní technika.



Obrázek 2.1: Schéma hlasového dialogového systému, Zdroj: [Šve23] (upraveno)

Komunikace se strojem v přirozeném jazyce je v mnoha ohledech problematická úloha, protože pro správnou funkci každého z modulů je potřeba počítat s velkou mírou stochastičnosti a nepředvídatelného chování, zejména pak ze strany uživatele. V následujících podkapitolách jsou velmi stručně popsány problémy, se kterými se návrháři HDS musejí vypořádat.

2.4.1 Úskalí rozpoznávání a porozumění řeči

Výpočetní technika sama o sobě nedokáže přirozenému jazyku porozumět, vyskytuje se v něm totiž velké množství proměnných a stochastičnosti. Na problém narazíme už během samotného rozpoznávání řeči. Zvuk je v počítači reprezentován jako vektor intenzity mechanického vlnění [Ouj23]. I kdybychom měli stoprocentně přesný mikrofón, stejný řečník nikdy nepronese stejnou promluvu tak, aby si její vektorové reprezentace vzájemně odpovídaly. To se týká nejen celých vět či slov, ale i samotných hlásek – fonémů.

Další složitou úlohou je porozumění tomu, co člověk řekl. V minulosti se k **sémantické analýze** textu přistupovalo hlavně z matematicko-statistického hlediska s přispěním metod strojového učení. V poslední době na tomto poli převažují neuronové sítě [SK17]. Milníkem v této oblasti se v roce 2022 stal komerčně dostupný jazykový model GPT vyvíjený společností OpenAI², který byl trénován jazykově nezávisle [Zhe+23]. Sám jsem si mohl vyzkoušet, že je porozumění jazykového modelu, a to i českému textu, pozoruhodně dobré.

²Více informací o jazykovém modelu GPT-4 na www.openai.com/product/gpt-4

2.4.2 Historie a úskalí syntézy řeči

Problém je i se strojovou syntézou řeči. U komplexního dialogového systému nelze např. sestavit databázi nahrávek jednotlivých slov tak, jako je tomu např. v případě nádražního rozhlasu, protože neexistuje možnost, jak postihnout všechna slova ve všech tvarech, která lze v češtině vyslovit. I kdyby to šlo, databáze by byla neuvěřitelně velká, prohledávání by trvalo dlouho a syntéza by tak byla neefektivní.

K syntéze řeči se v minulosti přistupovalo různými způsoby. První pokusy o **mechanické syntetizéry** se objevily již na konci 18. století, ty však lidskou řeč připomínaly jen velmi vzdáleně [Psu+06]. **Elektronické syntetizéry** se začaly vyvíjet na začátku 20. let 20. stol, první přesvědčivé výsledky se však začaly objevovat až o zhruba 20 let později, řeč však stále zněla příliš strojově [Psu+06].

V 60. letech se s nástupem číslicových počítačů objevily nové možnosti, jak syntézu řešit. Až do konce 80. let převládá tzv. **formantová syntéza** založená na teorii zdroje a filtru. Současně se rozvíjí druh syntézy, která převládala až do druhé dekády 21. století, a to je tzv. **konkatenační syntéza**. Ta je založena na principu řetězení vhodných řečových jednotek za sebe, přičemž řečové jednotky lze uchovávat jako nahrávky přímo od řečníka nebo je možné je generovat pomocí skrytých Markovových modelů [Psu+06]. V současné době však i na tomto poli převládají **neuronové sítě**, které mají do budoucna velký potenciál vzhledem k tomu, že tradiční metody syntézy řeči pravděpodobně již dosáhly svých limitů [HVT18].

2.4.3 Řízení dialogu

Řídicí modul je pomyslným mozkiem celého hlasového agenta. Jeho hlavním úkolem je sledovat stav uživatele prostřednictvím významového popisu řeči, na jehož základě se snaží změnit stav modelu úlohy podle požadavku uživatele. Výsledek úlohy pak musí v přirozeném jazyce interpretovat tak, aby mohla být jeho odpověď předána modulu syntézy a sdělena uživateli. Tyto informační vazby jsou jasně znázorněny na obrázku 2.1. Přitom je důležité si uvědomit, že dialog se obvykle neskládá pouze z jedné otázky a odpovědi, nýbrž může být dále rozvíjen. Řídicí modul by měl tyto změny stavů vnímat a za základě těchto změn ovlivňovat svůj stav tak, aby řešil, co mu skutečně bylo zadáno.

Právě návrh řízení bude hlavním cílem této práce. Celá aplikace dialogového systému bude postavena na frameworku **SpeechCloud**, který byl vyvinut Katedrou kybernetiky Fakulty aplikovaných věd Západočeské univerzity v Plzni a její spin-off firmou SpeechTech, s.r.o. Obrovskou výhodou je, že framework již plně implementuje funkce rozpoznávání a syntézy řeči. Při tvorbě aplikace se tedy budeme moci soustředit na samotný návrh dialogu, který zároveň budeme schopni hned otestovat.

2.4.4 Výhody multimodálního dialogu

Multimodální dialogový systém je navržen tak, aby s uživatelem komunikoval pomocí řeči **v kombinaci s dalšími informačními kanály**, typicky hlavně s grafickým. Pokud takový multimodální dialogový systém běží např. na osobním počítači, výhoda tkví v tom, že pro přehlednost promluvy do ní pak nemusejí být zakomponovány všechny informace, z nichž některé mohou být pro uživatele nadbytečné. Ty se mohou zobrazit na obrazovce počítače, např. v graficky přehledné podobě.

Jako příklad uveďme hypotetický dialogový systém pro rezervaci sedaček v kině. V případě čistě hlasového dialogového systému by uživatel na dotaz *Jaká místa jsou volná?* pravděpodobně dostal odpověď v podobě dlouhého výčtu sedaček, případně by se ještě musel např. doptat, která řada ho zajímá. Pokud se mu však schéma sedaček zobrazí na obrazovce, udělá si o naplnění kina mnohem lepší představu a může pak rovnou volit sedačky, které se mu zdají být atraktivní.

Městská hromadná doprava v Plzni

3


Plzeň je čtvrté největší město v České republice s rozlohou téměř 138 km² a necelými 170 000 obyvateli [úřa23]. Městská hromadná doprava (dále jen MHD) je tedy pro zajištění pohybu obyvatel na území města prakticky nezbytná. Drtivou většinu spojů MHD zajišťuje společnost **Plzeňské městské dopravní podniky, a.s.** (dále jen PMDP). Někdy lze pro přepravu na území města využít i některé vlakové a autobusové spoje regionálního charakteru, to však není populární především z důvodu dlouhých intervalů mezi jednotlivými spoji spojeného s nutností důkladnějšího plánování cest.

Hustota linek MHD v Plzni víceméně odpovídá rozloze a počtu obyvatel města. Nalezneme zde 47 linek MHD, z nichž 3 tramvajové zajišťují páteř celého dopravního systému, 10 trolejbusových zajišťujících spojení předměstí a sídlišť s centrem města, a 33 autobusových, z nichž většina má spíše doplňkový charakter nebo zajišťují spojení periferií s většími přestupními uzly [Plz23].

3.1 Informační zdroje pro širokou veřejnost

3.1.1 Zastávkové jízdní řády

Na každé zastávce MHD v Plzni můžeme nalézt zastávkový jízdní řád (JŘ) linky, které tuto zastávku obsluhují. Dozvíme se z něj časy odjezdů jednotlivých spojů z této zastávky v konkrétním směru linky, následující zastávky a přibližný čas, za který se na konkrétní zastávku dostaneme. Přestože papírové jízdní řády obecně zastarávají, tyto zastávkové jsou zatím nepostradatelné, protože cestující se přímo na zastávce dozví, kdy mu jede další spoj, aniž by musel např. vyhledávat spojení na svém mobilním telefonu. Doplněkem k papírovým zastávkovým JŘ jsou elektronické zastávkové JŘ, které můžeme nalézt na webu PMDP (kapitola 3.1.3) nebo v mobilní aplikaci Moje PMDP (kapitola 3.1.4). Ukázka papírového JŘ je zobrazena na obrázku 3.1.

4 		Technická → Košutka	Platnost od 03.01.2023	Plzeňské městské dopravní podniky PMDP Dennísová nábreží 12, Plzeň
PRACOVNÍ DNY				
01	Univerzita	0		
0	Technická	1		
1	Bory	2		
2	Borský park	3		
3	Dvořákova	4	37s 53s	
4	Náměstí Míru	5	03s 09s 19s 28s 36s 44s 52s	
5	Dobrovského	6	00s 08s 16s 24s 34s 41s 49s 58s	
6	Chodské náměstí	7	07s 13s 19 25s 31s 37s 43s 49s 53s	
8	Masarykovo náměstí	8	01 05s 14s 19s 26x 34s 39s 49s 56sx 59s	
9	U Práce	9	09s 14s 24s 34s 44s 54s	
12	Sady Pětatřicátníků	10	04s 14s 24s 34s 44s 54s	
15	Pod Záhorskem	11	04s 14s 24s 34s 44s 54s	
17	Zoologická zahrada	12	04s 14s 19s 24s 29s 34s 39s 44s 49s 53s	
18	Boženy Němcové	13	01s 09s 13s 21s 29s 33s 37s 41s 47s 53s 59s	
19	U Družby	14	05s 10s 15s 20s 25s 30 39s 48s 57s	
20	Sokolovská	15	06s 15s 24s 34s 43s 52s	
21	Severka	16	01s 10s 16 25s 29s 37s 45s 53s	
22	Ptzeňka	17	00s 08 16s 24s 32s 40s 49s 59s	
23	Košutka	18	09s 14s 23s 32s 42s 52s	
		19	02s 12s 22s 32s 40sx 48s 58sx	
		20	08s 26s 46s	
		21	06s 26s 46s 56s	
		22	06s 16s 21s 26s 38sn	
		23	08sn 38sn	
SOBOTY				
0		0		
1		1		
2		2		
3		3		
4		4	56s	4 56s
5		5	16s 36s 46s 56s	5 16s 36s 56s
6		6	06s 16s 26s 36s 56s	6 16s 36s 56s
7		7	16s 36s 56s	7 16s 36s 56s
8		8	16s 36s 56s	8 16s 36s 56s
9		9	16s 36s 56s	9 16s 36s 56s
10		10	16s 36s 56s	10 16s 36s 56s
11		11	16s 36s 56s	11 16s 36s 56s
12		12	16s 36s 56s	12 16s 36s 56s
13		13	13s 28s 43s 58s	13 13s 28s 43s 58s
14		14	05s 13s 28s 43s 58s	14 13s 28s 43s 58s
15		15	13s 28s 43s 58s	15 13s 28s 43s 58s
16		16	13s 28s 43s 58s	16 13s 28s 43s 58s
17		17	13s 28s 43s 58s	17 13s 28s 43s 58s
18		18	13s 28s 43s 58s	18 13s 28s 43s 58s
19		19	13s 28s 43s 58s	19 13s 28s 43s 58s
20		20	16s 21sx 36s 56s	20 16s 21sx 36s 56s
21		21	16s 36s 56s	21 16s 36s 46s 56s
22		22	06s 16s 26s 38sn	22 06s 16s 26s 38sn
23		23	06sx 08sn 23s 38sn	23 06sx 08sn 23s 38sn
Ve vozidle patří ař a přepravní podmínky dopravce.				
s	Spoj s bezbarérové přístupným vozidlem		n spoj křžuje v zast. Sady Pětatřicátníků s linkami č. 1 a 2	
x	jede jen do zast. U Práce, dále na Slovany a do vozovny			
Vytvořeno 10.04.2023 16:58:50				
Na základě dat PMDP, a.s. zpracováno společností ABIRAL CZ s.r.o. www.abiral.cz				

Obrázek 3.1: Ukázka zastávkových JŘ v papírové verzi (generováno na jizdnirady.pmdp.cz)

3.1.2 Elektronické informační tabule

LED diodové informační tabule se vyskytují zpravidla na **frekventovanějších zastávkách**, co se výskytu cestujících týče, a přinášejí informace o odjezdech nejbližších spojů. Informace jsou zobrazovány po řádcích, přičemž každý řádek obsahuje číslo linky, směr linky a dobu do odjezdu spoje v minutách, příp. pravidelný čas odjezdu. Výhodou je okamžitě dostupná informace o tom, jak dlouhou dobu stráví cestující čekáním na zastávce, navíc je do doby zbývající do odjezdu započítáváno i případné **zpoždění** spoje.

