



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV MIKROELEKTRONIKY

DEPARTMENT OF MICROELECTRONICS

## NÁVRH A IMPLEMENTACE INTERAKTIVNÍHO GRAFICKÉHO ROZHRAŇÍ PRO IVR

DESIGN AND IMPLEMENTATION OF INTERACTIVE GRAPHICAL INTERFACE FOR IVR

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Jakub Konečný

### VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Vladislav Škorpil, CSc.

BRNO 2020

# Diplomová práce

magisterský navazující studijní obor **Mikroelektronika**

Ústav mikroelektroniky

**Student:** Bc. Jakub Konečný

**ID:** 186114

**Ročník:** 2

**Akademický rok:** 2019/20

## NÁZEV TÉMATU:

### Návrh a implementace interaktivního grafického rozhraní pro IVR

## POKYNY PRO VYPRACOVÁNÍ:

Seznamte se a prostudujte použití funkce IVR ( automatická spojovatelka ) v telekomunikačních službách. Analyzujte stávající řešení, alternativy na trhu a zaměřte se na jejich realizaci v grafickém prostředí. Navrhněte webovou aplikaci jako modul komunikující s rozhraním PBX ústředny běžící na Asterisk verzi 13 pro uživatelsky jednoduchou implementaci IVR. Cílem práce bude plně funkční náhrada dosavadní koncepce a implementace aplikace do běžících systémů v produkci.

## DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

**Termín zadání:** 3.2.2020

**Termín odevzdání:** 1.6.2020

**Vedoucí práce:** doc. Ing. Vladislav Škorpil, CSc.

**doc. Ing. Lukáš Fucik, Ph.D.**  
předseda oborové rady

## UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Táto diplomová práca sa zaoberá vývojom grafického rozhrania pre vytváranie a editáciu IVR aplikácií. Práca je vo väčšej miere zameraná na softvérovú časť, analyzuje súčasné riešenie produktu iPBX od spoločnosti IPEX, popisuje použité technológie a predstavuje nový návrh interaktívneho systému spolu s generátorom Asterisk dial planu.

## **KLÚČOVÉ SLOVÁ**

IVR, PBX, IPEX, Asterisk, VoIP, call centrum, dial plan, React

## **ABSTRACT**

This diploma thesis focuses on development of graphical user interface for managing IVR applications. The work is more software oriented, it analyzes current state of the iPBX product belonging to the IPEX a.s. company, describes used technologies and introduces new concept of interactive user interface for generating IVR diagrams together with Asterisk dial plan generator.

## **KEYWORDS**

IVR, PBX, IPEX, Asterisk, VoIP, call center, dial plan, React

KONEČNÝ, J. *Návrh a implementace interaktivního grafického rozhraní pro IVR.*  
Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních  
technologií, Ústav mikroelektroniky. Diplomová práce. Vedúci práce Vladislav Škorpil.

# POĎAKOVANIE

Ďakujem vedúcemu diplomovej práce, doc. Ing. Vladislavovi Škorpilovi, CSc., za účinnú metodickú, pedagogickú a odbornú pomoc, konzultantovi Marekovi Červenkovi za cenné a profesionálne rady v oblasti VoIP technológií, kolegom Ing. Alešovi Janečkovi a Sebastiánovi Machovi za metodickú pomoc pri návrhu generátora dial planu, a spoločnosti IPEX za umožnenie realizovania tejto práce.

# OBSAH

<b>Úvod</b>	<b>6</b>
<b>1 Call centrum a technológia IVR</b>	<b>7</b>
1.1 PBX.....	7
1.1.1 IP PBX a VoIP .....	8
1.2 IVR.....	10
1.2.1 DTMF .....	11
1.2.2 Asterisk Dialplan .....	11
1.2.3 Nástroje pre implementáciu IVR .....	13
<b>2 Analýza súčasného riešenia</b>	<b>16</b>
2.1 Rozhranie iPBX .....	16
<b>3 Návrh grafického IVR</b>	<b>19</b>
3.1 Výber technológií .....	19
3.2 Popis grafického rozhrania .....	20
3.2.1 Popis implementovaných blokov .....	24
3.3 Optimalizácia výkonu .....	30
3.4 Implementácia kľúčových prvkov UX .....	33
3.5 Generátor dial planu.....	35
<b>4 Testovanie</b>	<b>40</b>
4.1 Testovanie blokov .....	40
4.2 Testovanie generátora dial planu .....	40
4.3 Testovanie kompletných scenárov .....	41
<b>Záver</b>	<b>45</b>
<b>Literatúra</b>	<b>46</b>
<b>Zoznam symbolov, veličín a skratiek</b>	<b>47</b>
<b>Zoznam obrázkov</b>	<b>48</b>
<b>Zoznam tabuliek</b>	<b>49</b>
<b>A Koncept grafického rozhrania</b>	<b>50</b>
A.1 Základné zobrazenie pracovnej plochy.....	50

A.2	Mód vytvárania prepojení .....	51
A.3	Editácia bloku .....	52
A.4	Výber hlások pre blok DTMF .....	53
A.5	Uloženie aplikácie.....	54

# ÚVOD

Vynález telefónu bol bez pochyb jeden z najväčších míľnikov v histórii komunikácie. Komunikácia prostredníctvom telefónu otvorila brány do sveta biznisu veľkým firmám a evolúcia spôsobila, že call centrá, slúžiace výlučne ako predajný nástroj, sa stali neoddeliteľnou súčasťou zákazníckej podpory. V dnešnej dobe moderné call centrum pozostáva z viacerých prvkov, prostredníctvom ktorých spoločnosti usilujú o čo najväčší komfort, či už pri poskytnutí podpory existujúcim klientom, alebo o konverziu nových zákazníkov.

Cieľom diplomovej práce je navrhnúť a zrealizovať grafické rozhranie pre službu IVR (hlasový asistent), ktorej úlohou je automatizovať smerovanie hovorov, poprípade vykonávať akcie v závislosti na užívateľskom klávesovom vstupe. Celkový projekt je realizovaný ako modernizácia aktuálneho riešenia pre telekomunikačnú spoločnosť IPEX. V diplomovej práci sú popísané aktuálne použité technológie, na ktoré postupne nadväzuje návrh užívateľsky prívetivej webovej aplikácie, prostredníctvom ktorej bude možné vytvárať a vizualizovať jednotlivé IVR diagramy.



# 1 CALL CENTRUM A TECHNOLOGIA IVR

Technologický vývoj ovplyvnil každú oblasť nášho života. Inak to nebolo ani v prípade hlasových služieb a call centier ako ich dnes poznáme. Patent bostonského profesora Alexandra Grahama Bella v podobe funkčného telefónneho prístroja v roku 1876 sa stal stavebným kameňom pre jeden z najpoužívanejších komunikačných kanálov v histórii – telefonický hovor.

Napriek tomu, že slovné spojenie „call centrum“ bolo použité prvý krát až v roku 1983, systémy využívajúce podobné princípy existovali už v 50-tych rokoch 20. storočia. Jedným z nich bol práve systém nazývaný ACD alebo Automatic Call Distributor, ktorý už podľa napovedajúceho názvu slúži na smerovanie prichádzajúcich hovorov konkrétnym agentom. Stratégia smerovania týchto hovorov je založená na sade pokynov, ktoré diktujú ako majú byť prichádzajúce hovory spracované a smerované. Často sú tieto pravidlá založené na jednoduchom princípe smerovať hovor na akéhokoľvek agenta v čo najkratšom časovom intervale, ale zvyčajne do procesu zasahuje viacero premenných s cieľom zistiť, prečo zákazník volá. Smerovanie tisícok hovorov správnym agentom bolo veľmi komplikované, preto vznikol priestor pre vývoj systémov akými sú napríklad IVR.

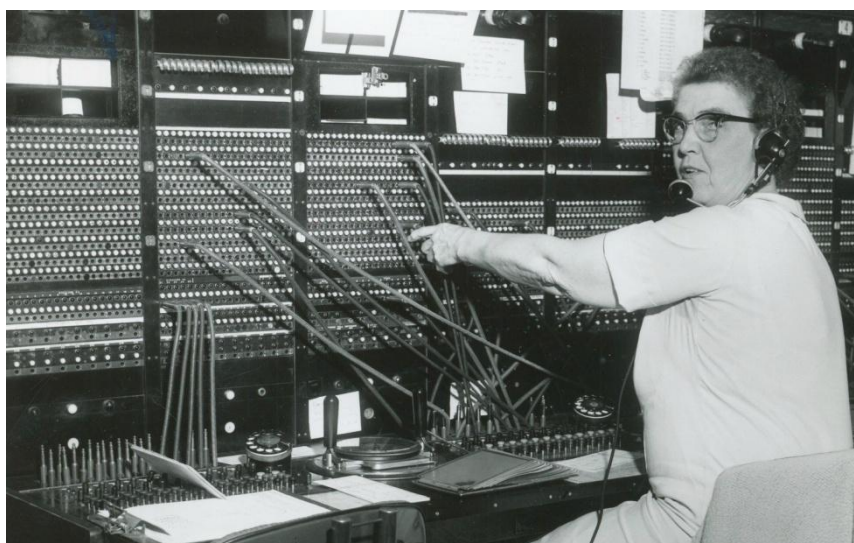
V nasledujúcich podkapitolách bude uvedený bližší popis jedných zo základných prvkov a technológií z oblasti moderných call centier súčasnosti [1, 2].

## 1.1 PBX

Private Branch Exchange (súkromná pobočková ústredňa) vznikla reakciou na potrebu väčších firiem minimalizovať svoje náklady za hovory do verejných sietí. Úlohou tohto zariadenia je zjednotiť telefónnu sieť podniku do jedného vstupno-výstupného bloku. Užívatelia telefónneho systému PBX môžu komunikovať interne (v rámci svojej spoločnosti) alebo externe (s okolitým svetom) prostredníctvom komunikačných kanálov ako sú VoIP, ISDN a analógova linka. PBX systémy sú v dnešnej dobe k dispozícii ako hostované alebo virtuálne riešenia, alebo ako on-premise riešenia fungujúce na vlastnom hardware [3].

### PABX vs PBX

Na počiatku vzniku prvých call centier, pojem PBX predstavoval miestnosť plnú operátorov obsluhujúcich manuálne prepojavacie dosky (switchboardy), kde prepojenie hovoru na konkrétnu linku bolo realizované prostredníctvom kabeláže. S vývojom technológie prišli aj nové vylepšenia, ktoré zahŕňali elektronické prepínanie. Z PBX sa vyvinul termín PABX (Private Automatic Branch Exchange) – automatizovaná pobočková ústredňa. V dnešnej dobe sú však všetky pobočkové ústredne automatizované, nie je nutné rozlišovať medzi PBX a PABX a tieto pojmy sú navzájom zameniteľné.[3]

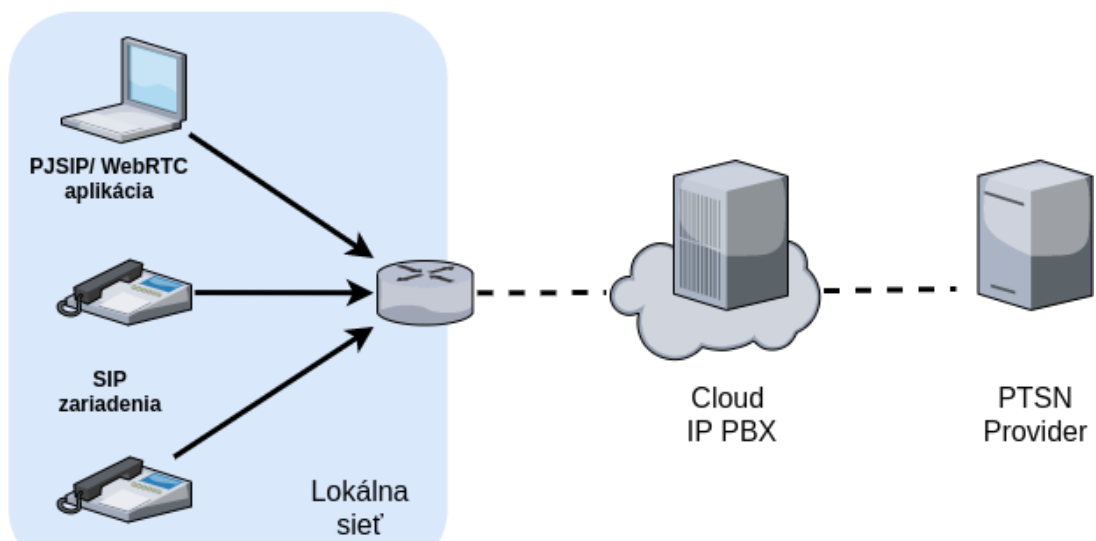


Obr. 1.1 Operátorka switchboardu v roku 1971 (prevzaté z <https://www.northcountrypublicradio.org>)

### 1.1.1 IP PBX a VoIP

Vzhľadom na to, že najstaršie PBX systémy boli analógové stroje, ktoré boli pripojené k verejnej sieti prostredníctvom medených liniek, realizácia a implementácia takýchto ústrední trvala dlhší čas a vyžadovala nutnosť spolupráce s technikmi telefónnych spoločností.