Existují dvě varianty těchto informačních tabulí. Na vytížených přestupních uzlech jsou tabule umístěny na samostatných sloupech a disponují větším počtem řádků (obrázek 3.2a). Na některých méně vytížených, avšak v poslední době modernizovaných zastávkách, jsou informační tabule zakomponovány přímo do zastávkové vývěsky, avšak s menším počtem řádků (obrázek 3.2b).



(a) Zastávka U Práce

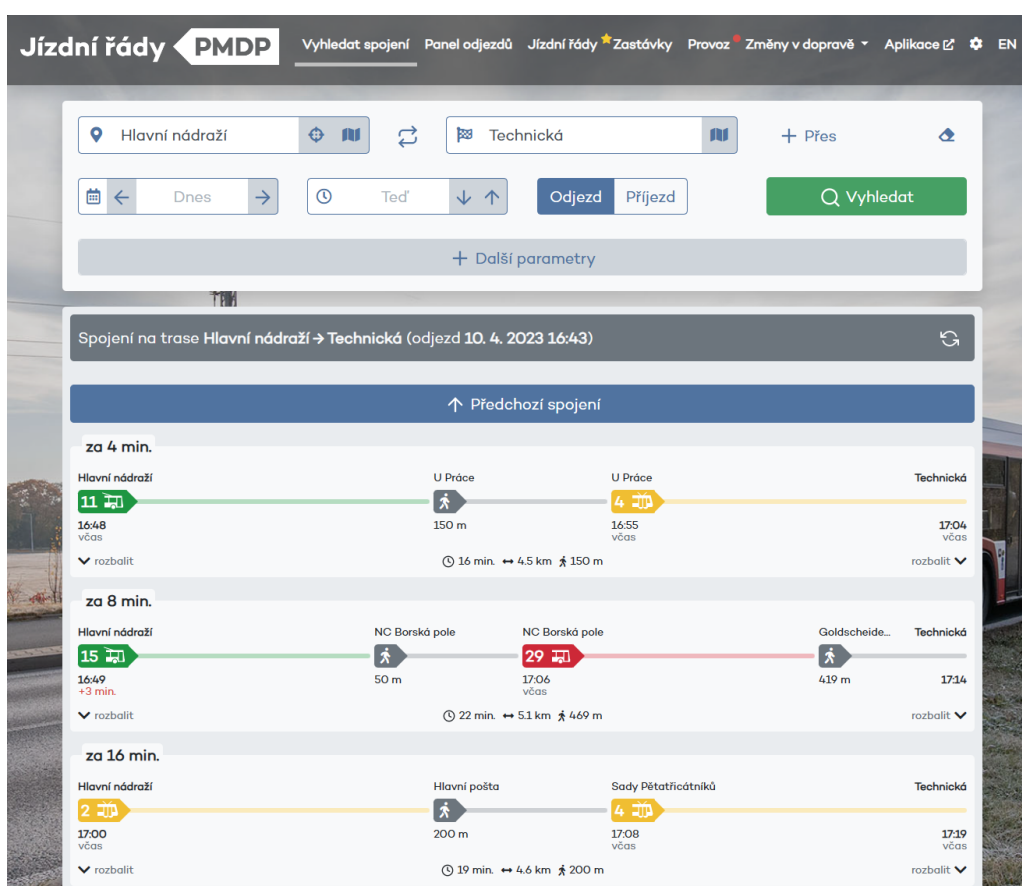


(b) Zastávka Bory

Obrázek 3.2: Zastávkové informační tabule

3.1.3 Web PMDP

Webové stránky PMDP dostupné na adrese jizdnirady.pmdp.cz prošly před několika lety zásadním redesignem a v současné době poskytují cestujícím kromě obecných informací o plzeňské MHD i vyhledávač spojení, virtuální tabule odjezdů jednotlivých zastávek, zastávkové JŘ a mnoho dalšího, a to vše v moderním a přehledném uživatelském rozhraní. Hlavní funkcí, kterou konkurenční weby postrádají, jsou pak **informace o mimořádnostech** v provozu i plánovaných událostech či výlukách na území města Plzně. Cestující tak čerpá informace přímo od zdroje, tím pádem hrozí minimální riziko zastaralých či nepravdivých informací. Ukázka uživatelského rozhraní webu je zobrazena na obrázku 3.3.

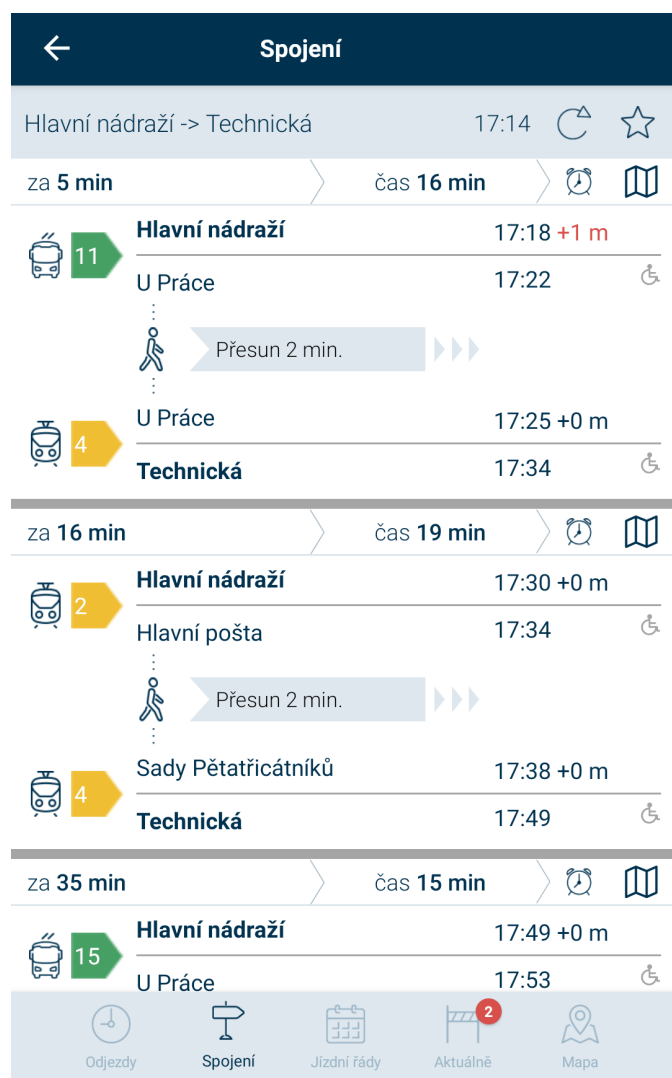


Obrázek 3.3: Webové rozhraní vyhledávače spojení PMDP (jizdnirady.pmdp.cz)

3.1.4 Aplikace Moje PMDP

Mobilní aplikace Moje PMDP je dostupná pro operační systémy Android a iOS a kromě příslušných mobilních obchodů ji lze stáhnout prostřednictvím webové stránky www.pmdp.cz/informace-o-preprave/mobilni-aplikace. V podstatě disponuje stejnými funkcemi jako webové stránky PMDP (kapitola 3.1.3), uživatelské rozhraní je ovšem přizpůsobeno dotykovým displejům dnešních chytrých telefonů.

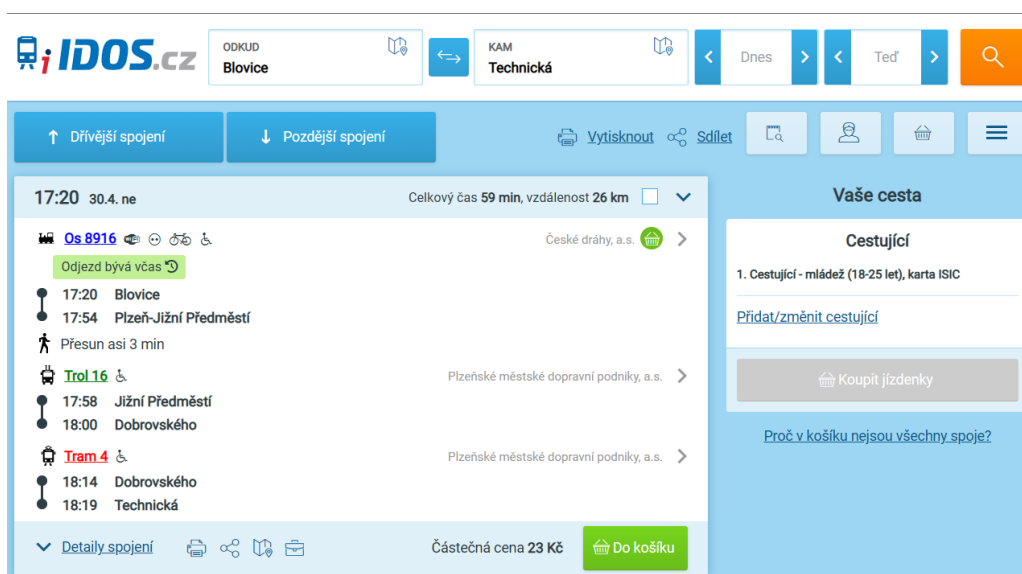
Přestože sám osobě aplikaci používám, mám k ní pár připomínek. Aplikace je sice stále aktualizována, často však náhodně přestává fungovat, nevydrží otevřená na pozadí a většina jejích funkcí je dostupná pouze online (s výjimkou offline zastávkových JŘ). Ukázka uživatelského rozhraní této aplikace je zobrazena na obrázku 3.4.



Obrázek 3.4: Uživatelské rozhraní aplikace Moje PMDP

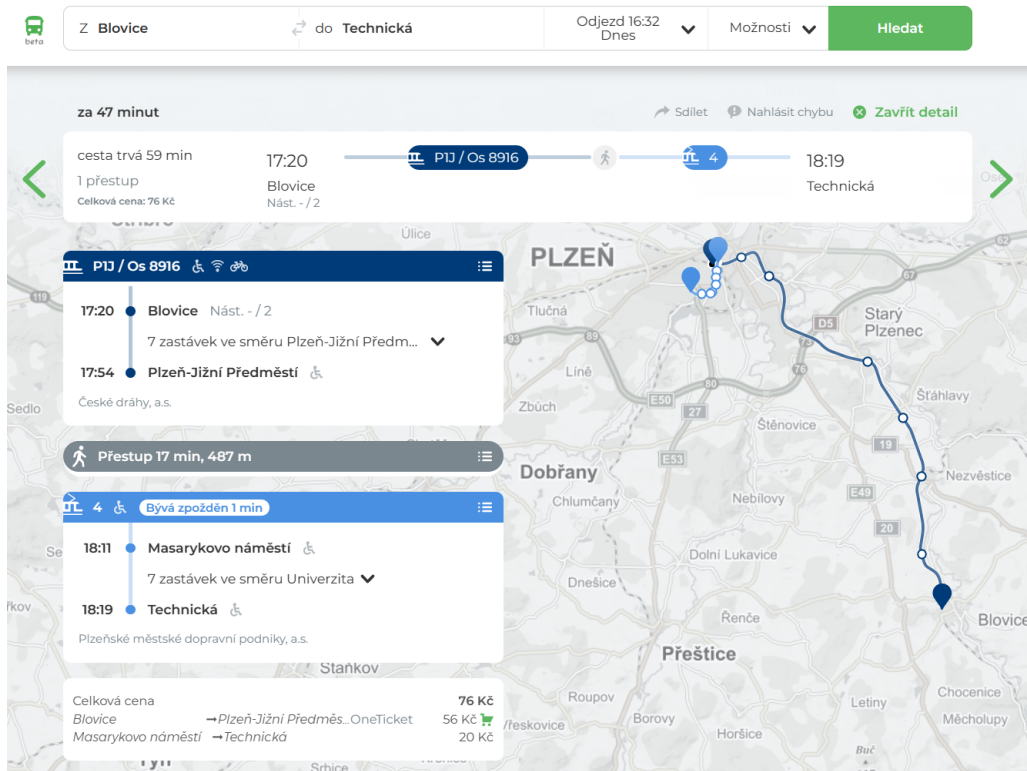
3.1.5 Další komerčně dostupné informační zdroje

Jednou z nejpoužívanějších služeb pro vyhledávání spojení a odjezdů spoju hromadné dopravy v ČR je webový server **IDOS**, provozovaný společností CHAPS spol. s r.o., dostupný na adrese www.idos.cz. Ten disponuje celostátními daty a lze jej tedy použít i pro vyhledávání spojení nad rámec systému plzeňské MHD (např. pokud hledám spojení se svého města na konkrétní zastávku v Plzni). IDOS existuje rovněž jako mobilní aplikace, avšak pro vyhledávání spojení i odjezdů je nutné připojení k internetu – jízdní řády nelze stáhnout a používat offline. Ukázka webového rozhraní aplikace IDOS je zobrazena na obrázku 3.5.



Obrázek 3.5: Webové rozhraní aplikace IDOS (dostupná z idos.cz, odstraněny reklamy)

Za zmínku rovněž stojí webové jízdní řády (resp. vyhledávač spojení) od společnosti **Seznam** dostupné na adrese www.jizdnirady.seznam.cz. Tato webová aplikace je poměrně mladá a přestože se stále oficiálně nachází v beta verzi, poskytuje moderní uživatelské rozhraní a díky propojení s aplikací [Mapy.cz](http://mapy.cz) dokáže spojení zobrazit i na mapě. I k této službě existuje aplikace pro mobilní telefony, její funkce jsou však rovněž dostupné pouze v online režimu. Ukázka webového rozhraní této aplikace je zobrazena na obrázku 3.6.



Obrázek 3.6: Webové rozhraní aplikace Jízdní řády Seznam.cz (dostupná z jizdnirady.seznam.cz)

3.2 Otevřená data pro vývojáře

3.2.1 Plzeň a Plzeňský kraj

PMDP na vyžádání poskytuje své jízdní řády ve formátu **GTFS** (General Transit Feed Specification), což je mezinárodně rozšířený formát pro textovou reprezentaci jízdních řádů, která umožňuje strojové zpracování. Jedná se vlastně o předem stanovenou strukturu textových souborů připomínajících jednoduchou databázi. Podrobný popis této specifikace lze najít na www.gtfs.org.

Bohužel, PMDP ani IDPK (Integrovaná doprava Plzeňského kraje) nezávislým vývojářům **neposkytují aktuální otevřená data o dopravních spojkách** na území města Plzně či celého Plzeňského kraje, např. ve formě otevřeného REST API, přestože s množstvím informačních kanálů zmíněných v kapitole 3.1 je absence společného API téměř vyloučena. Některá otevřená data týkající se dopravy město Plzeň zveřejňuje na webové stránce www.opendata.plzen.eu, až na pár výjimek se však jedná o data převážně statická (např. osy tramvajových kolejí, polohy zastávek atd.). Nezávislý vývojář má v tomto případě několik možností, jak absenci veřejně dostupných dynamických dat obejít.

Jedna z možností je si během vývoje vystačit s jízdními řády GTFS a např. vyhledávač spojení si naprogramovat svépomocí, což je značně obtížná úloha, navíc není možnost získat aktuální informace o zpoždění atd. Alternativně lze data *těžít* z veřejně dostupných zdrojů, to je však řešení velmi nerobustní a pravděpodobně dá více práce data *vytěžit* než vyvinout samotnou aplikaci. V případě vážného zájmu o spolupráci by potenciálně bylo možné přístup k API získat vyjednáváním s PMDP, nicméně by pravděpodobně dalo hodně práce svoji žádost obhájit.

3.2.2 Praha a Středočeský kraj jako vzor

Abychom mohli situaci v Plzni, potažmo v Plzeňském kraji, označit za nevyhovující, pojďme se pro srovnání podívat do sousedního Středočeského kraje, potažmo do Prahy, kde hromadnou dopravu organizuje společnost ROPID (Regionální organizátor Pražské integrované dopravy). Realtime otevřená data Pražské integrované dopravy lze čerpat na webových stránkách www.pid.cz/o-systemu/opendata, případně www.api.golemio.cz/v2/pid/docs/openapi.

Kromě jízdních řádů ve formátu GTFS poskytuje ROPID rovněž **otevřené REST API**, ze kterého lze čerpat aktuální informace např. o poloze konkrétních spojů, jejich zpoždění, příští zastávce apod. Rovněž jsou k dispozici odjezdové tabule jednotlivých zastávek. Dostupnost těchto dat dává amatérským i profesionálním vývojářům mnohem širší pole působnosti v oblasti vývoje aplikací souvisejících s cestováním hromadnou dopravou a je to pro ně skvělá příležitost, jak být prospěšní nejen sobě (např. zpeněžení vývoje), ale i společnosti, potažmo cestujícím.

Všimněme si však, že ani zde není dostupné jednoduché API, které by dokázalo člověku vyhledat spojení odpovídající zadaným parametrům. Pravděpodobně lze tak soudit na základě sofistikovanosti algoritmů vyhledávajících spojení, které si profesionální vývojáři střeží a nechtějí jejich využití poskytovat široké veřejnosti bezplatně.

3.2.3 Situace na celostátní úrovni

Obecně je v České republice situace ohledně dostupnosti jízdních řádů jakožto otevřených dat velmi tristní. V roce 2014 sice vyšla v účinnost vyhláška o jízdních rádech veřejné linkové dopravy (122/2014 Sb.), v jejímž § 9 odst. 8 se říká, že jízdní řády musí být uveřejňovány způsobem umožňujícím automatizované zpracování, není však dále specifikováno, jaké konkrétní požadavky by měly splňovat. Společnost CHAPS zmíněna výše tedy jakási surová data sice zveřejňuje, nedává k nim však k dispozici žádnou dokumentaci a podle některých expertů zveřejněná data dokonce nejsou formátově validní [Číž23]. Výsledek je tedy takový, že se ve zveřejněných datech vývojář prakticky nemá šanci vyznat.

Vývoj hlasového dialogu

4

4.1 Návrh hlasového dialogu

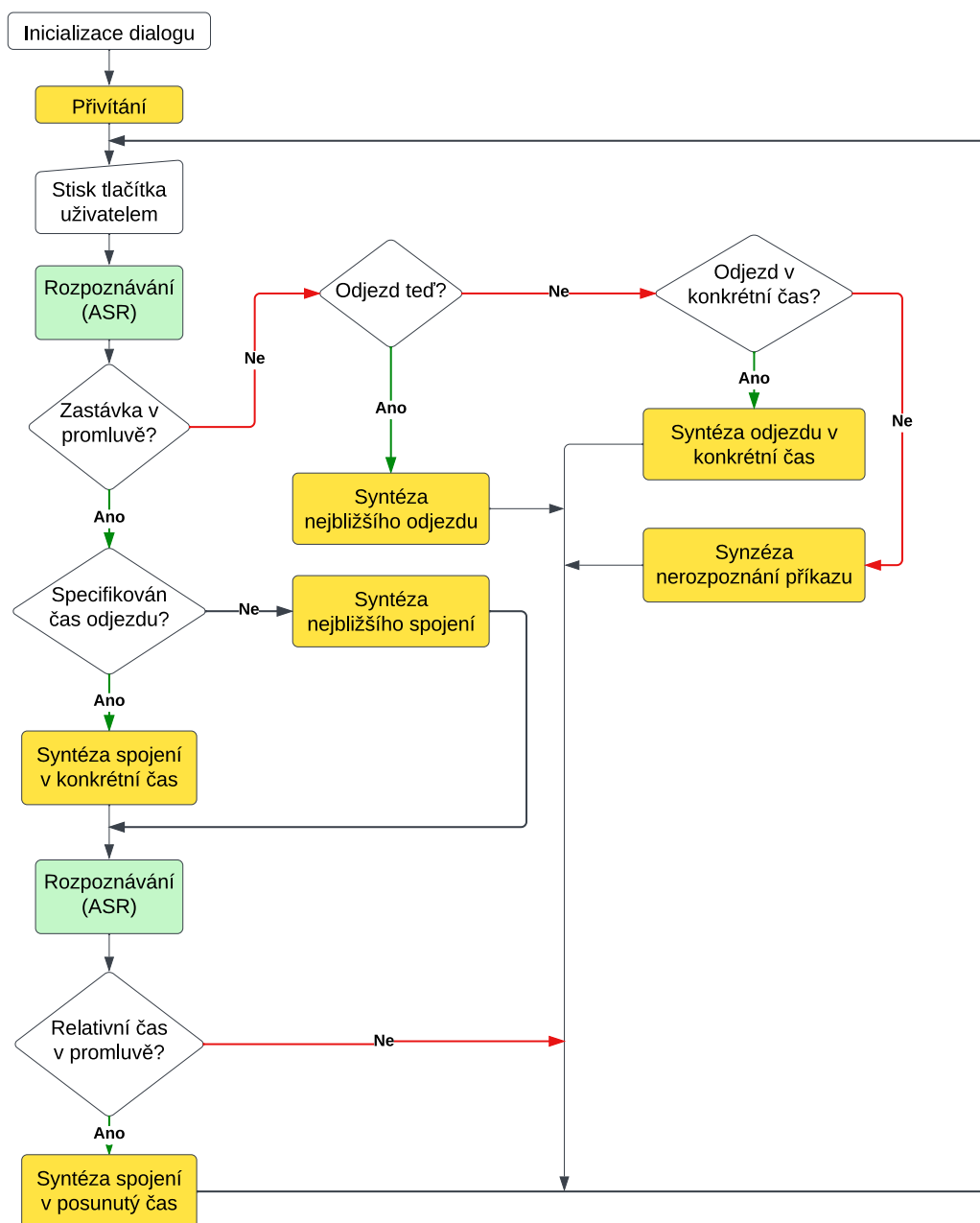
Samotný návrh struktury dialogu je důležitou úlohou návrháře. Kromě **vymezení funkcí**, které bude dialog schopen obsloužit, je důležité, aby mohl uživatel všech těchto funkcí pomocí hlasových promluv dosáhnout a zároveň pro něj byl dialog intuitivní natolik, aby mohl s hlasovým agentem komunikovat i s minimem znalostí o struktuře a funkcích dialogového systému. Pojďme si popsat některé klíčové funkce, které je třeba v této kompletní aplikaci hlasového dialogu implementovat.