Moderné riešenia súčasnosti využívajú vo väčšej miere protokol VoIP (Voice over Internet Protocol), ktorý funguje na báze prenosu digitalizovaného hlasu prostredníctvom protokolov TCP/UDP/IP, čo znamená, že konečnému používateľovi postačuje vysokorýchlostné pripojenie k internetu aby mohol používať hlasové, dátové a video služby. Ústredne v tomto prípade môžu predstavovať cloudové servery na ktorých beží software zabezpečujúci prepájanie telefonických liniek. Súčasťou infraštruktúry môžu byť aj tzv. VoIP brány umožňujúce prepojenie VoIP sietí s analógovými linkami PTSN [3].



Obr. 1.2 Infraštruktúra možného riešenia VoIP (IP PBX)

Medzi hlavné výhody VoIP patria:

- **cena** – prenos hlasu prostredníctvom internetu minimalizuje náklady za diaľkové hovory, ktoré tak nemusia prechádzať sieťami klasických operátorov,
- **prenosnosť** – operátori sú schopní pracovať aj mimo pracoviska, jedinou potrebou je pripojenie k internetu a VoIP zariadenie,
- **všestrannosť** – široký sortiment využitia od priameho volania z PC na PC cez podporu služieb akými sú konferenčné hovory, presmerovanie, čakajúce hovory a veľa ďalších.

Nevýhody :

- **pripojenie k internetu** – šírka prenosového pásma internetového pripojenia ovplyvňuje kvalitu hovorov,
- **závislosť na napájaní zo siete** – pri výpadku elektrickej energie je VoIP infraštruktúra ochromená a telefonovanie nefunguje, preto sa odporúča použitie záložných napájacích zdrojov,
- **tiesňové volania** – výhoda mobility pri používaní VoIP zariadení má vplyv na správnu lokalizáciu hovoru pri volaní v núdzi.

Trend technológie VoIP sa odzrkadlil aj na zvýšenom dopyte po cloudových riešeniach. Prevádzkovatelia call centier sú tak odbremenení od zložitých a často zdĺhavých implementácií lokálnych ústrední a v prípade problémov sa môžu spoľahnúť na technickú podporu poskytovateľov virtuálnych PBX. Na českom trhu v tomto segmente figurujú firmy ako IPEX, Daktela, Canistec a ďalší. Jadrom virtuálnych ústrední je softvér Asterisk a ich implementácia sa pohybuje v rádoch desiatok minút [4].

## Asterisk

Asterisk je open-source framework určený pre platformy Linux a Unix vyvinutý firmou Digium, ktorý obsahuje širokú škálu funkcií dostupných v komerčných a proprietárnych systémoch PBX. Okrem pobočkovej ústredne Asterisk implementuje funkcie, akými sú napríklad: VoIP brána, softvérová ústredňa (softswitch), konferenčný server, hlasová schránka či hlasové menu. Rozlišujeme LTS (Long Time Support) a štandardné verzie, ktoré sa líšia na základe dĺžky doby podpory. LTS verzie sú plne podporované po dobu štyroch rokov a nasledujúcim jedným rokom určeným pre bezpečnostné opravy. Štandardné verzie sú podporované minimálne jeden rok a nasledujúcim jedným rokom určeným pre vydania bezpečnostných opráv. V tab. 1.1 je prehľad všetkých vydaných verzií, zahrňujúci aj tie, ktorým už natrvalo skončila podpora (tzv. EOL – End Of Life) [5,6].

Tab. 1.1 Prehľad vydaných verzií Asterisk [6]

Verzia	Typ vydania	Dátum vydania	Bezpečnostné opravy	Koniec podpory
1.2.x		2005-11-21	2007-08-07	2010-11-21
1.2.x	LTS	2006-12-23	2011-04-21	2012-04-21
1.6.0.x	Standard	2008-10-01	2010-05-01	2010-10-01
1.6.1.x	Standard	2009-04-27	2010-05-01	2011-04-27
1.6.2.x	Standard	2009-12-18	2011-04-21	2012-04-21
1.8.x	LTS	2010-10-21	2014-10-21	2015-10-21
10.x	Standard	2011-12-15	2012-12-15	2013-12-15
11.x	LTS	2012-10-25	2016-10-25	2017-10-25
12.x	Standard	2013-12-20	2014-12-20	2015-12-20
13.x	LTS	2014-10-24	2020-10-24	2021-10-24
14.x	Standard	2016-09-26	2017-09-26	2018-09-26
15.x	Standard	2017-10-03	2018-10-03	2019-10-03
16.x	LTS	2018-10-09	2022-10-09	2023-10-09
17.x	Standard	2019-10-28	2020-10-28	2021-10-28

Alternatívne platformy: OpenPBX, sipXpbx, YATE, SEMS, OpenSER.

## 1.2 IVR

IVR (Interactive Voice Response), automatická spojovateľka alebo hlasové asistent je pomenovanie pre známu aplikáciu, ktorá vznikla v 80-tych rokoch minulého storočia, nastavila trend moderného call centra a stala sa tak jeho neoddeliteľnou súčasťou. V princípe je to hlasový systém, ktorý je v interakcii s volajúcim prostredníctvom číslícovej klávesnice a umožňuje poskytnúť informácie bez nutnosti priamej komunikácie s operátorom, prepojiť hovor do vhodnej fronty, poprípade vykonávať množstvo ďalších automatizačných akcií. Prvé reálne využitie systému IVR sa našlo v bankovom sektore, kedy zákazník mohol zavolaním na bezplatnú linku zistiť finančný stav jeho účtu, poprípade požiadať o históriu transakcií.

Príkladov systémov IVR existuje celá rada a záleží na konkrétnych požiadavkách zákazníka, preto sú systémy vždy vyvíjane na mieru, aby prínos pre firmu bol maximálny. V mnohých prípadoch je vhodné integrovať IVR so systémami CRM (Customer Relationship Management). To umožňuje s prichádzajúcim hovorom dohľadať užitočné informácie o volajúcom a poskytnúť ich telefónnemu operátorovi [7, 8].

### 1.2.1 DTMF

Ako bolo vyššie spomenuté, interakcia medzi volajúcim a volanou linkou je realizovaná pomocou číslícovej klávesnice. Tónová voľba alebo DTMF (Dual-tone Multi Frequency) predstavuje tón zložený z 2 sínusových signálov o presne danej frekvencii. Po prehratí hlasového menu volajúci zvolí hodnotu alebo kombináciu hodnôt na klasickej telefónnej klávesnici so znakmi 0–9, \* a #, a zariadenie vygeneruje štandardne 50 ms dlhé pípnutie o 2 tónoch s príslušnými frekvenciami.

Tab. 1.2 Prehľad frekvenčných párov pre jednotlivé DTMF voľby [9]

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

V tab. 1.2 je uvedený prehľad jednotlivých DTFM volieb a ich frekvencií. Pre príklad, po stlačení klávesy 9 je vygenerovaný tón, ktorý vznikne súčtom sínusových signálov s frekvenciou 1477 Hz a 852 Hz. Tóny A až D sa na bežnej telefónnej klávesnici nevyskytujú. Nazývajú sa systémové a používajú sa pre špeciálne účely, napr. programovanie ústrední [9].

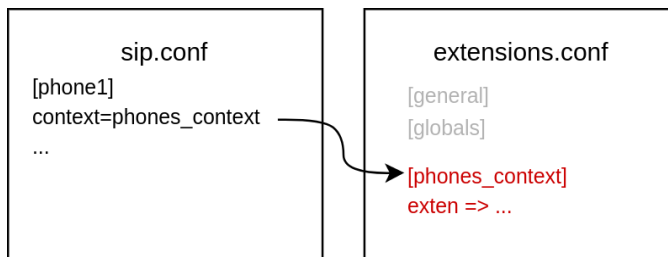
### 1.2.2 Asterisk Dialplan

Programovanie jednotlivých IVR systémov je realizované v prostredí softvéru Asterisk pomocou aplikácie, tzv. dial plan. Dial plan predstavuje niečo ako skriptovací jazyk špecifický pre Asterisk, ktorý popisuje ako bude prebiehať manipulácia s prichádzajúcimi hovormi do ústredne.

Hlavný konfiguračný súbor pre dial plan - *extensions.conf* sa nachádza v adresári */etc/asterisk*. Po modifikovaní dial planu je nutné spustiť reload modulu, k čomu slúži v Asterisk CLI (Command Line Interface) príkaz *diplan reload*. Ten zabezpečí, že nová konfigurácia nastavených IVR aplikácií bude načítaná bez obmedzenia ostatných služieb PBX. Formát dial planu je organizovaný to sekcií, ktoré sú označované ako kontexty. Tie udržujú jednotlivé časti dial planu na sebe nezávislé, čo umožňuje oddeliť funkcionality sekcií, poprípade použiť rôzne druhy služieb rozdielnym skupinám užívateľov. Kontext uvádzame v hranatých zátvorkách - [] a na začiatku dial planu sú definované 2 špeciálne kontexty - *[general]* a *[globals]*, kde sa nachádzajú položky globálnej konfigurácie (napr. globálne premenné). Všetky nasledujúce kontexty

však môžu mať ľubovoľný názov.

Pri konfigurovaní SIP zariadení (väčšinou v súbore `/etc/asterisk/sip.conf`) je jeden z povinných parametrov zariadenia `context`. Kontext je v tomto prípade bod, v ktorom hovor vstupuje do dial planu. Na obr. 1.3 je znázornenie prepojenie konfigurácie zariadenia s kontextom dial planu.



Obr. 1.3 Vzťah medzi konfiguráciou zariadení a kontextom dial planu

V nasledujúcej časti kódu je naprogramovaná jednoduchá IVR aplikácia, ktorej účelom je po zavolaní na linku 100 prehrať hlásku:

```
[prehrani_hlasky]
exten => 100, 1, Answer()
exten => 100, n, Playback(/var/lib/asterisk/sounds/cz/auth-thankyou)
exten => 100, n, Return
```

V prvom riadku definujeme kontext aplikácie. Riadky nasledujúce za definíciou kontextu nesú označenie `extensions` a predstavujú súbor inštrukcií, ktoré Asterisk sekvenčne vykonáva po zavolaní klapky. Syntax riadku je nasledovná:

```
exten => číslo,priorita,aplikácia([parameter[,parameter2...]])
```

Dôležitým parametrom je priorita. IVR plán začína vždy prioritou 1 a pokračuje v riadkoch s prioritou o jedna väčšou. Poradie riadkov nie je podstatné (avšak vysoko sa odporúča radenie podľa postupnosti priorít, vzhľadom na prehľadnosť), dôležitá je iba postupnosť priorít.

Posledným z parametrov v riadku je aplikácia – funkcia, ktorá sa vykoná. Zoznam všetkých podporovaných funkcií je možné vypísať príkazom v Asterisk CLI:

```
CLI> core show applications
```

alebo ich je možné nájsť na adrese:

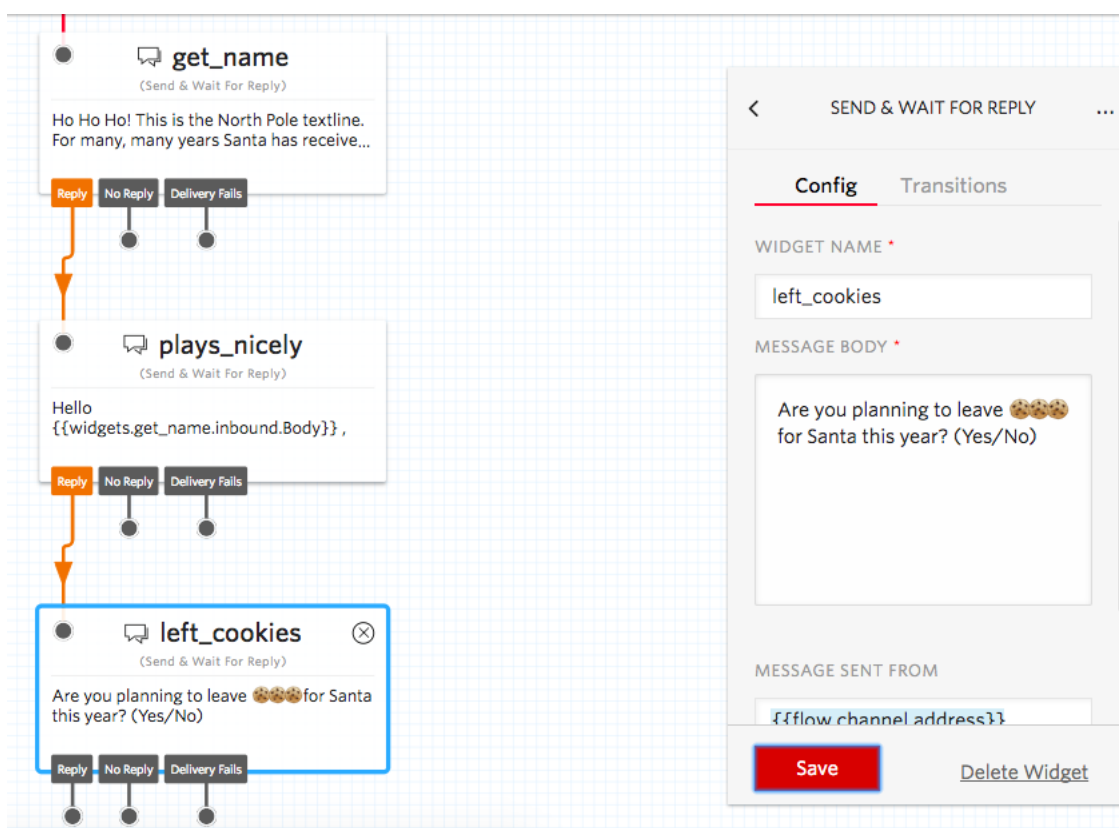
<https://wiki.asterisk.org/wiki/display/AST/Asterisk+13+Dialplan+Functions> [10].