4.1.1 Vývojový diagram z pohledu uživatele

Pro lepší představu o struktuře a fungování hlasového dialogu byl vytvořen vývojový diagram, který z pohledu uživatele zachycuje probíhající procesy, jejich závislost, větvení atd. Diagram je vykreslen na obrázku 4.1 a bude na něj odkazováno i v následujících kapitolách. Procesy jsou obecně značeny jako obdélníky, ty zelené symbolizují proces rozpoznávání řeči a ty červené proces syntézy řeči. Větvení na základě podmínek je znázorněno kosodélníky a akce uživatele (stisk tlačítka) lichoběžníkem.

4.1.2 Rozpoznávání řeči

Samotné rozpoznávání řeči je již implicitně řešeno pomocí frameworku SpeechCloud a v této práci jej budeme považovat za black box (tento termín se v kybernetice používá pro systém, u něhož je známa pouze závislost výstupu na vstupu, nikoli však jeho vnitřní struktura). V konkrétní aplikaci můžeme alespoň specifikovat, za jakých okolností bude systém rozpoznávat řeč (resp. pořizovat její nahrávku).



Obrázek 4.1: Vývojový diagram hlasového terminálu

Pokud chce uživatel zahájit dialog, musí stisknout příslušné tlačítko uživatelského rozhraní. U některých očekávaných promluv během dialogu není stisk tlačítka vyžadován. Ukončení promluv SpeechCloud detekuje automaticky a spolu s koncem promluvy ukončí i zvukový záznam tak, aby mohla být nahrávka uložena a zpracována rozpoznávačem řeči.

4.1.3 Syntéza řeči

Syntéza řeči je podobně jako rozpoznávání řešena implicitně pomocí Speech-Cloudu, máme tak k dispozici nástroj pro syntézu řeči přímo ze psaného textu. Syntéza by měla obsahovat by měla být kompromisem mezi informační bohatostí, přirozeností a délkou promluvy stroje. Hlavním úkolem syntézy v této aplikaci je předávání informací o spojeních či odjezdech, jež by byly pro uživatele relevantní na základě jeho požadavků. Podrobněji se podobě promluv budeme věnovat v kapitole 4.3.2.4.

SpeechCloud nabízí v základu několik mužských i ženských hlasů pro syntézu řeči. Tato aplikace mluví hlasem interně pojmenovaným Alena, který je ve frameworku nastaven jako výchozí a rovněž splňuje základní požadavky na přirozenost, sympatičnost, pronikavost a srozumitelnost i v případě nasazení v prostředí pouličního hluku. Do budoucna se uvažuje o nahrání vlastního hlasu (viz kapitola 5.4), je však ještě třeba vyhodnotit přínos oproti výchozím hlasům.

4.1.4 Funkce související s jízdními řády

Právě tyto funkce jsou klíčové pro skutečný přínos aplikace. Přestože zatím má aplikace spíše demonstrativní účely pro ukázkou jednoho z mnoha možných nasazení HDS, v případě splnění alespoň některých bodů z kapitoly 5.4 se aplikace může stát dostupnou pro širší masu lidí. V současné době aplikace disponuje dvěma hlavními funkcemi – vyhledávání spojení z konkrétní zastávky na libovolnou zastávku a vyhledávání odjezdů spojů z konkrétní zastávky.

4.1.4.1 Vyhledávání spojení

Pro vyhledání vhodného spojení na libovolnou zastávku musí uživatel zmínit konkrétní zastávku a případně, čas, kdy chce jet (zatím implementováno jen několik základních časů: ráno, dopoledne, atd.). V momentě, kdy je spojení nalezeno, lze ještě blíže specifikovat, jestli by nebylo lepší spojení dříve či později (implementovány relativní časy: čtvrt hodiny, půl hodiny, tři čtvrtě hodiny a jedna hodina dopředu/dozadu). Přehledně je tento proces znázorněn na obrázku 4.1.

4.1.4.2 Vyhledávání odjezdů

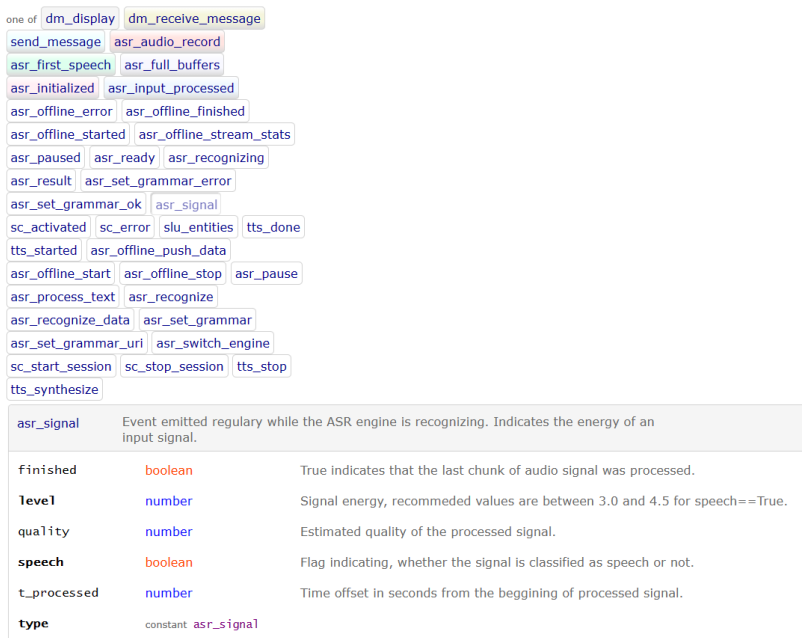
Pro vyhledání odjezdu spoje stačí hlasovému agentovi ohlásit, zda hledám nejbližší odjezd nebo odjezd v konkrétní čas (dopoledne, poledne, odpoledne, atd.). Funkce nejbližšího odjezdu do značné míry supluje informační tabule z kapitoly 3.1.2, další funkce je pak doplňuje.

Zatím se v případě odjezdové tabule jedná o statická data z jízdních řádů, ale pokud se povede v budoucnu aplikaci napojit na API PMDP (viz kapitola 5.4), bude do času odjezdu započteno i zpoždění, což je z pohledu cestujícího důležitá informace zvláště, jedná-li se o trolej-busové a autobusové spoje, jejichž dochvilnost je obvykle závislá na hustotě provozu a hlavně v denních špičkách nabírají zpoždění i vyšší jednotky minut.

4.2 Představení frameworku SpeechCloud

SpeechCloud je framework pro tvorbu hlasových dialogových systémů, který je vyvíjen na Fakultě aplikovaných věd Západočeské univerzity v Plzni ve spolupráci s její spin-off firmou SpeechTech, s.r.o. Je psán v jazyce Python a skrz API umožňuje komunikaci s front-endovou částí aplikace napsanou v jazyce JavaScript. Framework není volně dostupný, v případě zájmu o vývoj je nutné kontaktovat pověřenou osobu (někoho z lidí s platformou pracujících, např. Ing. Jan Švec, Ph.D. nebo Ing. Luboš Šmídl, Ph.D).

Dokumentace je poskytována prostřednictvím v tzv. DOCson formátu. Po otevření odkazu s dokumentací (dostupný skrz každou instanci SpeechCloudu) se objeví přehled všech funkcí API a po kliknutí na konkrétní funkci se zobrazí i poskytované proměnné a jejich datové typy. Vzorový příklad můžeme vidět na obrázku 4.2.



The image shows a screenshot of the SpeechCloud API documentation. At the top, there is a list of API endpoints and events, such as `dm_display`, `dm_receive_message`, `send_message`, `asr_audio_record`, `asr_first_speech`, `asr_full_buffers`, `asr_initialized`, `asr_input_processed`, `asr_offline_error`, `asr_offline_finished`, `asr_offline_started`, `asr_offline_stream_stats`, `asr_paused`, `asr_ready`, `asr_recognizing`, `asr_result`, `asr_set_grammar_error`, `asr_set_grammar_ok`, `asr_signal`, `sc_activated`, `sc_error`, `slu_entities`, `tts_done`, `tts_started`, `asr_offline_push_data`, `asr_offline_start`, `asr_offline_stop`, `asr_pause`, `asr_process_text`, `asr_recognize`, `asr_recognize_data`, `asr_set_grammar`, `asr_set_grammar_uri`, `asr_switch_engine`, `sc_start_session`, `sc_stop_session`, `tts_stop`, and `tts_synthesize`.

Below the list, there is a detailed view of the `asr_signal` event. It is described as: "Event emitted regularly while the ASR engine is recognizing. Indicates the energy of an input signal." The event has several properties:

Property	Type	Description
<code>finished</code>	<code>boolean</code>	True indicates that the last chunk of audio signal was processed.
<code>level</code>	<code>number</code>	Signal energy, recommended values are between 3.0 and 4.5 for <code>speech==True</code> .
<code>quality</code>	<code>number</code>	Estimated quality of the processed signal.
<code>speech</code>	<code>boolean</code>	Flag indicating, whether the signal is classified as speech or not.
<code>t_processed</code>	<code>number</code>	Time offset in seconds from the beginning of processed signal.
<code>type</code>	<code>constant asr_signal</code>	

Obrázek 4.2: API dokumentace frameworku SpeechCloud

4.3 Vývoj backendu

Samotná aplikace pro řízení dialogu se nachází na virtuálním stroji zřízeném pomocí frameworku OpenNebula. Tuto službu poskytuje pro studijní účely Centrum informatizace a výpočetní techniky (CIV) Západočeské univerzity v Plzni. Konkrétně se jedná o stroj disponující veřejnou IP adresou `sulis31.zcu.cz`, na kterém běží operační systém **Debian**. Ten ovládáme pomocí textového terminálu, a to přímo z lokální instalace OS založeného na Debianu (např. Ubuntu a jeho příchutě), z prostředí OS Windows pak např. pomocí utility PuTTY. Všechny soubory nezbytné pro správné fungování dialogové aplikace se nacházejí v příslušné složce na virtuálním stroji (na jednom stroji může běžet více instancí SpeechCloudu).

Všechny kódy nutné k běhu aplikace, včetně těch vytvořených v rámci této práce, jsou k dispozici ve veřejném repozitáři služby **GitHub** na webové adrese: www.github.com/simio99/speechcloud-MHD.

4.3.1 Výchozí struktura skriptu

Pro účely této práce nás bude zajímat hlavně soubor `main.py`. V tom bude znamenána hlavní struktura dialogu, přičemž běh tohoto kódu je nezbytný pro běh celé aplikace. Pro správnou funkci dialogového systému je potřeba provést korektní instalaci knihovny `SpeechCloud.dialog` na virtuální stroj podle dokumentace, nebo o to požádat některou z pověřených osob na katedře (mnou zvolený způsob). Tento soubor v základu obsahuje strukturu vzorového dialogu, pomocí kterého je možné si prostředí osahat a dozvědět se, jak fungují některé základní metody frameworku.

Zde je namístě podotknout, že komunikace mezi serverem a klientem probíhá **asynchronně**. Asynchronní funkce jsou v jazyce Python označeny klíčovými slovy `async` a `await` a práce s nimi vyžaduje alespoň základní znalosti problematiky asynchronní komunikace ¹.

Hlasový dialog je v tomto kódu reprezentován jako samostatná třída, která ve výchozím stavu definuje několik pomocných funkcí. Důležitá je především funkce `async def main(self)`. To, co obsahuje, je považováno za řízení hlasového dialogu a bude se tedy po inicializaci dialogu vykonávat. Ve výchozím nastavení jsou to funkce `await self.slu()`, `await self.recog()` a `await self.synth()`. Každá z těchto funkcí je v podstatě samostatný hlasový dialog, který demonstruje stěžejní funkce hlasového dialogu (možnosti rozpoznávání, syntézy a práce s gramatikami). V jeden moment však může běžet jen jeden hlasový dialog, proto je vždy jen jedna funkce aktivní. Na tomto místě se bude nacházet i funkce reprezentující algoritmus našeho vlastního hlasového dialogu.

¹Více o principech a použití asynchronní komunikace v Pythonu se můžete dočíst např. na naucse.python.cz/lessons/intro/async.

4.3.2 Vlastní funkce pro konkrétní úlohu

4.3.2.1 Extrakce dat z aplikace IDOS

Kvůli nedostupnosti otevřených řešení pro vyhledávání spojení hromadné dopravy, jak je popsáno v kapitole 3.2, se jako nejdostupnější varianta pro demonstraci funkce hlasového terminálu pro jízdní řády městské hromadné dopravy ukázala možnost **extrakce dat z webové stránky** obsahující výsledky vyhledávání aplikace IDOS. Přesně to implementuje funkce `get_spoje`, která je součástí skriptu. Nejprve je pomocí známé funkce `requests.get()` stažena webová stránka obsahující výsledky vyhledávání spojení na konkrétní zastávku v konkrétní čas. Soubor obsahující webovou stránku je pak pomocí známé knihovny BeautifulSoup² parsován na HTML elementy, z nichž jsou vybírány a ukládány do struktury `ty`, které nesou důležité informace o spojeních.

4.3.2.2 Vyjádření v rozdílu dvou absolutních časů v minutách

Odjezdy nejbližších spojů z konkrétní zastávky (konkrétně zastávky Technická) máme z aplikace IDOS dostupné v absolutních časech (např. 12:34). Jedná-li se však o blížící se odjezd spoje, pro cestující je často užitečnější časový údaj vyjadřující zbývající čas do odjezdu spoje. Tento údaj již nyní v minutách vyjadřují informační tabule na obrázku 3.2 a v našem hlasovém dialogu bychom chtěli tento údaj zachovat. Proto byla napsána funkce `time_diff` se dvěma parametry časových údajů vyjádřenými jako řetězce ve formátu HH:MM.

4.3.2.3 Přičítání a odečítání relativních časů od absolutního

Může se stát, že vhodný spoj nalezený hlasovým terminálem nebude cestujícímu vyhovovat, přestože výsledek odpovídá zadání cestujícího. V takovém případě by pak měl cestující mít šanci blíže specifikovat čas, který měl na mysli (např. o čtvrt hodiny dříve, o půl hodiny déle atd.). Tato funkce by tedy měla k času původně nalezeného spojení přičíst nebo odečíst relativní časový údaj. Výsledek této operace by se pak měl stát výchozím časem pro nové vyhledání spojení.

4.3.2.4 Generování odpovědi

Generování odpovědi je důležitou součástí dialogového systému. V případě obecného dialogového systému je generování odpovědi poměrně složitá úloha, kde se v dnešní době uplatňují složité jazykové modely. Naše aplikace má však zatím úzké

²Informace a dokumentace o knihovně BeautifulSoup dostupné na www.beautiful-soup-4.readthedocs.io

zaměření a navíc promluvy týkající se jízdnicích řádů lze poměrně snadno strukturovat do šablony tak, jako se tomu děje např. v nádražním rozhlase.

V případě generování odpovědi na dotaz **spojení** je důležité vyrovnat se hlavně s různým počtem přestupů. Délku spojení lze vyčíst z délky pole, které reprezentuje aktuální nalezené spojení. V závislosti na počtu přestupů pak existují dvě šablony pro odpověď podle toho, jestli hledáme spojení právě teď nebo v jiný konkrétní čas:

Nyní nastupte do <linka>, pravidelný odjezd <čas>, a vystupte na zastávce <výstup>. Tam přestupte na <linka>, pravidelný odjezd <čas>, a vystupte na zastávce <výstup>. Tam...

Do těchto šablon byly zakomponovány hlavně údaje o lince a časech odjezdu. Časy příjezdu totiž nejsou z pohledu cestujícího tak důležité, a pokud by se je přece chtěl cestující dozvědět, celý výpis spojení se mu zobrazí v grafickém uživatelském rozhraní.

V případě generování odpovědi na dotaz **odjezdu** se bude šablona pro odpověď (alespoň prozatím) skládat pouze z jedné věty. Pokud se budeme ptát na nejbližší odjezd tramvaje (zastávku technická obsluhuje pouze linka č. 4), dočkáme se odpovědi: *Nejbližší tramvaj odjíždí za <minuty> minut.* V případě, že nás zajímá odjezd tramvaje v konkrétní čas (např. odpoledne, večer, atd.), odpověď bude mít tvar: *Tramvaj odjíždí v <čas>.*

V případě odpovědi na nejbližší odjezd bylo potřeba vypořádat se s různým skloňováním různého počtu minut (1 minuty, 2 minuty, 5 minut). Tento problém lze naštěstí relativně snadno vyřešit pomocí podmínek. Pro 0 minut volíme místo časového údaje promluvu *právě nyní*. Pro jednu minutu volíme odpověď *za jednu minutu*. Pro dvě až čtyři minuty volíme např. odpověď *za tři minuty*. Pro pět a více minut již postačí tvar *minut*.

4.3.3 Gramatiky

V případě hlasového dialogového systému, u kterého očekáváme pouze promluvy týkající se pouze určité oblasti (v tomto případě dotazy na spojení a odjezdy MHD) je vhodné tento okruh vymezit pomocí tzv. gramatik. SpeechCloud sice dokáže rozpoznávat řeč i bez gramatik, bez apriorní informace o tom, jaké entity (slova) v textu hledat, je však rozpoznávání nepřesné.

Abychom si gramatiky dokázali představit konkrétněji, vysvětlíme si její strukturu na příkladu. Na obrázku 4.3 je zobrazena gramatika pro seznam akcí, resp. příkazů, které je hlasový agent schopen splnit. Každý řádek obsahuje jeden typ příkazu, jehož formální jméno je uvedeno ve složených závorkách. V kulatých závorkách jsou pak uvedeny všechny promluvy, kterými se může uživatel vykonání tohoto příkazu dožadovat. Alternativní promluvy do sebe lze vnořovat, např. *chci (pomoci | pomoci)*, nebo lze dosadit i entitu z jiné gramatiky, např. na zastávku *< zastavky >*.

```
1 #ABNF 1.0 UTF-8 cs;  
2  
3 root $akce;  
4  
5 public $akce = (  
6 (zeptat | nevím | chci (pomoc|pomoci) | nápověda) {help} |  
7 (jak | dostat | dostanu | na zastávku <zastavky>) {spojeni} |  
8 (kdy | v kolik | jede | odjíždí | odjezd) {odjezd} |  
9 (už nic nechci | konec | už nic nepotřebuji | děkuji) {end}  
10 );  
11
```

Obrázek 4.3: Ukázka gramatiky příkazů (upraveno, zkráceno)

Sami jsme náš hlasový terminál testovali i bez přítomnosti gramatik, rozpoznávání slov tedy bez jakékoliv apriorní informace o účelu dialogu. Ukázalo se, že ani s kvalitní domácí náhlavní soupravou není takové rozpoznávání přesné. Problém dělají hlavně předložky, koncovky, vzájemně si podobná slova a v neposlední řadě názvy zastávek, které se běžně jinak v českém jazyce nevyskytují vůbec nebo jen velmi zřídka (např. Adelova, Samaritská, Panasonic atd.). Po přidání slov do jazykového modelu se však přesnost rozpoznávání zastávek prudce zvýšila a i když (zatím) nebyla statisticky vyhodnocována, během testování při vývoji již dochází k chybným rozpoznáním pouze velmi zřídka (dříve se jednalo o zhruba třetinu promluv).

4.4 Vývoj frontendu

Frontend, čili **grafické uživatelské rozhraní** aplikace, je **webová stránka** skládající se ze souboru HTML definující strukturu stránky, souboru kaskádových stylů CSS definujících vzhled stránky a souboru JavaScript zajišťujícího dynamičnost a interaktivitu webové stránky a propojení s instancí SpeechCloudu spuštěnou na virtuálním stroji. Propojení s virtuálním strojem je zajištěno pomocí protokolů WebSockets a SIP. Skripty zajišťující tuto komunikaci byly již předepsány během vytváření instance SpeechCloudu a vzorového uživatelského rozhraní, proto nebylo nutné se jimi v rámci této práce příliš zabývat.