### 1.2.3 Nástroje pre implementáciu IVR

Konfigurácia IVR aplikácií môže byť v mnohých prípadoch veľmi zložitá. Pre uľahčenie práce s ich vytváraním boli navrhnuté grafické prostredia, ktoré ponúkajú intuitívne zobrazenie aplikácií a poskytujú užívateľsky prívetivý spôsob úprav, bez nutnej znalosti Asterisk dial planu.

#### Twilio Studio

Twilio, spoločnosť poskytujúca služby v segmente cloud komunikácií, disponuje nástrojom Twilio Studio. Prostredie aplikácie je založené na webových technológiách, čo eliminuje problémy súvisiace s kompatibilitou pre odlišné platformy operačných systémov ako je to v prípade natívnych aplikácií.

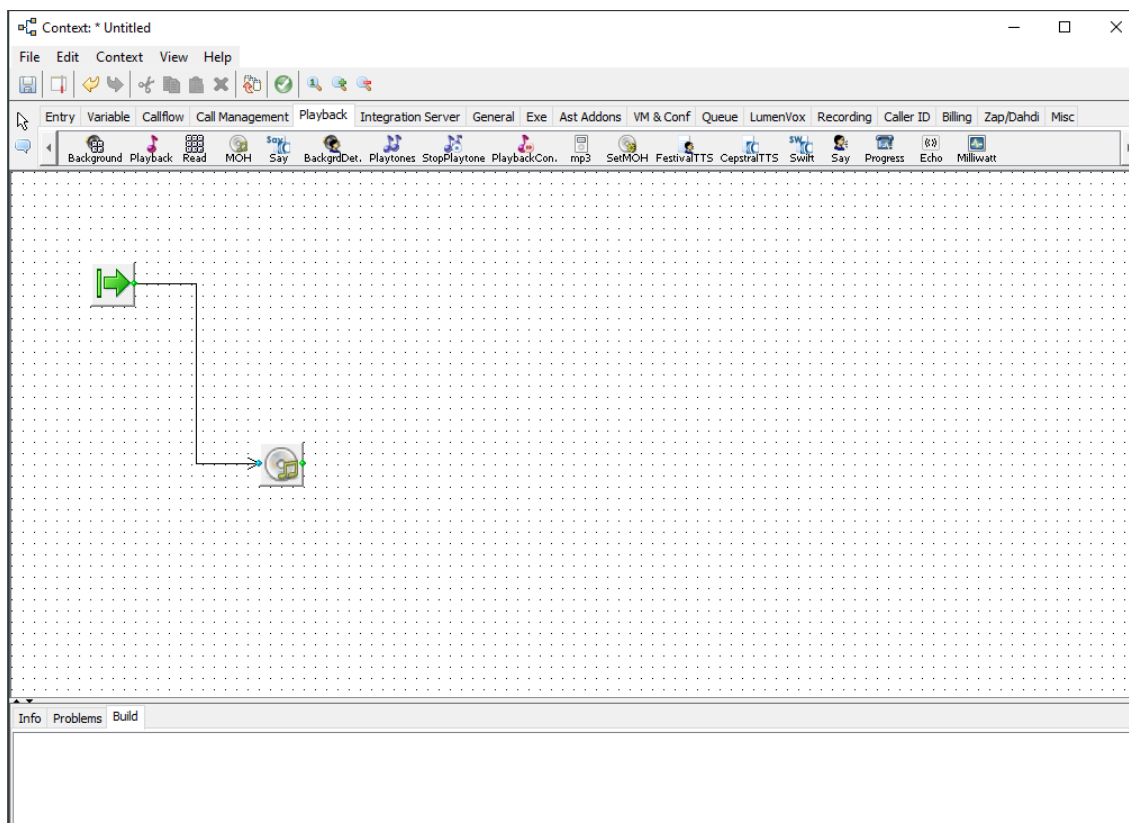


Obr. 1.4 Grafické rozhranie Twilio Studio (prevzaté z [11])

Na obr. 1.4 je zobrazené grafické prostredie Twilio Studio. Užívateľ má na výber z konfigurovateľných blokov, ktoré sú medzi sebou prepájané a vytvárajú tak schému toku hovoru (Twilio túto schému definuje pojmom *flow*). Po opustení webovej stránky sú rozpracované schémy (*flows*) uchované a po opätovnom prihlásení je možné v ich úpravách ďalej pokračovať. Podobnými alternatívami sú aj riešenia od firiem: Inference, Telitics, a iné [11].

## Visual Dialplan

Visual Dialplan je natívna aplikácia vyvinutá firmou Apstel určená pre platformy Windows a Linux. Po spustení sa pripája priamo na Asterisk server a umožňuje vytváranie, editáciu a testovanie IVR. Na obr. 1.5 môžeme vidieť grafické rozhranie založené na princípe point-and-click prepojovacích blokov.



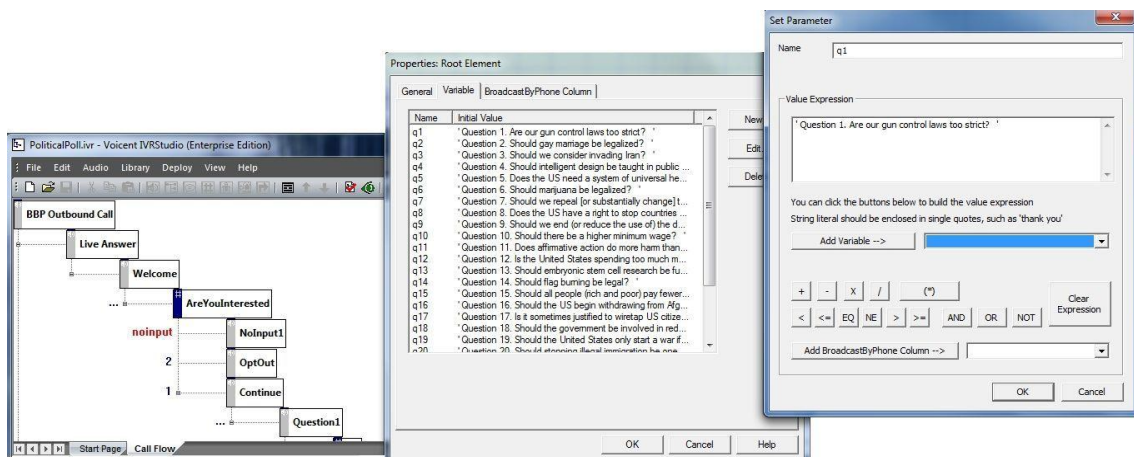
Obr. 1.5 Grafické rozhranie Visual Dialplan Professional

Vývoj aplikácie Visual Dialplan je momentálne ukončený a vlastníkom softvéru už nie je firma Apstel. Skúšobnú verziu je však stále možné stiahnuť z oficiálnych stránok: <https://www.apstel.com/visual-dialplan-professional> [12].

## Voicent IVR Studio

Profesionálnym riešením pre pokročilé konfigurácie IVR systémov je softvér od spoločnosti Voicent, IVR Studio. Aplikácia je určená pre platformy Windows a cena za retail verziu sa pohybuje v intervale od \$799 (1 hlasový kanál) do \$9850 (100 kanálov) [13].





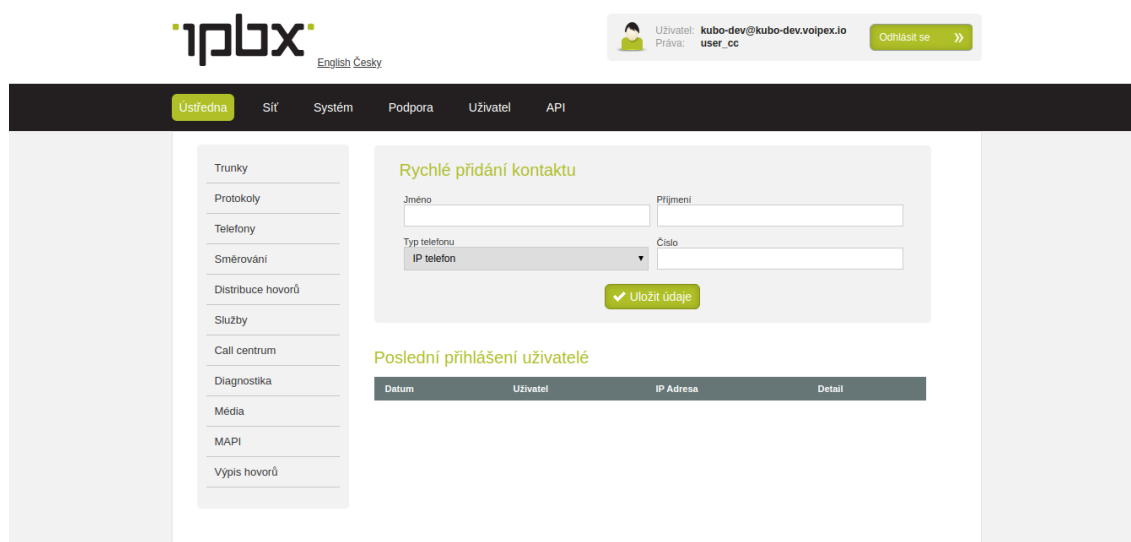
Obr. 1.6 Grafické rozhranie Voicent IVR Studio (prevzaté z [13])

## 2 ANALÝZA SÚČASNÉHO RIEŠENIA

V nasledujúcej kapitole bude popísaná analýza aktuálneho riešenia pre vytváranie IVR aplikácií, ktoré poskytuje produkt virtuálnych ústrední iPBX od spoločnosti IPEX, fungujúcich na jadre Asterisk verzie 13. Predstavený bude koncept novej interaktívnej webovej aplikácie, generátora Asterisk dial planu a odôvodnený bude aj výber použitých technológií.

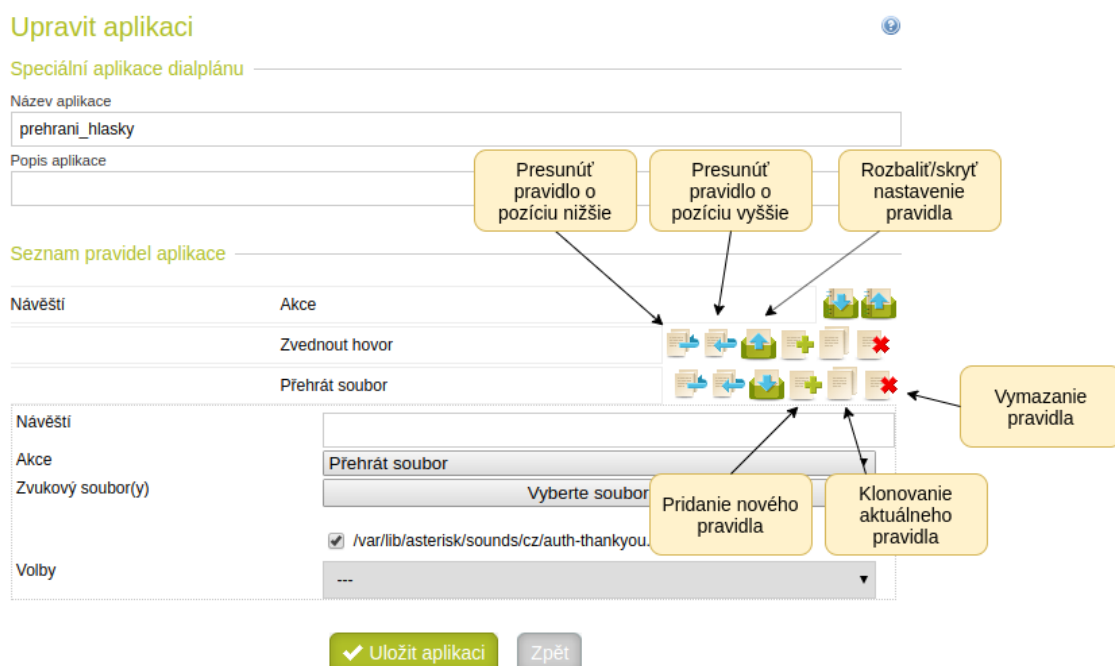
### 2.1 Rozhranie iPBX

Po úspešnej implementácii novej virtuálnej ústredne iPBX má zákazník k dispozícii administratívne rozhranie, v ktorom prebieha dodatočná konfigurácia liniek, zariadení, smerovania či iných nastavení týkajúcich sa správy samotnej ústredne a call centra (viď obr. 2.1).



Obr. 2.1 Administratívne rozhranie iPBX ústredne

Súčasťou rozhrania je aj prostredie pre realizáciu IVR aplikácií. V súčasnosti je tvorba IVR založená na systéme vytvorenia sekvenčného poradia jednotlivých operácií, ktoré sa uskutočnia po vytočení konkrétnej linky. Prostredníctvom definovania návěstí pre jednotlivé operácie a skokov na určitú pozíciu je možné vytvárať logické slučky, čo je priamou analógiou k tomu ako fungujú kontexty v Asterisk dial plane.



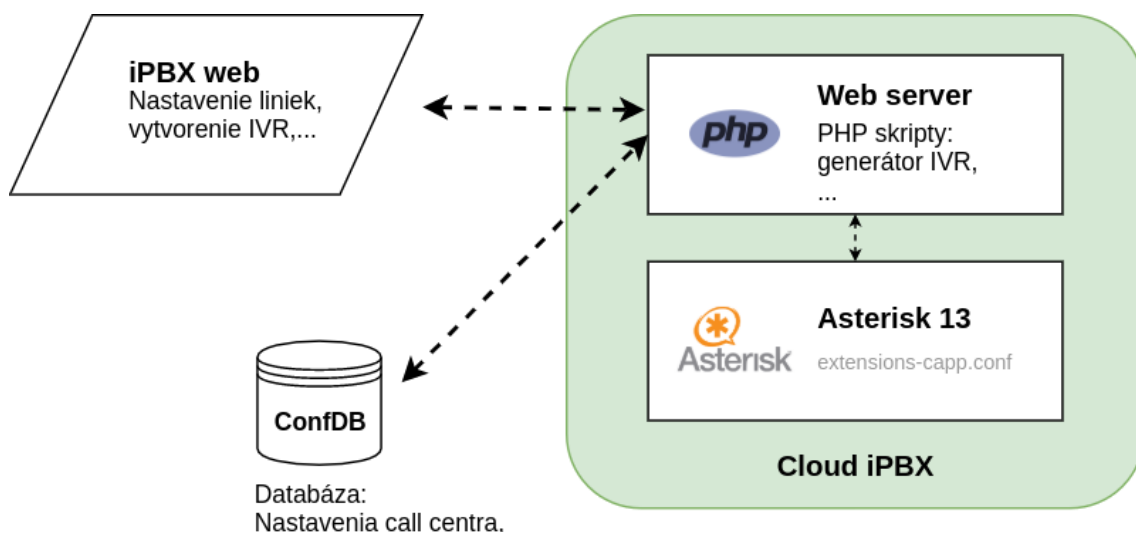
Obr. 2.2 Popis prostredia konfigurácie IVR aplikácie v systéme iPBX

Na obr. 2.2 je nakonfigurovaná aplikácia, ktorá bola použitá v ukážke kódu dial planu v kapitole 1.2.2.

Úpravy zložitejších IVR systémov bývajú často krát komplikované. Implementátor musí pri ich editácii disponovať znalosťou systému a malá zmena niektorých z pravidiel alebo návěstí môže viesť k nechceným následkom v podobe nesprávneho prepojenia hovoru, ukončenia hovoru, alebo v horšom prípade k iniciovaniu spojenia mimo lokálnu sieť.

Web ústredne iPBX je naprogramovaný v prostredí skriptovacieho jazyka PHP. Jazyk PHP patrí do skupiny tzv. server-side programovacích jazykov, čo znamená, že celková logika zobrazenia využíva výpočtový výkon servera a klientovi sa zobrazujú len statické dáta. Výhodou tohto riešenia je možnosť priameho prístupu do súborovej štruktúry operačného systému a tým pádom aj ku konfiguračným súborom Asterisk.

Po uložení aplikácie je štruktúra dát uložená do relačnej databázy. Z tej nakoniec generátor dial planu vygeneruje súbor *extensions-capp.conf*. Nevýhodou aktuálneho riešenia je, že pri upravení jednej IVR aplikácie je nutné znovu regenerovať celý konfiguračný súbor, čo zvyšuje režiu prenosu dát medzi serverom a databázou. Na obr. 2.3 je znázornená schéma komunikácie webového rozhrania s virtuálnou ústredňou.



Obr. 2.3 Schéma komunikácie medzi administratívnym rozhraním a virtuálnou ústredňou

## 3 NÁVRH GRAFICKÉHO IVR

Pri návrhu grafického riešenia bol kladený dôraz na faktor užívateľskej prívetivosti. Inšpiráciou preto bolo overené a obľúbené Twillio Studio, ktoré umožňuje presúvať jednotlivé logické bloky a následne vytvárať tok hovoru ich prepojeniami. V nasledujúcich podkapitolách budú predstavené použité technológie, grafický koncept aplikácie a vysvetlený bude nový princíp generovania dial planu z blokovej štruktúry.

### 3.1 Výber technológií

Jedným z technologických cieľov bolo úplne nahradiť doterajšiu logiku naprogramovanú v jazyku PHP (webové rozhranie + serverové skripty) programovacím jazykom JavaScript (ďalej iba JS). Flexibilitnosť JS v dnešnej dobe umožňuje nielen vytvárať dynamické webové stránky ale aj plne funkčné serverové aplikácie.

#### React

Knižnica pre JS vyvinutá spoločnosťou Facebook si za posledné 3 roky získala extrémnu popularitu v oblasti vývoja dynamických webových aplikácií, k čomu stále prispieva rastúca komunita vývojárov. Už z názvu je postrehnuteľné, že React bol vyvíjaný pre „reaktívne“ aplikácie, s vysokou mierou dynamiky a interakcie. Hlavnou úlohou tejto knižnice je uľahčiť prácu s tzv. DOM (Document Object Model), čo je objektovo orientovaná reprezentácia XML a HTML dokumentov. Vo výsledku to znamená, že celková práca s dynamickým zobrazovaním HTML elementov je pre programátora vo väčšej miere skrytá a dôraz je kladený na biznis logiku aplikácie. Veľkou výhodou je využitie component-based architektúry, ktorej podstatou je rozdelenie UI do menších logických celkov, komponentov, znovu použiteľnými naprieč aplikáciou. React dokáže pracovať s komponentmi veľmi efektívne. Porovnáva nový stav aplikácie so stavom predchádzajúcim a prekresľuje iba tie komponenty, v ktorých došlo ku zmenám. Na obr. 3.1 je znázornené členenie React aplikácie na komponenty.

```
<App>
  <NavBar />
  <Header title="ReactJS Academy" />
  <BookFilter />
  <BookList />
  <Footer />
</App>
```

Obr. 3.1 Príklad členenia štruktúry React.js aplikácie na komponenty

#### IndexedDB & Dexie.js

Pre implementáciu automatického ukladania na pozadí bolo zvolené použitie natívnej technológie webových prehliadačov, a to klientskej databázy IndexedDB. Svojimi výhodami, ako sú podpora indexovania, uchovávanie dátových štruktúr

v podobe objektov a asynchrónna API, prevažuje nad často obľúbenejšou alternatívou lokálneho klientskeho úložiska - localStorage. Najväčšou výhodou IndexedDB je už vyššie spomenuté asynchrónne programovacie rozhranie. Rýchle zmeny užívateľského rozhrania nie sú blokované a zápis do databázy prebieha kvázi na pozadí. To umožňuje plynulé presúvanie blokov prostredníctvom myši bez trhania a zásekov. Pre pohodlnejšiu prácu s API IndexedDB bola zvolená externá JS knižnica *dexie* (<https://www.npmjs.com/package/dexie>).

### Node.js

Node.js je softvérové prostredie určené pre programovanie škálovateľných serverových aplikácií založené na jadre Chrome V8. Použitím obľúbeného programovacieho jazyka JavaScript sa tak postupne zmenšuje tzv. knowledge gap (znalosťná medzera) medzi frontend a backend programátormi. Majorita serverových systémov spoločnosti IPEX už momentálne spoľahlivo funguje práve na tejto technológii.

### MySQL

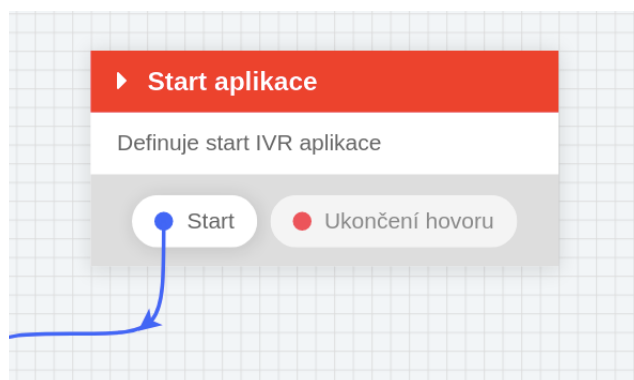
Databázový systém relačného typu je používaný ako úložisko pre nakonfigurované IVR aplikácie. Táto technológia zostáva naďalej zachovaná, úpravami však prešli niektoré tabuľky vzhľadom na kompatibilitu s novými dátovými štruktúrami. Bližší popis úprav je mimo rozsah tejto diplomovej práce.

## 3.2 Popis grafického rozhrania

V nasledujúcej kapitole sa budú vo väčšej miere vyskytovať pojmy ako sú blok, prepojenie a trigger. Pre lepšie pochopenie problematiky je nutné si tieto pojmy najskôr definovať.

### Blok

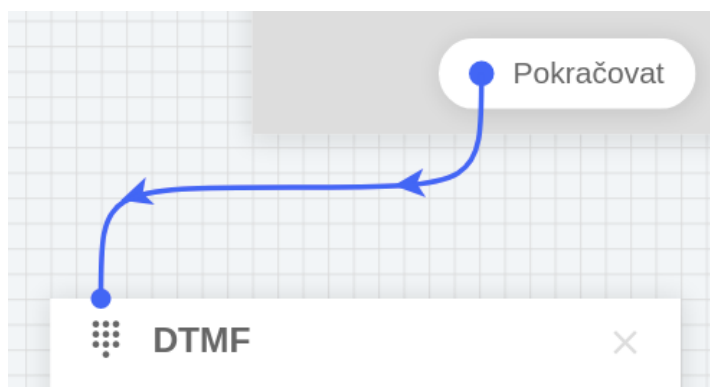
Základná stavebná jednotka IVR diagramu. Každý blok reprezentuje jednu alebo viacero akcií ktoré sa vykonajú po smerovaní toku hovoru cez tento blok. V súčasnom stave je definovaných 34 blokov. Bližší popis blokov je uvedený v nasledujúcej podkapitole. Reprezentácia bloku je znázornená na obrázku obr. 3.2.



Obr. 3.2 Blok Start aplikace - vstupný bod IVR aplikácie

## Prepojenie

Smerovanie toku hovoru cez aplikáciu je realizované prostredníctvom prepojení. Na základe splnených podmienok a vykonaných akcií tok hovoru kopíruje jednotlivé cesty prepojení. Východiskom prepojenia je vždy trigger a cieľom ďalší blok. Pravidlom pri ich vytváraní je, že do jedného bloku môže byť pripojených viacero prepojení, kde naopak z východiska, triggeru, vždy smeruje iba jedno aktívne prepojenie. Všetky krivky prepojení obsahujú šípky (viď obr. 3.3). Tie slúžia pre lepšie identifikovanie smeru toku hovoru.