4.4.1 Struktura HTML stránky

Design HTML stránky jsme se zatím snažili koncipovat tak, aby mohla být aplikace hlasového terminálu nasazena přímo na některé ze zastávek MHD. Tělo HTML stránky se tedy na základě předpokládaného použití skládá (od shora dolů) z následujících prvků:

- hlavička informující o přítomnosti *chytré zastávky*,
- odjezdová tabule (podobná té na obrázku 3.2),
- tlačítko mikrofону – stiskem se aktivuje rozpoznávání řeči,
- textová indikace aktuálního stavu aplikace (inicializace, rozpoznávání připraveno, rozpoznávání běží),
- tlačítka pro přerušování dialogu a zahájení nového dialogu,
- odkaz na mapu (schéma) MHD v Plzni, včetně názvů všech zastávek,
- okno pro textové zobrazení výsledků + logování (hlavně vývojářské účely),
- patička informující o účelu vývoje aplikace, autorovi a instituci (KKY ZČU).

4.4.1.1 Hlavička HTML stránky

Hlavička HTML stránky obsahuje kromě názvu stránky a základních metadat také importované CSS knihovny a skripty jazyka JS, kterých stránka pro svou funkcionalitu využívá. Zde stojí za zmínku, že jsme pro grafický návrh stránky využili rozšířeného frameworku **Bootstrap**³, který přináší spoustu předdefinovaných grafických stylů a poziční systém, kterých lze při tvorbě stránky využít. Využití Bootstrapu jednak šetří čas a jednak umožňuje i stránkám amatérského charakteru vypadat více profesionálně.

4.4.1.2 Tělo HTML stránky

Framework SpeechCloud má naštěstí pro použití webového uživatelského rozhraní přizpůsobené API, takže co se funkce zvuku a ovládání dialogu týče, stačí na stránku jen umístit příslušné elementy a označit je příslušným ID podle API.

Co se týká dat zobrazovaných na odjezdové tabuli, ta jsou zatím čerpána pouze z offline reprezentace odjezdů spojů ze zastávky Technická. Do budoucna bychom však rádi čerpali aktuální online data včetně zpoždění přímo od PMDP, bude-li k tomu společnost svolná (viz kapitola 5.4.1.1).

Odkaz na mapu (schéma) MHD zatím uživatele navede na obrázek oficiálního schéma PMDP na jeho stránkách. Do budoucna bychom rádi schéma integrovali přímo do samotné stránky.

Ukázky a popis uživatelského rozhraní viz kapitola 5.1.

³Informace, dokumentace a návody k použití frameworku Bootstrap dostupné na www.getbootstrap.com

4.4.2 Struktura JavaScriptu

V JavaScriptu jsou mimo jiné offline uloženy jízdní řády odjezdů tramvají ze stanice Technická, které jsou použity jako podklad pro odjezdovou tabuli zobrazenou na webové stránce, a to ve formě dvourozměrného pole, z nichž první rozměr je hodina (0-23) a v dalším rozměru jsou uloženy konkrétní minuty odjezdů v dané hodině. Reprezentace tedy víceméně odpovídá zastávkovým jízdní řádům na obrázku 3.1. Tuto funkcionalitu bych však chtěl do budoucna nahradit online daty poskytovanými přímo PMDP nebo dostupnými jinými prostředky (viz kapitola 5.4.1.1).

Dále JavaScript obsahuje většinou původní kód zajišťující funkcionalitu API mezi webovou stránkou a virtuálním strojem. S pomocí API funkce, která během rozpoznávání detekuje promluvu, jsme však dokázali vytvořit jednoduchou animaci ikony mikrofonu na webové stránce, aby bylo jasné, že správně funguje mikrofon. Tlačítko mikrofonu dále mění vzhled v případě, kdy není aplikace inicializována (nebo není povolen mikrofon), nebo detekce řeči vůbec neprobíhá (je podmíněna stiskem tlačítka). Tlačítko mikrofonu dále doplňují instrukce, co v daný okamžik dělat (stisknout, mluvit, atd.).

S pomocí API funkcí bylo také vytvořeno okno, které informuje o průběhu dialogu a hlavně zobrazuje nalezená spojení. Kvůli omezeným možnostem funkce výpis spojení zatím probíhá jen v textové podobě, do budoucna bychom však chtěli zapojit i grafické prvky.

Výsledná aplikace

5

V současné době je webová aplikace hlasového terminálu pro testovací a demonstrační účely umístěna na webové adrese:

<https://cak.zcu.cz:9444/edu/bp/simio//index.html>,

kde je přístupná pro všechny uživatele internetu, tedy i mimo univerzitní síť a bez jakýchkoliv nutných oprávnění. Vzhledem k tomu, že je aplikace stále ve vývoji, však může docházet k výpadkům, které pravděpodobně znamenají, že aplikace na virtuálním stroji je zastavena.

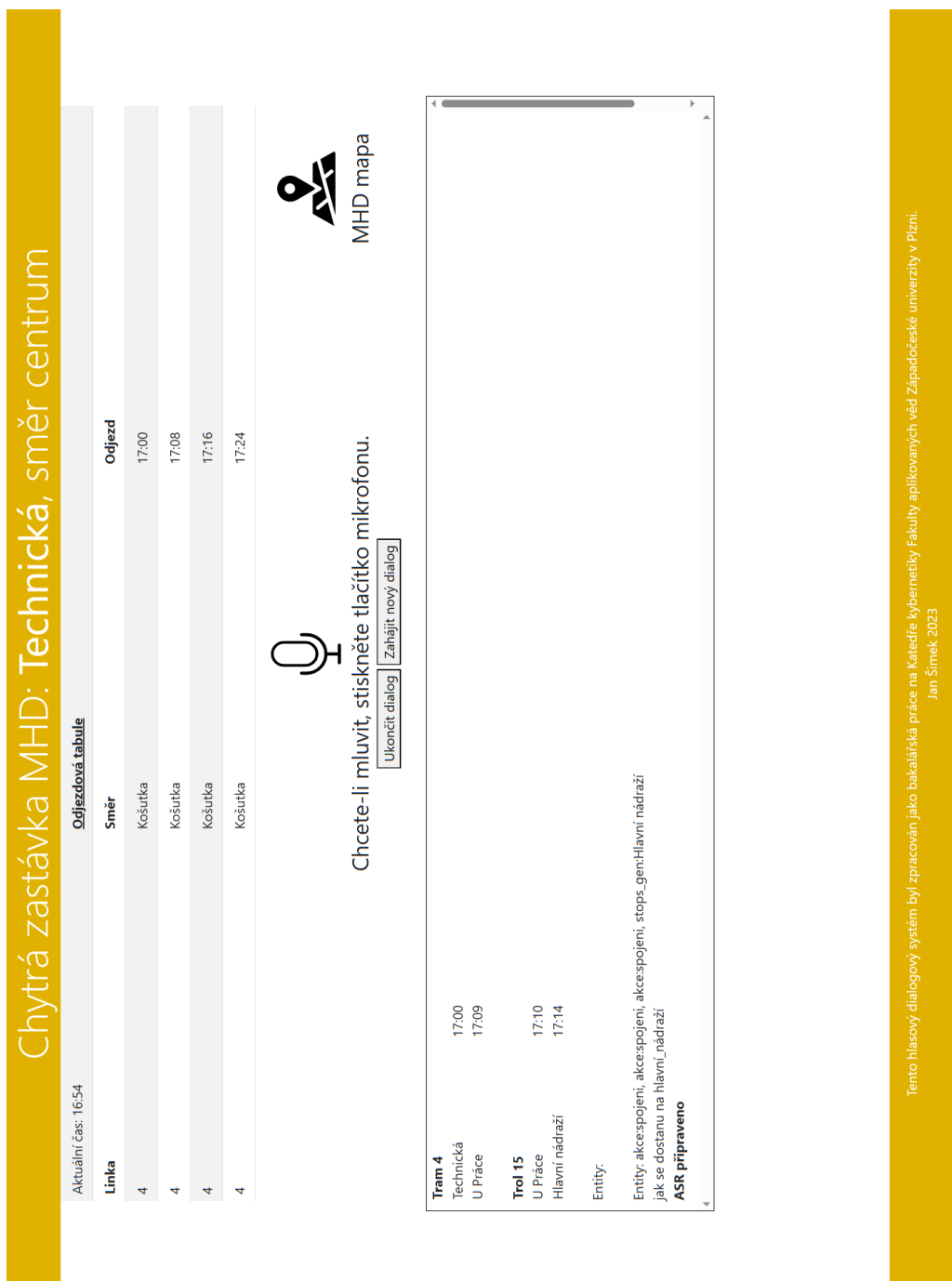
5.1 Uživatelské rozhraní

Návrh uživatelského rozhraní je o kompromisech mezi funkčností a líbivostí. V současné době může na náročnější uživatele uživatelské rozhraní působit poněkud stroze, nechtěli jsme však hned v prvopočátcích plochu uživatelského rozhraní zaplnit prvky, které s hlasovým dialogem a jeho využitím nesouvisí. Koncept uživatelského rozhraní by měl být výsledkem diskuse vývojáře, grafika a uživatele (nebo jejich skupin) a toho bychom se v budoucnu rádi drželi. V této práci byl však kladen důraz především na vývoj, uživatelské rozhraní je tedy čisté a účelné.

Ukázku uživatelského rozhraní můžeme vidět na obrázku 5.1. Od shora na něm můžeme vidět prvky, které popisuje kapitola 4.4.1, tedy hlavní nadpis, odjezdovou tabuli, tlačítko pro aktivaci dialogu, odkaz na mapu (schéma) MHD, tlačítko pro ukončení a restart dialogu a okno s výsledky.

Velký důraz byl kladen na **tlačítko mikrofону**, které má 4 různé vzhledy podle aktuálního stavu aplikace. Tyto vzhledy spolu s popisem jsou zobrazeny na obrázku 5.2 a znamenají:

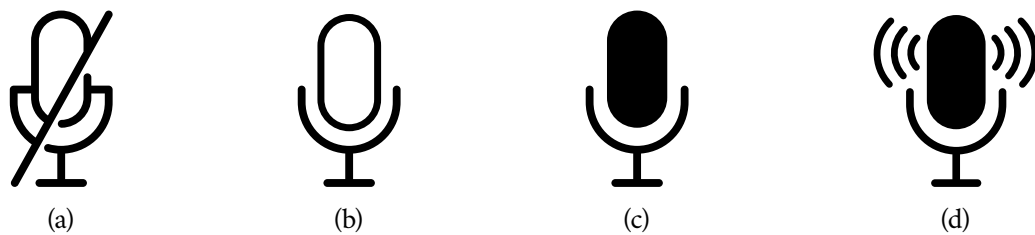
- Probíhá inicializace nebo není povolen přístup k mikrofону (obrázek 5.2a).
- Rozpoznávání připraveno, stiskněte tlačítko mikrofону (obrázek 5.2b).
- Rozpoznávání probíhá, ale není detekována řeč (obrázek 5.2c).
- Rozpoznávání probíhá a je detekována řeč (animace, obrázek 5.2d).