Obr. 3.3 Vzájomné prepojenie blokov IVR aplikácie

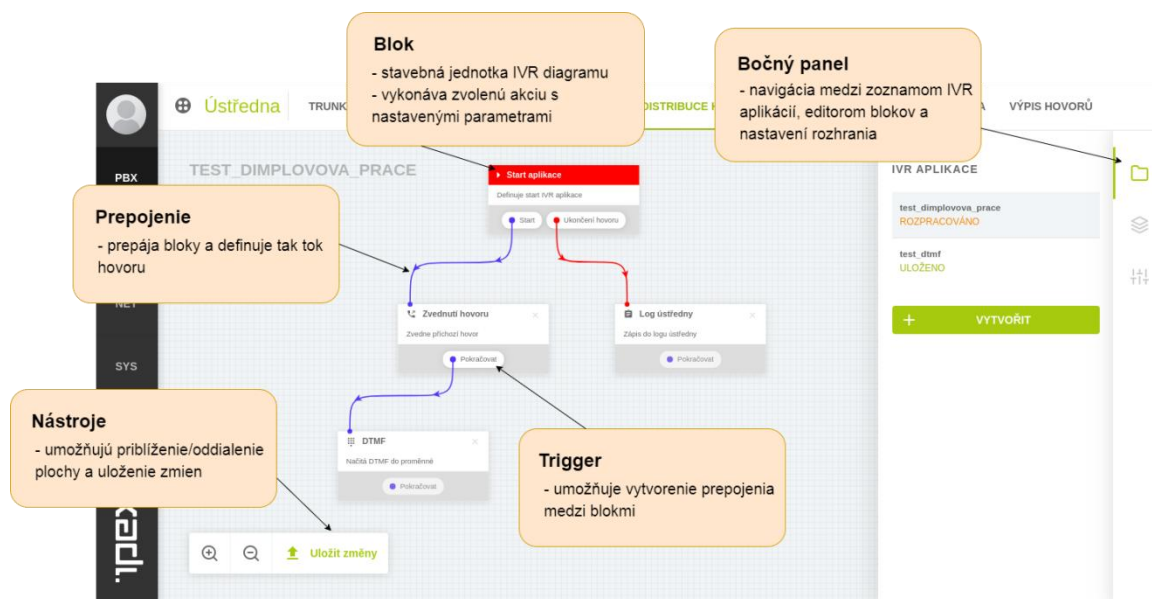
## Trigger

Trigger, pojem, ktorý by sa vo voľnom preklade preložil ako „spúšťač“, definuje východisko pre vytvorenie nových vzájomných prepojení blokov. Väčšina blokov obsahuje práve jeden trigger s názvom „Pokračovať“, ktorý signalizuje, že akcia súvisiaca s typom bloku bola vykonaná a tok hovoru pokračuje ďalej. Bloky obsahujúce viacero triggerov signalizujú, že práve v nich dochádza k vetveniu hovoru.

## Pracovná plocha

Základom nového rozhrania je plocha pre vytvorenie IVR diagramu. Predstavuje niečo ako pracovnú dosku, na ktorej užívateľ vytvára z blokov logickú a funkčnú schému toku hovoru. Plochu je možné približovať a oddaľovať, či už prostredníctvom kolieska myši alebo tlačidla lupy. Drag and drop funkcionality umožňujú presúvanie a organizáciu jednotlivých blokov po ploche.

Na obr. 3.4 je zobrazené grafické rozhranie s popismi predstavených častí.



Obr. 3.4 Popis grafického rozhrania IVR

Vytváranie nových prepojení je realizované v dvoch krokoch. V prvom kroku, kliknutím na trigger aktivujeme mód prepojenia a následným kliknutím na iný blok vytvoríme medzi nimi väzbu. Pre väčšiu efektívnosť bola implementovaná funkcia skupinového kopírovania a zmazávania blokov prostredníctvom klávesových skratiek Ctrl + C, Ctrl + V a klávesy Delete. Výber viacerých blokov je možný pridržením klávesy CTRL a následným kliknutím na konkrétne bloky.

Pravé bočné menu umožňuje navigáciu medzi zoznamom nakonfigurovaných IVR aplikácií, editorom blokov a všeobecnými nastaveniami. Pri zvolení jednej z položiek je rozšírené menu vysunuté s obsahom pre danú možnosť. Všeobecné nastavenia v aktuálnej verzii obsahujú iba jednu možnosť, a to vypnutie/zapnutie mriežky. Poskytujú tak priestor pre budúce užívateľské nastavenia.

Zoznam IVR poskytuje rýchly a stručný prehľad nakonfigurovaných aplikácií s informáciou o tom, či je v stave neuložených zmien, teda „Rozpracováno“, alebo je nahraná na server, vid' obr. 3.5.



## IVR APLIKACE

test\_diplomova\_prace  
ROZPRACOVÁNO

test\_ipex  
ULOŽENO

TTS  
ULOŽENO



VYTVORIŤ

Obr. 3.5 Zoznam nakonfigurovaných IVR aplikácií

Kliknutím na tlačidlo „Vytvoriť“ je otvorené modálne okno s formulárom (obr. 3.6), v ktorom užívateľ zadá názov a popis aplikácie, a môže pokračovať v konfigurácii nového IVR diagramu.

Nová IVR aplikace

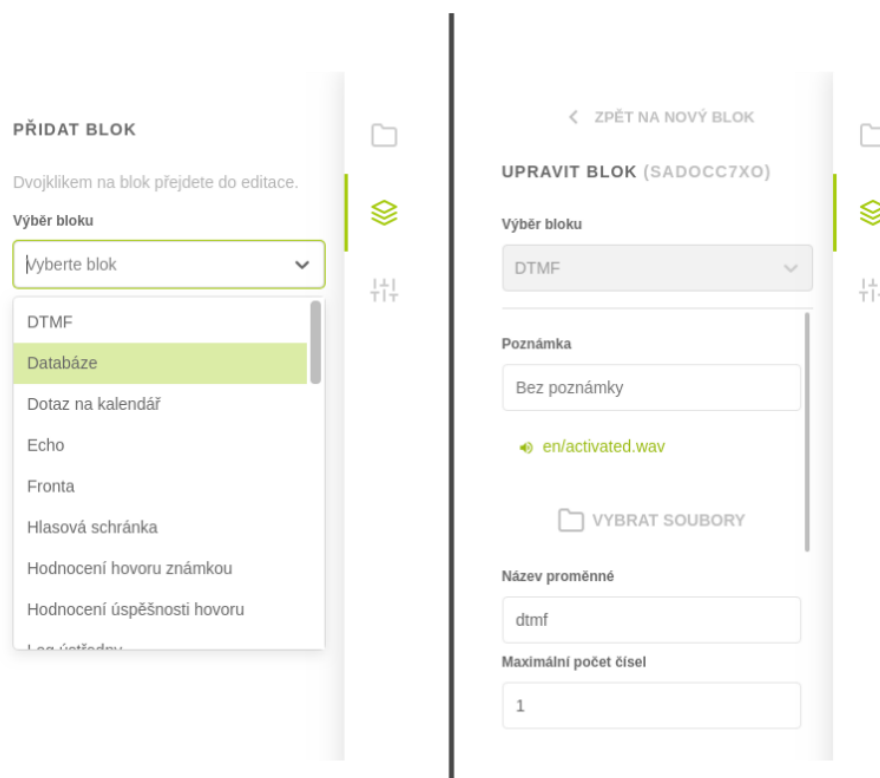
Název aplikace  
Zadejte název

Popis aplikace  
Bez popisu

+ VYTVORIŤ

Obr. 3.6 Modálne okno pre vytvorenie novej aplikácie

Pridávanie a editácia blokov prebieha spoločne v prostredí editora blokov (druhá položka v menu). Editačný mód je aktivovaný dvojklikom na zvolený blok. Vysunutím bočného panelu je k dispozícii formulár s poľami pre konfiguráciu daného bloku (viď obr. 3.7)



Obr. 3.7 Editor blokov - pridanie nového bloku (vľavo), editácia pôvodného bloku (vpravo)

Po finálnych úpravách je možné aplikovať tieto zmeny a odoslať ich na server pomocou tlačidla „Uložit zmeny“ v paneli nástrojov v ľavom dolnom rohu (viď obr. 3.4). Toto tlačidlo plní zároveň dve funkcie: nahratie novej aplikácie na server a aktualizovanie pôvodnej v prípade jej editácie.

### 3.2.1 Popis implementovaných blokov

Zoznam akcií (a tým pádom aj blokov) v IVR aplikácii prešiel menšou reštrukturalizáciou. Akcie ako „Skok na pozici“ alebo „Skok na návěšti po ukončení hovoru“ boli úplne odstránené, pretože v novom prostredí pojmy ako „skok“ už nedávajú zmysel. Niektoré akcie boli zlúčené do jedného komplexnejšieho bloku. Vo finále vzniklo z celkového počtu 57 akcií 35 blokov v novom grafickom rozhraní. V tejto kapitole bude predstavený finálny zoznam implementovaných blokov a stručne uvedená ich funkcionality.

#### Start aplikace

Blok „Start aplikace“ je automaticky vygenerovaný po vytvorení novej IVR aplikácie a slúži ako vstupná brána, z ktorej užívateľ definuje ďalšie prepojenia. Okrem toho spracováva akciu ukončenia hovoru. Špecialitou bloku je, že ho nie je možné vymazať.

## **Čekat**

Ako vyplýva z názvu, funkciou bloku je čakať po určitú dobu, po ktorej bude tok hovoru znova pokračovať do ďalšieho bloku. Počas doby čakania je v konfigurácii na výber možnosť prehrávať hudbu na pozadí.

## **Databáze**

Pridaním tohto bloku je umožnený zápis alebo načítanie hodnôt z databázy do premennej dial planu. Hlavným dôvodom využitia tohto bloku môže byť zdieľanie hodnoty v databáze naprieč viacerými IVR aplikáciami.

## **Dotaz na kalendár**

Blok umožňuje získanie telefónneho čísla z najbližšej udalosti v kalendári, napr. kvôli automatickému presmerovaniu na agenta, ktorý má momentálne službu.

## **DTMF**

Použitím tohto bloku je užívateľovi prehraná zvolená hláska, po ktorej sa čaká na zvolenie voľby na klávesnici. Zvolená hodnota je uložená do premennej a je k dispozícii k načítaniu v nasledujúcich častiach IVR diagramu. Na základe jej hodnoty je potom možné prepojiť hovor do rôznych front, poprípade iným spôsobom smerovať tok hovoru.

## **Echo**

Zvyčajne sa využíva pre testovanie alebo overenie funkčnosti zvukových periférií. Zvuk odosielaný mikrofónom užívateľa prechádza ústredňou a je spätne odosielaný volajúcemu.

## **Fronta**

V tomto bloku je hovor presmerovaný do jednej z aktuálne nakonfigurovaných front. Každá fronta zväčša pozostáva z operátorov špecializovaných na určitú náplň práce. Okrem výberu fronty zo zoznamu je možné načítať názov fronty z premennej.

## **Hlasová schránka**

Presmerovaním hovoru do zvolenej hlasovej schránky je odoslaný email na priradenú adresu s hlasovou nahrávkou. Pri konfigurácii je možné vypnúť alebo zapnúť prehratie úvodnej hlásky.

## **Hodnocení hovoru známkou**

Umožňuje volajúcemu oznámkovať operátora, ktorý sa venoval jeho požiadavkám. Známkovanie prebieha na stupnici od 1 do 5 (pri známkovaní ako v škole) a výstup s priemerom známok je uvedený v štatistikách call centra.

## **Hodnocení úspěšnosti hovoru**

Slúži pre evidenciu, či bola požiadavka z pohľadu volajúceho úspešne vybavená. Výstup v podobe formátu „ANO“ alebo „NE“ je potom uvedený v štatistikách call centra.

## **Log ústředny**

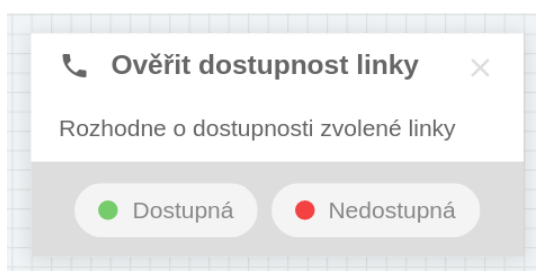
Používa sa pre diagnostiku a monitorovanie priebehu hovoru. Akciou tohto bloku je zapísanie vlastného záznamu do databázy logov. Logy je možné sledovať cez rozhranie diagnostiky v administrácii ústredne.

### Nastavit proměnnou

V tomto bloku je možné nastaviť hodnotu do užívateľsky definovanej premennej. Táto premenná je platná len v tej aplikácii, v ktorej bola nastavená. Pri konfigurovaní bloku má implementátor možnosť vybrať jednu z najčastejšie používaných preddefinovaných premenných alebo uviesť vlastný názov.

### Ověřit dostupnost linky

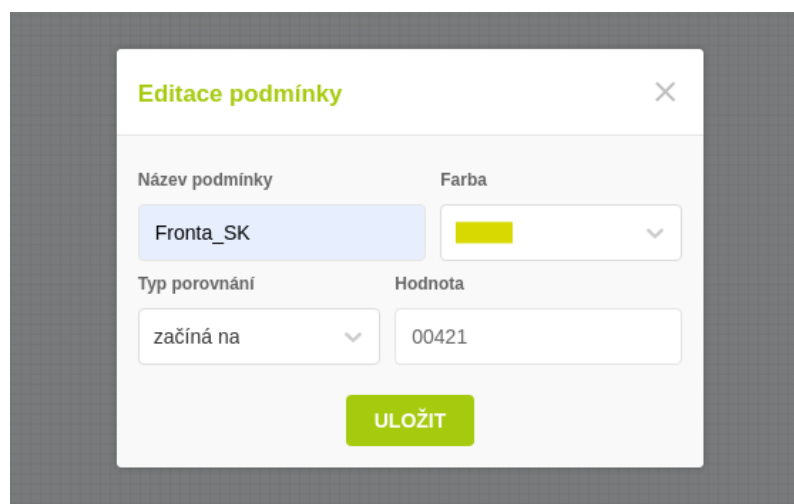
Výberom linky je otestovaná jej dostupnosť. Ak je linka dostupná, tok hovoru pokračuje triggerom „Dostupná“, v opačnom prípade triggerom „Nedostupná“ (viď obr. 3.8).



Obr. 3.8 Ukážka bloku „Ověření dostupnosti linky“

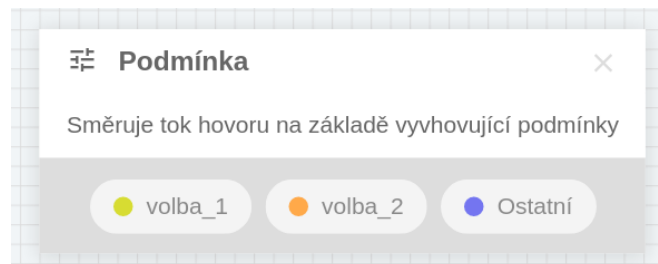
### Podmínka

Ako jediný z blokov umožňuje užívateľovi smerovať tok hovoru na základe vlastných vytvorených podmienok, čo sa vo veľkej miere využíva napríklad pri porovnávaní stlačených číslíc (DTMF) alebo porovnávaní prefixov zahraničných čísel.



Obr. 3.9 Vytvorenie novej podmienky pre vetvenie toku hovoru

Vo formulári konfigurácie tohto bloku užívateľ definuje vlastné podmienky pre sledovanú hodnotu premennej (obr. 3.9). V prípade, že nedôjde k naplneniu ani jednej zo zvolených podmienok, tok hovoru pokračuje triggerom „Ostatní“, viď obr. 3.10.



Obr. 3.10 Ukážka bloku „Podmínka“

### **Poslat email**

Po konfigurácii parametrov, je akciou tohto bloku odoslanie emailu s vlastným telom správy na zvolenú emailovú adresu.

### **Poslat SMS**

Analogicky k predchádzajúcemu bloku, blok „Poslat SMS“ odošle SMS správu prostredníctvom IPEX SOAP brány. Pre túto funkciu musí byť aktivovaná SMS služba v administračnom rozhraní ústredne.

### **Přečíst**

Blok slúži k prečítaniu hodnoty z premennej volajúceho. Pri konfigurácii má implementátor na výber 4 spôsoby prečítania hodnoty, a to sú „Číslo“, „Číslovka“, „Datum“ a „Čas“.

### **Přečíst z textu**

Funkciou tohto bloku je prekonvertovať zvolený text do podoby audio súboru vo formáte mp3, ktorý je následne prehraný volajúceho. Pre syntézu textu bolo zvolené použitie Google Cloud Text-to-Speech API.

### **Přehrát hlásku**

Ústredňa prehrá vybraný zvukový súbor, hlásku, volajúceho. V konfigurácii je možné zvoliť či bude hláska prehrávaná na pozadí. V opačnom prípade je pri prehrávaní stále detekované stlačenie klávesnice (DTMF kód).

### **REST klient**

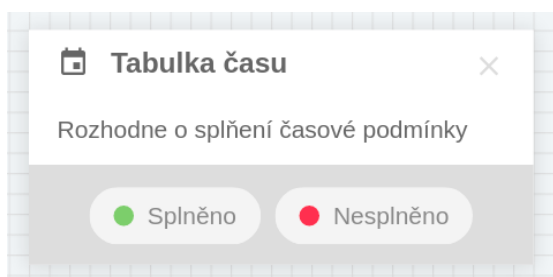
Blok vytvorí HTTP požiadavku pre zvolenú adresu serveru. Po nastavení formátu odpovede je možné načítať návratovú hodnotu v zákazníckej aplikácii z premennej *web\_query*. Táto funkcia môže slúžiť napríklad pre informovanie externých systémov CRM o prichádzajúcom hovore.

### **Signalizace**

Akciou bloku je odoslanie SIP kódu na stranu volajúceho. Na výber je medzi signalizáciou SIP Progress (SIP 183 Session Progress), signalizáciou obsadeného kanálu (SIP 486 Busy Here) alebo signalizáciou vyzváňania (SIP 180 Ringing).

### Tabulka času

Na základe zvolenej časovej tabuliek rozhoduje, či aktuálny čas vyhovuje časovému úseku v tabuľke. V prípade že je podmienka splnená, tok hovoru pokračuje triggerom „Splnené“, v opačnom prípade triggerom „Nesplnené“ (obr. 3.11).



Obr. 3.11 Ukážka bloku „Tabulka času“

### SOAP klient

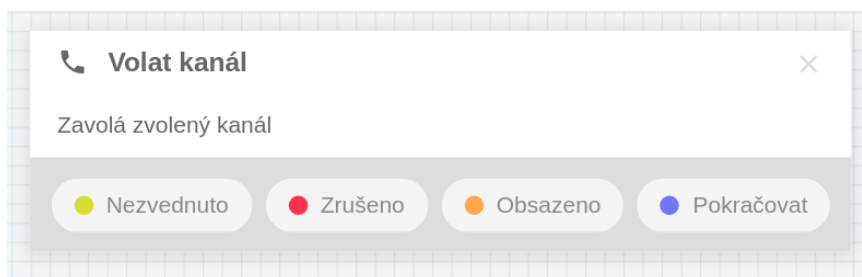
Umožňuje vytvorenia pripojenia k SOAP (Simple Object Access Protocol) serveru prostredníctvom autentizácie HTTP Basic alebo NTLM (NT LAN Manager), ktorá je využívaná v Microsoft Windows systémoch.

### Typ vyzvánění

Definuje použitý typ vyzvánania pre odchádzajúci hovor prostredníctvom hlavičky Alert-Info v SIP INVITE správe. Jednotlivé typy vyzvánania sa líšia dĺžkou páuz medzi pípnutiami. Na výber je k dispozícii typ Bellcore-dr1 až Bellcore-dr5.

### Volat kanál

Blok slúži pre zavolanie konkrétneho SIP kanálu. V prípade neúspechu obsluhuje stavy ako „Nezvednuto“, „Zrušeno“ alebo „Obsazeno“, vid' obr. 3.12.



Obr. 3.12 Ukážka bloku „Volat kanál“

### Vytočiť kanál a poslat DTMF

Zavolá zvolený SIP kanál odošle DTMF tón. Konfigurácia podporuje nastavenie časového oneskorenia. Táto funkcionlita môže byť použitá napr. pre otvorenie brány prostredníctvom elektronického vrátnika.

### **Volat linku**

Umožňuje volanie jednej alebo viacerých liniek a obsluhuje stavy v prípade nedostupnosti. Pri nastavení je možné zapnúť periodické pípnutie, ktoré je po určitej dobe prehrané ako volajúcemu, tak aj volanému. Nastavenie doby pre pípnutie je takisto možné definovať.

### **Volat linku dle proměnné**

Číslo linky je v tomto prípade načítané zo zadanej premennej, ktorá bola nastavená v predchádzajúcich krokoch (napr. prostredníctvom bloku „DTMF“) a umožňuje obsluhu stavov v prípade neúspechu.

### **Volat posledního agenta**

Na základe telefónneho čísla je hovor prepojený na agenta, ktorý bol v kontakte s volajúcim ako posledný. Predpokladom teda je, že volaný agent bude mať skúsenosť s riešením požiadavky zákazníka a nebude nutné popisovať situáciu znovu (napr. pri hlásení technického problému).

### **Vyhledat agenty**

Funkciou tohto bloku je nájdenie agenta priradeného ku kontaktu na základe volajúceho telefónneho čísla. Ak kontakt existuje a obsahuje parametre *agent1* a *agent2*, v ktorých sú uvedené čísla ich liniek, tieto hodnoty sa uložia do premenných IVR aplikácie.

### **Volat vyhledané agenty**

Postupné zavolanie vyhládaných agentov pre dané telefónne číslo. Ak sa hovor nepodarí spojiť s prvým agentom, IVR aplikácia pokračuje volaním druhého. Ak sa nepodarí spojiť ani s jedným z agentov, hovor môže pokračovať do obecného call centra.

### **Volat výstupní směrování**

Nasmeruje prichádzajúci hovor do vybraného výstupného smerovania.

### **Záznam hovoru**

Prechodom aplikácie na tento blok je spustený alebo ukončený záznam hovoru. Užívateľovi je umožnené pri nastavovaní bloku zvoliť rôzne úrovne hlasitosti záznamu.

### **Zavěsit hovor**

Ukončenie hovoru. Aj tento blok obsahuje trigger „Pokračovať“ nakoľko IVR diagram môže pokračovať aj po ukončení hovoru, napr. zápisom dát do databázy, zalogovaním potrebných parametrov, atď.

### **Zvednout hovor**

Slúži na zdvihnutie prichádzajúceho hovoru. V konfigurácii je možné nastaviť dobu čakania pred zdvihnutím.

### 3.3 Optimalizácia výkonu

Pri procese návrhu frontend API bolo potrebné sústrediť sa na optimalizáciu dátových štruktúr a zabezpečiť tak dobrú škálovateľnosť riešenia, čo do veľkosti a komplexnosti vytváraných diagramov, tak aj do ďalšej rozšíriteľnosti novými funkcionalitami v budúcnosti. V tejto podkapitole budú predstavené metódy, ktoré tieto optimalizácie sprevádzali.

#### Objekty a polia

Z pohľadu interpretácie dát sú jednými z najpoužívanejších dátových štruktúr v prostredí JS objekty a polia. Tieto dátové typy zdieľajú spoločný základ podobný asociatívnym poliam ako ich poznáme z objektovo orientovaných jazykov, teda štruktúram zložených z dvojíc kľúč-hodnota.

```
const blocksArray = [
  {
    id: 'PPBqWA9',
    type: 'SMS',
    label: 'Poslat SMS',
    description: 'Odešle SMS zprávu zadanému příjemci',
    triggers: [...],
    ...
  }
];

const blocksMap = {
  'PPBqWA9': {
    id: 'PPBqWA9',
    type: 'SMS',
    label: 'Poslat SMS',
    description: 'Odešle SMS zprávu zadanému příjemci',
    triggers: [...],
    ...
  }
}
```

Obr. 3.13 Porovnanie dátových štruktúr – pole a mapa

Na obr. 3.13 sú pre porovnanie znázornené štruktúry dát nakonfigurovaných blokov v globálnom stave IVR aplikácie počas jej vytvárania a editácie. V prvom prípade sa jedná o pole objektov. Pri potrebe získania dát z bloku alebo jeho editácie je nutné iteračne toto pole prehľadávať. Vzhľadom na veľkosť a zložitosť bežných IVR diagramov by to bol značný problém. Preto je pre naše účely efektívnejší druhý typ dátovej štruktúry, tzv. mapa. V tomto prípade sú jednotlivé bloky adresované na základne kľúču, ktorým je unikátne vygenerované ID tohto bloku.





Obr. 3.14 Test efektívnosti dátových štruktúr (pole vs. mapa)

Na obr. 3.14 je vyhodnotenie testovania výkonu jednotlivých blokov JavaScript kódu webovou službou JSBench (<https://jsbench.me>). Ako môžeme vidieť, dátová štruktúra mapy, adresovaná prostredníctvom unikátneho ID bloku, je v prípade vyhľadávania efektívnejšia.