Obrázek 5.1: Uživatelské rozhraní aplikace hlasového terminálu

5.2 Příklady použití

Aplikaci můžeme využít k **vyhledání spojení** ze zastávky Technická na jakoukoliv jinou zastávku v systému plzeňské MHD. V tom případě stačí stisknout tlačítko a sdělit systému, na kterou zastávku se chceme dostat. Pro dotaz na spojení je nutné



Obrázek 5.2: Ikony mikrofonu indikující stav dialogu

v promluvě pronést některé z následujících slov: **jak, dostat, dostanu, spojení, do, na**. Jinak ale můžeme hovořit přirozenou řečí a celými větami, např. *Dobrý den. Prosím vás, jak se dostanu na náměstí Republiky?* nebo *Zdravím. Já hledám spojení na zastávku Divadlo Alfa. Pomůžete mi?* Na tuto otázku dostaneme odpověď v podobě nejbližšího možného spojení.

Pokud bychom chtěli **lépe specifikovat čas** odjezdu, musíme v promluvě pronést některý z výrazů upřesňující čas. Všechna možná slova včetně jejich skutečné časové interpretace, která systém zatím podporuje, jsou uvedena v kódu na obrázku 5.3, který je dobře čitelný i pro člověka neznalého technických detailů implementace (přesný popis v kapitole 4.3.3). Promluva pak může vypadat např. *Dobřej. Je nějak možný se večer dopravit na Sídlíště Bory?* nebo *Ve dvanáct hodin chci jet na zastávku Macháčkova*. Jak je vidět na obrázku 5.3, možnosti specifikace času jsou zatím poněkud omezené a do budoucna je plánováno systém vylepšit tak, aby dokázal rozpoznat jakýkoliv časový údaj, např. *třináct hodin, padesát sedm minut*.

```

1 #ABNF 1.0 UTF-8 cs;
2
3 root $cas;
4
5 public $cas = (
6   (ráno | sedm | sedmé) {7:00} |
7   (dopoledne | deset | desáté) {10:00} |
8   (poledne | dvanáct | dvanácté | oběd | oběda) {12:00} |
9   (odpoledne | patnáct | patnácté) {15:00} |
10  (vpodvečer | podvečer | sedmnáct | sedmnácté) {17:00} |
11  (večer | dvacet | dvacáté) {20:00} |
12  (teď | nyní | právě | nejbližší) {now}
13 );
14

```

Obrázek 5.3: Gramatika pro specifikaci času spojení nebo odjezdu

Pokud chceme čas nalezeného spojení ještě **upřesnit**, máme možnost po tom, co systém dokončí odpověď. Můžeme relativně upřesnit čas vyhledávání (např. *o hodinu později*) a od systému dostaneme novou odpověď. Všechny promluvy tohoto typu shrnuje kód na obrázku 5.4.

```
1 #ABNF 1.0 UTF-8 cs;  
2  
3 root $cas_rel;  
4  
5 public $cas_rel = (  
6     (čtvrť) {0:15} |  
7     (půl) {0:30} |  
8     (tři čtvrtě) {0:45} |  
9     (hodinu) {1:00} |  
10    (chvíli | okamžik) {0:15}  
11 );  
12
```

Obrázek 5.4: Gramatika pro upřesnění času již vyhledaného spojení nebo odjezdu

Pokud nás zajímá pouze **odjezd** spoje ze zastávky Technická, stačí v promluvě pronést pouze pronést některé ze slov **kdy**, **v kolik**, **jede**, **odjíždí**, **odjezd** a upřesnit čas podle gramatiky na obrázku 5.3, např. *Kdy to jede teď?* nebo *V kolik to jede ráno?* V případě blízkého odjezdu (do zhruba deseti minut) dostaneme odpověď v minutách, v opačném případě se z odpovědi dozvíme absolutní časový údaj (např. 13:10).

5.3 Testování na uživatelích

Abychom se na aplikaci dokázali podívat i **pohledem běžného uživatele**, jehož znalosti o aplikaci mohou být na rozdíl od vývojáře značně omezené, vytvořili jsme formulář s úkoly a hodnocením aplikace. Tento formulář jsme rozdali uživatelům, kteří se snažili plnit úkoly a podle obtížnosti plnění aplikaci hodnotit. Formulář byl z důvodů blízkého se termínu odevzdání této práce rozdán desítky lidí, výsledky průzkumu tedy nelze označit za plně průkazné, nicméně v první iteraci uživatelského testování může i takto malý vzorek respondentů ukázat slabá místa aplikace, na které se vývojář nezaměřil, protože si jich vůbec nevšiml nebo je nepovažoval za důležité.

Testovací formulář obsahuje webový odkaz na aplikaci (spolu s QR kódem), seznam úkolů, základní pokyny, tabulku pro vyhodnocení úkolů a dále oddíl s hodnocením, kde má uživatel za úkol ohodnotit aplikaci počtem hvězdiček (1 až 5) a vypsat postřehnuté klady a zápory aplikace. Celý testovací formulář je k nalezení jako příloha A.1 této práce.

Z tak malého počtu vzorků nemá smysl vyvozovat žádné statistické údaje, nicméně i tak lze z posbíraných formulářů vyčíst, co dělalo uživatelům problémy. Nejvíce problémů dělalo úkol 3: *Vyhledejte nejbližší spojení na zastávku Sady Pětatřicátníků a po skončení odpovědi se doptejte, zda by šlo vyhledat spojení o půl hodiny později.* Imple-

mentace této funkce není obecně natolik intuitivní, jak se během vývoje zdálo, a než uživatelé přišli na správný postup, trvalo to i vyšší jednotky minut. Naopak vyhledání nejbližšího spojení nebo odjezdu zvládali uživatelé dobře, nejhůře na 4 pokusy, Zeptání se na spojení nebo odjezd v konkrétní čas většinou nedělalo problém, což příkládám tomu, že si uživatelé již dialog vyzkoušeli na některé z předchozích úloh.

Kromě standardních funkcí vyhledávání spojení a odjezdů a srozumitelné syntézy řeči se uživatelé oceňují např. animace mikrofonu nebo přepis uživatelské promluvy, který jsem za důležitý nepovažoval, ale uživatelům se líbí, že ví, kterou část promluvy řekli nesrozumitelně.

Naopak by uživatelé ocenili rozšíření gramatik (resp. klíčových slov) tak, aby se na otázky dalo zeptat více způsoby než nyní nebo grafickou nápovědu klíčových slov v případě, že se jim nedaří správně zadat příkaz. Celkové hodnocení aplikace v tomto stádiu se pohybovalo mezi třemi a čtyřmi hvězdami z pěti.

5.4 Uvažovaná vylepšení do budoucna

Jako výhodu vývoje aplikace spatřuji to, že se nejedná o pevně ohraničený projekt s pevně zadaným cílem či termínem, naopak lze aplikaci v případě dostatku vůle aktualizovat, vylepšovat a rozšiřovat klidně i několik let či desetiletí. Rád bych této příležitosti využil, a proto jsem si do budoucna vytyčil několik více či méně ambiciózních cílů, kterých bych chtěl dosáhnout. Většinu bych chtěl splnit již ve své diplomové práci, v rámci které bych rád na vývoji hlasového terminálu pokračoval a během kterého bych dokázal uplatnit i znalosti nabyté v rámci svého magisterského studia.

5.4.1 Robustnější vyhledávač spojení

V současné době je jako vyhledávač spojení využívána aplikace IDOS. Data jsou však extrahována přímo z webové stránky, což je řešení velmi nerobustní, závislé na konkrétní struktuře stránky a pro komerční nasazení nepoužitelné. Ve hře jsou dvě možnosti, jak vyhledávat spojení efektivněji.

5.4.1.1 Navázání spolupráce s PMDP

Vzhledem k tomu, že mobilní aplikace Moje PMDP dokáže spojení vyhledávat pouze online (viz kapitola 3.1.4), je vysoce pravděpodobné, že aplikace bude komunikovat s vyhledávačem spojení umístěným na interním serveru společnosti skrz nějaké API (nebo jinými prostředky komunikace). Pokud by se s PMDP dokázala vyjednat určitá forma spolupráce, aplikace by kromě itineráře spojení mohla uživatelům poskytovat i informace o případném zpoždění spojů a další aktuální informace o změnách v dopravě. Osobně bych byl velmi rád, pokud bychom s protistranou

na přínosu shodli, protože by to vývoj aplikace značně ulehčilo a poskytované výsledky aplikace by byly relevantnější.

5.4.1.2 Návrh vlastního vyhledávače spojení

Úloha vyhledání spojení mezi dvěma libovolnými zastávkami by se dala převést na úlohu prohledávání grafu. Aplikace PMDP však dokáže preferovat i přesun pěšky, pokud je výhodnější než čekání na některý spoj. Tak či onak by se jednalo o poměrně komplikovanou úlohu, kterou už jistě musel řešit někdo před námi, a navíc bychom byli ochuzeni o aktuální informace o zpožděních a změnách provozu (viz kapitola 5.4.1.1). Jedinou výhodou takové implementace by bylo možné vyhledávání spojení i bez přístupu k internetu, framework SpeechCloud však stejně přístup k internetu vyžaduje. V případě zvolení této varianty bych zvažil navázání spolupráce s informatiky, kteří se těmito úlohami zabývají.

5.4.2 Vylepšení gramatik

Gramatiky tak, jak jsou popsány v kapitole 4.3.3, nejsou pro dialogový systém definované ideálním způsobem, jak jsem se dozvěděl po hlubším proniknutí do problematiky hlasových dialogových systémů. Poskytují totiž malou variabilitu promluv a neumožňují např. to, aby se hlasový agent ptal na informace, které uživatel ve své promluvě nezmínil. To bude třeba do budoucna změnit.

Stejně tak bude potřeba rozšířit i zadávání časů, aby systém nebyl omezen pouze na několik málo denních dob (gramatika na obrázku 5.3), ale aby bylo možné zadat jakýkoliv časový údaj v hodinách a minutách.

5.4.3 Vylepšení struktury dialogu

V současné době je struktura dialogu poměrně pevná a pro zdárné použití dialogového systému je nutné strukturu alespoň zběžně znát. Ideální hlasový dialogový systém by měl však být člověk schopen použít i v případě, že se s ním nikdy dříve nesetkal, zkrátka, jako kdyby jednal s člověkem. K tomuto ideálu bychom se v budoucnu chtěli alespoň přiblížit.

5.4.4 Rozdělení na zastávkovou a počítačovou (mobilní) část

V současné době se předpokládá, že aplikace by mohla být nasazena na některé ze zastávek MHD a tomu primárně odpovídá i vzhled a struktura uživatelského rozhraní. Nic však nebrání tomu, aby uživatel mohl hlasový terminál využívat i na svém chytrém telefonu, případně tabletu nebo počítači. K tomu je však třeba hlasový dialog modifikovat tak, aby se stal více univerzálním, resp. tak, aby nebyl omezen pouze na jedinou výchozí zastávku. S tím souvisí i rozšíření responzivnosti uživatelského rozhraní na různé velikosti a poměry obrazovek tak, aby byly ovládací prvky vždy snadno přístupné.

5.4.5 Náповěda a návod k použití

V současné době není v rámci uživatelského rozhraní poskytnuta žádná nápověda (kromě instrukcí souvisejících s ikonou mikrofonu) a novému uživateli tedy může trvat delší dobu, než se se systémem seznámí. Pomocť by mu přitom mohl buď přímo dialogový systém, pokud by se uživatel zeptal, jak jej správně používat. Dále přichází v úvahu i textová nápověda, např. ve formě vyskakovacího okna.