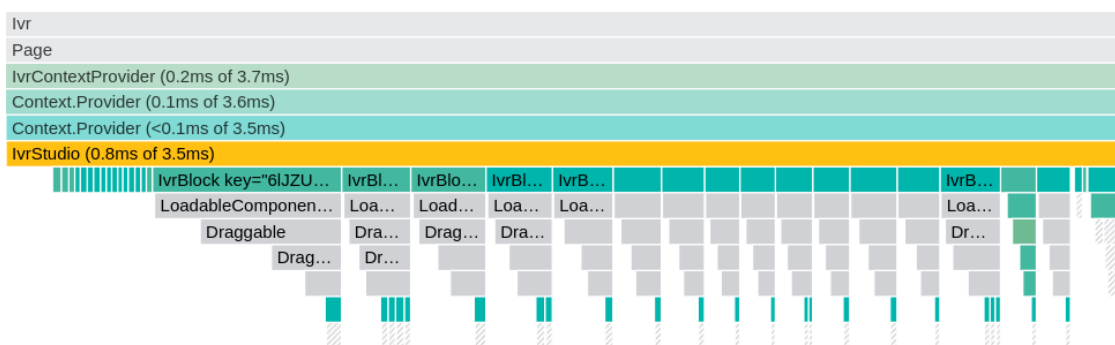
### State management

Management stavu aplikácie je dôležitým faktorom pri návrhu architektúry komplexnejších riešení. V tomto prípade je stav aplikácie riadený pomocou React Context API. Najvyšší, tzv. root komponent, plní funkciu providera dát pre všetky zanorené komponenty. Tie majú priamy prístup ku globálnemu stavu aplikácie a k funkciám, prostredníctvom ktorých je tento stav modifikovaný. Toto riešenie razantne prispieva ku kvalite kódu čo do prehľadnosti a členenia, avšak bez určitých optimalizácií by došlo časom k problémom s výkonom.



Na obr. 3.16 vidíme profil všetkých prekreslených komponentov na stránke (zeleno zvýraznené). Vyvolanie tohto stavu bolo spôsobené označením bloku kliknutím, čo malo za následok aktualizovanie globálneho stavu aplikácie a prekreslenie všetkých zanorených komponentov. Toto správanie nie je ideálne, a aby sa týmto prípadom predišlo, výstupy komponentov kritických pre výpočtový výkon (filtrovanie, hľadanie v poliach s väčším počtom prvkov) boli zaobalené vstavanou metódou knižnice React, *useMemo()*.

Po vykonaní úprav a spätnej analýze v okne React Profiler bolo docielené značné zníženie počtu zbytočných prekreslení, a tým zvýšeniu výkonu (obr. 3.17).



Obr. 3.17 Profil prekreslených React komponentov po optimalizácii

Vyššie predstavené metódy umožnili, aby podvozok aplikácie bol stabilný a neprichádzalo tak k výrazným poklesom výkonu pri dynamických operáciách s väčším počtom blokov (presúvanie po ploche, vytváranie prepojení). Vznikol tak priestor pre budúcu implementáciu ďalších prvkov pre vylepšenie a zjednodušenie užívateľského prostredia.

### 3.4 Implementácia kľúčových prvkov UX

V ďalšom kroku sa sústredíme na implementáciu prvkov ovplyvňujúcich užívateľský zážitok, tzv. user experience (UX). Medzi tieto prvky patria v prvom rade funkcionality ako „drag and drop“ a „zoom and pan“.

#### Drag and drop

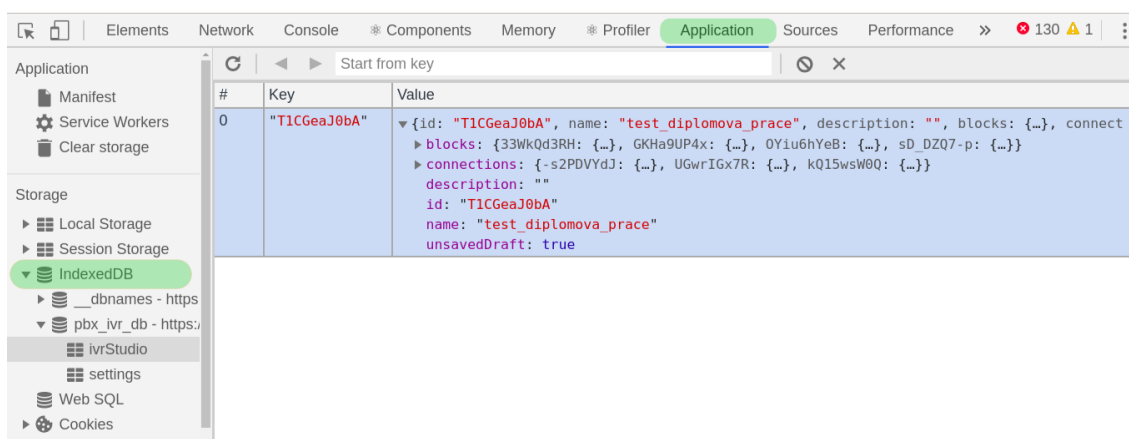
Metóda presúvania prvkov prostredníctvom myši po ploche je známa hlavne z prostredí CAD softvérov. V rozhraní IVR je táto funkcionality aplikovaná na bloky. Použitá knižnica *react-draggable* (<https://www.npmjs.com/package/react-draggable>) zabezpečuje logiku presúvania HTML elementov prostredníctvom natívnej API webového prehliadača.

## Zoom and pan

Kombináciu s drag and drop dotvára obľúbená funkcionálna posúvanie pracovnej plochy ťahom myši alebo jej približovanie/odďaľovanie kolieskom myši. V tomto prípade nebola použitá žiadna knižnica tretích strán. Priblíženie plochy simuluje parameter v CSS - *transform*, s nastavením *scale()* a *translate()* a je riadené natívnym *wheel* eventom. Posúvanie pracovnej plochy rovnako využíva natívne eventy Web API, ako sú *mousedown*, *mousemove* a *mouseup*.

## Ukladanie dát na pozadí

Pri konfigurovaní IVR aplikácie je každá zmena v dátovej štruktúre automaticky na pozadí ukladaná do internej databázy webového prehliadača, IndexedDB. Pri opustení stránky tak nepríde ku strate údajov, poprípade môže byť v procese vytvárania aplikácie umožnené pokračovať v konfigurácii neskôr.



Obr. 3.18 Prístup k IndexedDB z rozhrania Chrome DevTools

V prípade ladenia chýb je možné k dátam v IndexedDB pristúpiť prostredníctvom DevTools pod záložkou Application → IndexedDB, ako je znázornené na obr. 3.18.

## Validácia formulárov

Niektoré bloky pre svoju funkčnosť potrebujú správne nastavenie rozličných parametrov. Nové grafické rozhranie nedokáže predpovedať, či vytvorená aplikácia bude fungovať správne, avšak validácia užívateľského vstupu dopomáha k minimalizácii chýb pri zadávaní nesprávneho formátu vstupných parametrov. Ako validátor bola zvolená knižnica *react-hook-form* (<https://www.npmjs.com/package/react-hook-form>), monitorujúca vstupy polí formulárov a vyhodnocujúca chyby pred vytvorením nového bloku.

## PŘIDAT BLOK

Dvojklikem na blok přejdete do editace.

Výběr bloku

Poslat SMS

Poznámka

Bez poznámky

Číslo odesílatele

900100200

Číslo musí obsahovat předvolbu.

Číslo příjemce

Zadejte platné telefonní číslo.

Obr. 3.19 Příklad validace dat při vytváření bloku „Poslat SMS“

Na obr. 3.19 je příklad validace vstupných parametrov bloku „Poslat SMS“. Okrem chýbajúcich hodnôt je takisto kontrolovaný správny formát. V tomto prípade pole „Číslo odesílatele“ obsahuje telefonné číslo bez predvoľby.

## 3.5 Generátor dial planu

Pred odoslaním a uložením IVR aplikácie na server prebieha transformácia dátových štruktúr, pri ktorej sú mapy blokov, prepojení a triggerov spätne pretransformované na polia. Vo finále je model dat odosielaných na server zhodný s typom *DataModel* na obr. 3.20.

```

type Trigger = {
  id: string,
  action: string,
  label: string,
  color: string,
  isDynamic: boolean
}

type Block = {
  id: string,
  type: string,
  position: {
    x: number,
    y: number
  },
  data: {
    [key: string]: string
  },
  triggers: Trigger[]
}

type Connection = {
  id: string,
  source: string,
  sourceTrigger: string,
  target: string
}

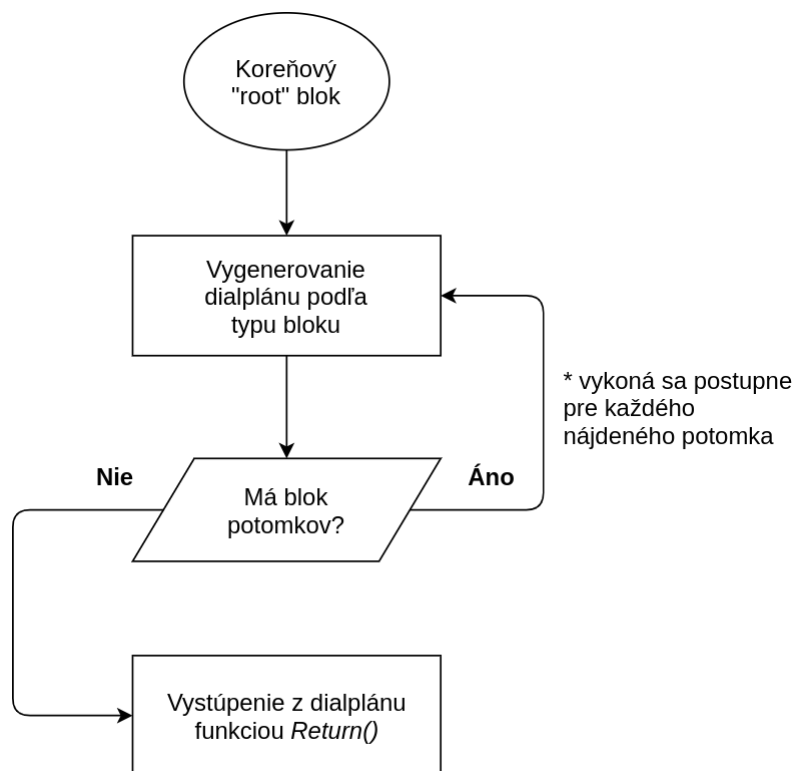
// Final server data structure
type DataModel = {
  id: string,
  name: string,
  description: string,
  blocks: Block[],
  connections: Connection[]
}

```

Obr. 3.20 Dátový model IVR aplikácie

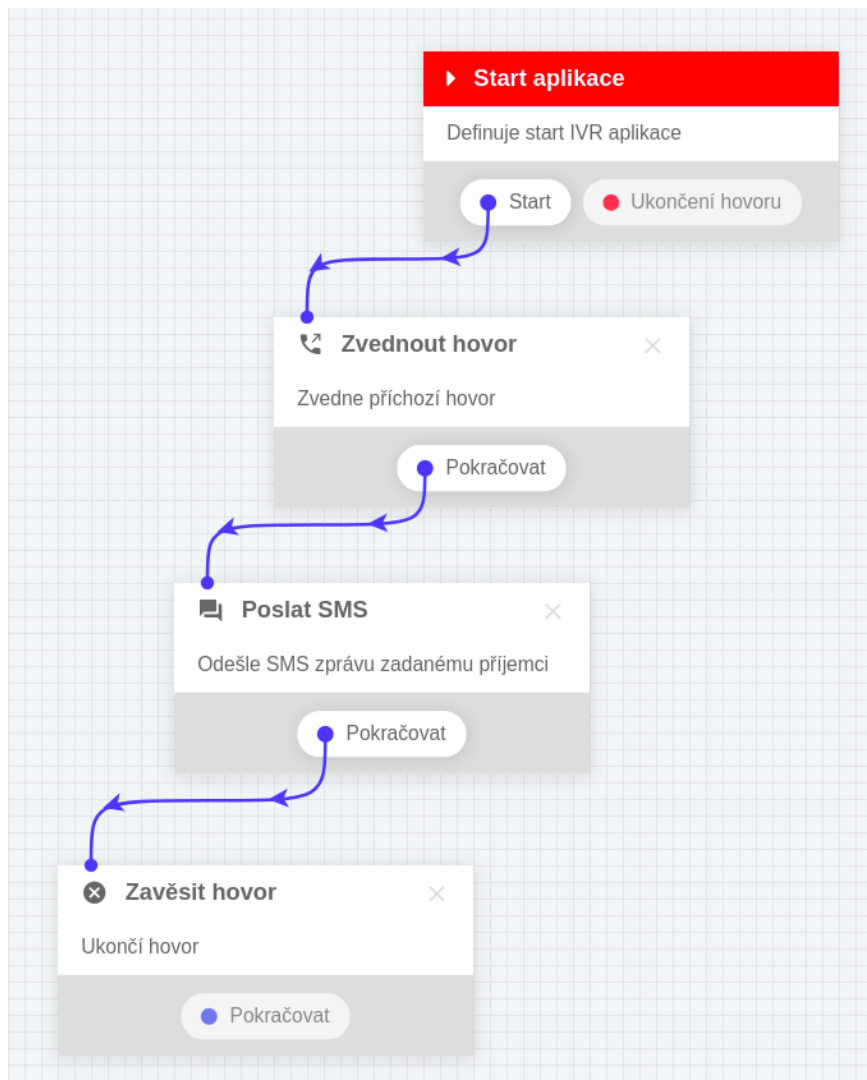
Túto úlohu vykonáva serverový endpoint centrálnej *ipbx-api*. Ten je zodpovedný za zápis dát do konfiguračnej MySQL databázy, vygenerovanie konfiguračných súborov pre dial plan a odoslanie súboru na virtuálnu ústredňu.

Generovanie prebieha na základe rekurzívneho vyhľadávania potomkov jednotlivých blokov. Algoritmus je popísaný na obr. 3.21.



Obr. 3.21 Algoritmus generovania dial planu z grafickej štruktúry

V prvom kroku je nájdený koreňový blok, ktorý je v našom prípade blok typu „Start aplikácie“. Pre každý blok je definovaný zoznam príkazov, ktoré sa pridávajú postupne do dial planu. Po vygenerovaní potrebných riadkov v dial plane sú vyhľadani ďalší potomkovia – bloky, s ktorými je v priamom prepojení. Pre každý z nájdených blokov sa tento proces opakuje. Ak blok neobsahuje žiadne prepojenia, generuje sa funkcia *Return()*, ktorá zaistí, aby v danom prípade dial plan nepokračoval na riadok s nasledujúcou prioritou.



Obr. 3.22 Jednoduchá IVR aplikace vytvořená v prostředí grafického štúdia

Každá IVR aplikace je separovaná do samostatného súboru *\*.conf*, ktorý nesie v názve ID aplikácie vytvorenej grafickým prostredím. Výhodou toho je jednoduchšia vzdialená správa na serveri virtuálnej ústredne, kedy odpadá nutnosť vyhľadávať kontext aplikácie v jednom rozsiahlom súbore. Príklad vygenerovanej aplikácie z obr. 3.22 v syntaxi dial planu je uvedený na obr. 3.23.



```

;custom app logging: 0
;custom app: test_diplomova_prace()
[capp-Test_diplomova_prace]
exten => _[*+#0-9].,1,ipbxlog("${UNIQUEID} - )
exten => _[*+#0-9].,n(block-GKHa9UP4x),Noop("Ivr application start")
exten => _[*+#0-9].,n(block-33WkQd3RH),ipbxlog("${UNIQUEID} - Zvednout hovor (5000 ms)")
exten => _[*+#0-9].,n,Answer(5000)
exten => _[*+#0-9].,n(block-sD_DZQ7-p),ipbxlog("${UNIQUEID} -
SMS +420910800800,+420910900900,Test diplomove prace...)
exten => _[*+#0-9].,n,Set(message=Test diplomove prace...)
exten => _[*+#0-9].,n,AGI(sms.php,+420910800800,+420910900900,${message})
exten => _[*+#0-9].,n(block-0Yiu6hYeB),ipbxlog("${UNIQUEID} - Hangup")
exten => _[*+#0-9].,n,Hangup()
exten => _[*+#0-9].,n,Return

exten => h,1,Gosub(extended_CDR,s,1)
exten => h,n,Hangup

```

Obr. 3.23 Výsledný konfiguračný súbor IVR aplikácie v syntaxi dial planu

Načítanie všetkých kontextov aplikácií zaobstaráva jeden riadok v súbore *extensions-capp.conf*.

```

// extensions-capp.conf
#include ivr.d/*.conf

```

Pri vytváraní alebo upravovaní IVR generátor upravuje iba konkrétnu konfiguračný súbor v priečinku *ivr.d* a nie je potrebné znovu vygenerovať všetky dial plany, tak ako tomu bolo doteraz.

Po ukončení generovania IVR je k dispozícii funkčný súbor so syntaxou dial planu, pripravený pre odoslanie na lokálne úložisko ústredne. Priradením konkrétnej aplikácie k linke v administračnom rozhraní iPBX je tak IVR systém plne pripravený na použitie.

## 4 TESTOVANIE

V tejto kapitole bude popísaný priebeh testovania nového grafického rozhrania. Keďže jednou z požiadaviek bolo, aby všetky funkcionality zostali vo väčšej miere zachované, testovanie prebiehalo vo viacerých fázach.

### 4.1 Testovanie blokov

V prvej fáze prebiehalo testovanie klientskej a serverovej validácie dát. Do rozhrania boli pridávané postupne všetky bloky s nakonfigurovanými testovacími dátami. Po každom pridanom bloku bola aplikácia odoslaná na server a sledovalo sa či uloženie prebehlo úspešne alebo bola požiadavka vrátená s chybou. Chybné bloky boli spätne opravené a znova otestované.

### 4.2 Testovanie generátora dial planu

V druhej fáze bol analyzovaný výstup z generátora dial planu. Sledovaná bola správna postupnosť vygenerovaných riadkov, správnosť parametrov vo funkciách dial planu a odpovedajúce ID blokov pri skokoch do rôznych návěstí dial planu. Testovaním bolo odhalených pár chýb, kde v jednej z nich dochádzalo k nesprávnej generácii hodnôt premenných a v niektorých častiach konfiguračného súboru sa tak objavili hodnoty *undefined*. Druhá z nájdených chýb bola priamo spätá so samotným unikátnym ID blokov. Unikátne ID je vytvorené pre každý blok pred jeho pridaním na plochu grafického rozhraní. Túto funkcionality zabezpečuje knižnica *shortid* (<https://www.npmjs.com/package/shortid>), ktorá vo svojom základe vytvára reťazec dlhý 7–14 znakov, pozostávajúci zo znakov a–z, A–Z, 0–9, \_ a - (spojovník). Tento reťazec je neskôr použitý ako návěstie (label) pre prvý riadok kódu príslušného bloku v dial plane a vo funkcii *GotoIf()*, ktorá iniciuje podmienené skoky. Problém nastáva v prípade výskytu \_ alebo - na začiatku ID, kedy dial plan Asterisk-u nevyhodnotí toto návěstie správne. Riešením bolo automatické pridanie prefixu *block-*, prostredníctvom generátora, pred každé ID bloku. Na obr. 4.1 je znázornený vygenerovaný súbor dial planu s červeno vyznačenými chybnými miestami.

```

;custom app logging: 0
;custom app: test_diplomova_prace()
[capp-Test_diplomova_prace]
exten => _[+##0-9].,1,ipbxlog("${UNIQUEID} - )
exten => _[+##0-9].,n(GKHa9UP4x),Noop("Ivr application start")
exten => _[+##0-9].,n(33WkQd3RH),ipbxlog("${UNIQUEID} - Zvednout hovor (5000 ms)")
exten => _[+##0-9].,n,Answer(5000)
exten => _[+##0-9].,n(sD_DZQ7-p),ipbxlog("${UNIQUEID} - SMS +420910800800,+420910900900,Test diplomove prace...")
exten => _[+##0-9].,n,Set(message=Test diplomove prace...)
exten => _[+##0-9].,n,AGI(sms.php,+420910800800,undefined,${message})
exten => _[+##0-9].,n(OYiu6hYeB),ipbxlog("${UNIQUEID} - Hangup")
exten => _[+##0-9].,n,Hangup()
exten => _[+##0-9].,n,Return
exten => _[+##0-9].,n(-j_2nHn4jE),ipbxlog("${UNIQUEID} - hovor ukoncen")
exten => _[+##0-9].,n,Return

exten => h,1,Gosub(extended_CDR,s,1)
exten => h,n,Goto(capp-Test_diplomova_prace,${IF([ foo${CALLERID(dnid)} != foo?
${CALLERID(dnid)}:${CALLERID(num)}]),-j_2nHn4jE)
exten => h,n,Hangup

```

Obr. 4.1 Súbor dial planu s vyznačenými chybami

### 4.3 Testovanie kompletných scenárov

Poslednou fázou v testovaní bolo testovanie kompletných scenárov. Z pôvodného rozhrania bola vybraná aplikácia, ktorej funkcionality bola rovnako nakonfigurovaná aj prostredníctvom nového grafického rozhrania. Po uložení aplikácie boli uskutočnené hovory na príslušné telefónne čísla liniek obsluhujúcich IVR aplikácie a porovnaním overená správna funkčnosť. Na nasledujúcich obrázkoch je pre porovnanie zobrazená IVR aplikácia s rovnakou funkcionality, nakonfigurovaná ako v doterajšom, tak aj v novom grafickom rozhraní.

## Porovnanie testovaných aplikácií

**Upravit aplikaci** ?

**Speciální aplikace dialplánu**

Název aplikace

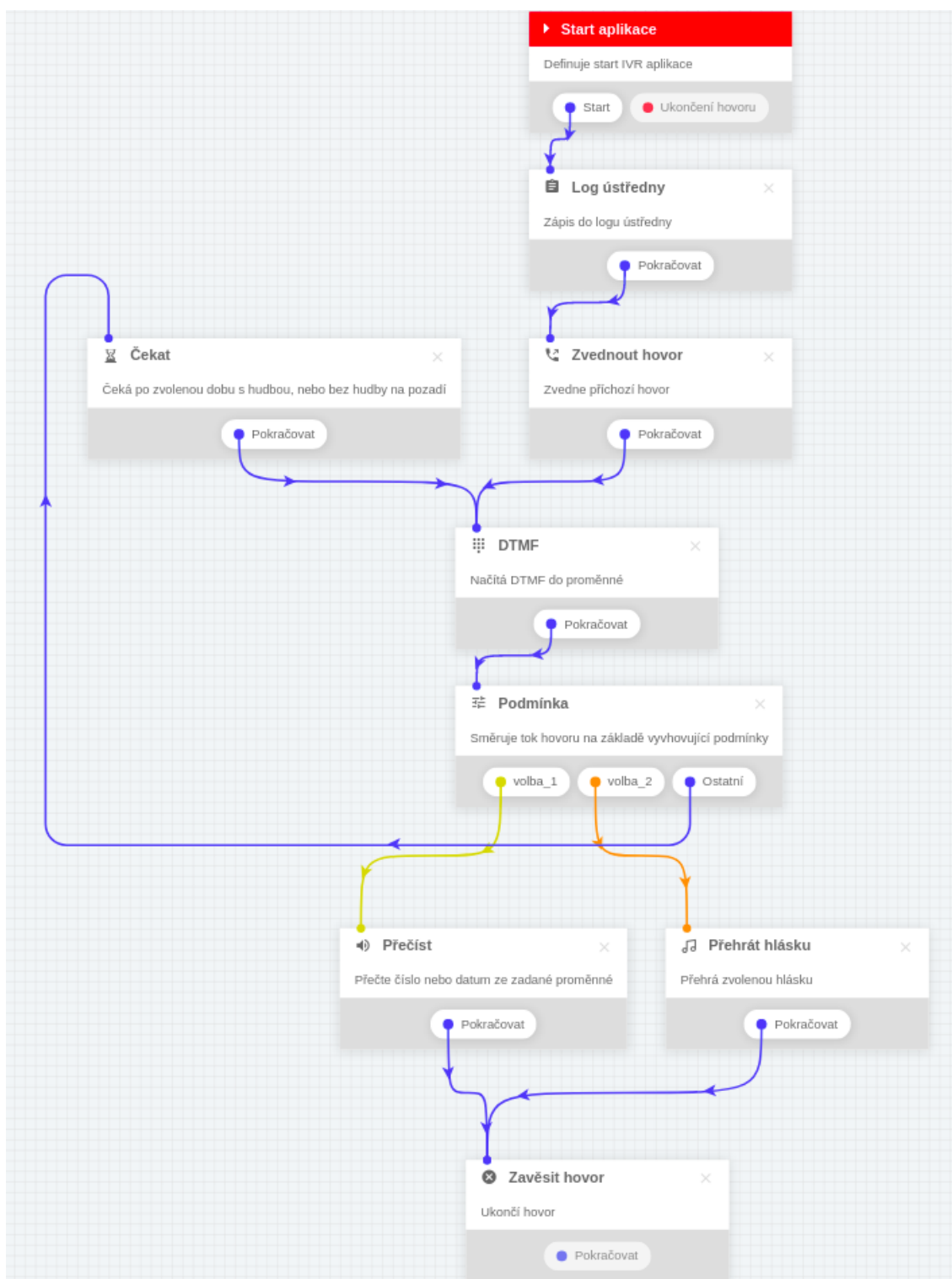
Popis aplikace

**Seznam pravidel aplikace**

Návěští	Akce	
	Zápis do logu ústředny	
	Zvednout hovor	
nacist_dtmf	Načíst DTMF do proměnné	
	Podmíněný skok na pozici	
	Podmíněný skok na pozici	
	Čekání	
	Skok na pozici	
volba_1	