5.4.6 Alternativní možnosti zahájení dialogu

V současné době je systém používán především skrz osobní počítač a tomu je přizpůsobené i ovládání. Pokud by však byl systém skutečně umístěn na některé ze zastávek MHD, ovládacím prvkem by nemusely být pouze prvky uživatelského rozhraní, ale např. i senzor detekující přiblížení uživatele k hlasovému terminálu. V budoucnu bychom tedy chtěli tyto možnosti testovat, např. na populární platformě Raspberry Pi. Tento krok by byl klíčový pro nasazení hlasového terminálu na skutečnou zastávku.

5.4.7 Rozšíření funkcionality

V současné době je hlasový dialog orientován čistě na vyhledávání spojení či odjezdů. Některé aplikace popsané v kapitole 3.1 však poskytují i doplňující informace, například konečnou zastávku spoje, bezbarierové označení spoje či aktuální mimořádnosti v provozu a plánované výluky. Tyto informace mohou mít pro cestujícího zásadní vliv pro volbu konkrétního spojení, a proto by je měl být hlasový agent schopen poskytnout.

5.4.8 Vlastní hlas pro syntézu řeči

Přestože hlas použitý pro syntézu řeči je dostatečně srozumitelný, na někoho může stále působit příliš univerzálně nebo strojově. Jaké by to ale bylo, kdyby systém mluvil např. hlasem některé známé osobnosti? SpeechCloud umožňuje pro syntézu řeči použít i vlastní hlas natrénovaný s vhodných dat a bylo by při nejmenším zajímavé to vyzkoušet.

Rád bych do budoucna našel pro svůj dialogový systém profesionálního řečníka spojeného s městem Plzeň, např. někoho z plzeňského Divadla Josefa Kajetána Tyla. Otázka však zní, zda by s tím dotyčná osoba souhlasila, jestli by trvala na nějakém honoráři (pro kvalitní hlas je nutno nahrát desítky hodin anotovaných promluv) a zda by to pro uživatele bylo v konečném důsledku příjemnější, nebo by syntéza naopak zněla např. děsivě či nepřírozně. Tyto aspekty je potřeba při učinění finálního rozhodnutí ještě zvážit.

Výsledkem více než půlročního snažení na této práci se stala webová aplikace sloužící jako hlasový terminál pro vyhledávání spojení a odjezdů spojů městské hromadné dopravy v Plzni. Aplikace splňuje požadavky zadání práce a lze díky ní snadno a rychle vyhledat spojení na konkrétní zastávku či odjezdy spojů z aktuální zastávky, přičemž můžeme s aplikací komunikovat přirozenou řečí a v celých větách. Jsem si vědom některých nedostatků aplikace, které v práci popisuji. Ty bych chtěl napravit v rámci své diplomové práce a dovést aplikaci do stádia vývoje, které bude umožňovat široké nasazení.

Kromě samotné struktury a implementace hlasového dialogu jsem si pro korektní fungování aplikace potřeboval opatřit i příslušné datové zdroje. Potýkal jsem se s absencí snadno dostupných otevřených dat pro účely, které jsem v aplikaci hlasového terminálu zamýšlel. V rámci řešení této práce jsem si údaje obstaral díky strojovému čtení veřejně dostupné webové stránky, nicméně se jedná o řešení velmi provizorní a do budoucna bych chtěl data získávat jiným vhodným způsobem.

Během vypracování této práce jsem si uvědomil, že návrh hlasového dialogového systému, podobně jako návrh jakékoliv jiné aplikace, není jen otázkou technického provedení, ale je důležité si uvědomit, že aplikace má být primárně určena pro širokou škálu lidí všech věkových kategorií, kteří mají různé vzdělání či sociální postavení. Není tedy důležité aplikaci jen naprogramovat tak, aby dokázala splnit zamýšlené úlohy, ale je třeba klást důraz i intuitivnost použití, která u aplikace hlasového terminálu souvisí hlavně se strukturou dialogu a uživatelským rozhraním. I těmto aspektům se chci v rámci své budoucí práce věnovat.

Hlasové dialogové systémy se zatím rozvíjejí v rámci mobilních či domácích hlasových asistentů, velcí hráči na trhu s technologiemi však češtinu do značné míry ignorují. Proto považuji za důležité se tomuto výzkumnému odvětví věnovat i u nás v České republice, aby nebyla čeština z pole těchto technologií nadobro vytlačena. Nechť je výsledek této práce pro aplikaci češtiny v hlasových dialogových systémech možným příkladem.

Přílohy



A

A.1 Formulář pro uživatelské testování

Testování hlasového dialogového systému

Konkrétně se jedná o hlasový terminál pro MHD v Plzni.

Aplikace se nachází na webové stránce <https://cak.zcu.cz:9444/edu/bp/simio//index.html>



Aplikaci otevřete ve webovém prohlížeči na svém počítači, mobilu či tabletu.

Během plnění následujících úkolů si představte, že se nacházíte v Plzni na tramvajové zastávce **Technická**.

Úkoly

1. Vyhledejte nejbližší možné spojení na zastávku Opavská.
2. Vyhledejte večerní spojení na zastávku Macháčkova.
3. Vyhledejte nejbližší spojení na zastávku Sady Pětatřicátníků a po skončení odpovědi se doptejte, zda by šlo vyhledat spojení o půl hodiny později.
4. Vyhledejte nejbližší odjezd tramvaje z aktuální zastávky.
5. Vyhledejte odjezd v poledne.

Prohlédněte si tabulku pro vyhodnocení a запиšte příslušné údaje plynoucí z plnění úkolů.

Pokyny

1. Dialog zahájíte stisknutím tlačítka mikrofonu (nebo mezerníku). Rozpoznávání skončí samo.
2. Mluvte v celých větách a srozumitelně, avšak stále přirozeně.
3. Pokud se splnění úkolu nezdaří, zkuste větu formulovat jinak.
4. U 3. úkolu můžete upřesnění času spojení pronést (téměř) okamžitě po dokončení odpovědi (sledujte mikrofon a instrukce).

Vyhodnocení úkolů

Číslo úkolu	Podařilo se splnit? A/N	Potřebný počet pokusů	Přibližný čas plnění
1.			
2.			
3.			
4.			
5.			

Hodnocení: ohodnoťte aplikaci počtem hvězdiček (1 až 5)

Co se vám líbilo?

Co byste ještě vylepšili nebo udělali jinak?

Kontakt na vývojáře (náповěda)

E-mail: simio@students.zcu.cz

Mobil: 721 269 142

Bibliografie

- [Číž23] ČÍŽEK, Jakub. *České jízdní rády jsou konečně open data. Nikdo se v nich ale nevyzná*. 2023-04-30. Dostupné také z: <https://www.zive.cz/bleskovky/ceske-jizdni-rady-jsou-konecne-open-data-nikdo-se-v-nich-ale-nevyzna/sc-4-a-179546/default.aspx>.
- [HVT18] HANZLÍČEK, Zdeněk; VÍT, Jakub; TIHELKA, Daniel. WaveNet-Based Speech Synthesis Applied to Czech. In: SOJKA, Petr; HORÁK, Aleš; KOPÉČEK, Ivan; PALA, Karel (ed.). *Text, Speech, and Dialogue*. Cham: Springer International Publishing, 2018, s. 445–452. ISBN 978-3-030-00794-2.
- [Ouj23] OUJEZDSKÝ, Aleš. *Zpracování zvuku na počítači*. Ostravská univerzita v Ostravě, 2023-04-26. Dostupné také z: <https://digifolio.rvp.cz/artefact/file/download.php?file=73294&view=11595>.
- [Pav23] PAVLÍČEK, Radek. *Jak používají zrakově postižení uživatelé počítač nebo mobil?* 2023-04-28. Dostupné také z: <https://poslepu.cz/jak-pouzivaji-zrakove-postizeni-uzivatele-pocitac-nebo-mobil/>.
- [Pet23] PETER ACKLAND Serge Resnikoff, Rupert Bourne. World blindness and visual impairment: despite many successes, the problem is growing. 2023. Dostupné také z: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5820628/>.
- [Plz23] PLZEŇSKÉ MĚSTSKÉ DOPRAVNÍ PODNIKY, a.s. *Vozový park*. 2023-04-10. Dostupné také z: <https://www.pmdp.cz/o-nas/vozovy-park/>.
- [Psu+06] PSUTKA, J.; MÜLLER, L.; MATOUŠEK, J.; RADOVÁ, V. *Mluvíme s počítačem česky*. Prague: Academia, 2006. ISBN 80-200-1309-1. Dostupné také z: http://www.kky.zcu.cz/en/publications/PsutkaJ_2006_Mluvimes.
- [RS90] RAMSEY, William; STICH, Stephen. Connectionism and three levels of nativism. *Synthese*. 1990, roč. 82, s. 177–205.

- [SK17] SHARMA, Akanksha Rai; KAUSHIK, Pranav. Literature survey of statistical, deep and reinforcement learning in natural language processing. In: *2017 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, 2017, s. 350–354.
- [Šve23] ŠVEC, Jan. *Hlasové dialogové systémy (podklad k přednášce č. 7 předmětu KKY/HDS)*. Západočeská univerzita v Plzni, 2023.
- [úřa23] ÚŘAD, Český statistický. *Veřejná databáze VDB*. 2023-04-12. Dostupné také z: <https://vdb.czso.cz/vdbvo2/>.
- [Zhe+23] ZHENG, Ou; ABDEL-ATY, Mohamed; WANG, Dongdong; WANG, Zijin; DING, Shengxuan. ChatGPT is on the horizon: Could a large language model be all we need for Intelligent Transportation? *arXiv preprint arXiv:2303.05382*. 2023.

Seznam obrázků

2.1	Schéma hlasového dialogového systému, Zdroj: [Šve23] (upraveno) . . .	8
3.1	Ukázka zastávkových JŘ v papírové verzi (generováno na jizdnirady.pmdp.cz)	12
3.2	Zastávkové informační tabule	13
3.3	Webové rozhraní vyhledávače spojení PMDP (jizdnirady.pmdp.cz) . .	14
3.4	Uživatelské rozhraní aplikace Moje PMDP	15
3.5	Webové rozhraní aplikace IDOS (dostupná z idos.cz, odstraněny reklamy)	16
3.6	Webové rozhraní aplikace Jizdní řády Seznam.cz (dostupná z jizdnirady.seznam.cz)	17
4.1	Vývojový diagram hlasového terminálu	20
4.2	API dokumentace frameworku SpeechCloud	22
4.3	Ukázka gramatiky příkazů (upraveno, zkráceno)	26
5.1	Uživatelské rozhraní aplikace hlasového terminálu	30
5.2	Ikony mikrofону indikující stav dialogu	31
5.3	Gramatika pro specifikaci času spojení nebo odjezdu	31
5.4	Gramatika pro upřesnění času již vyhledaného spojení nebo odjezdu .	32

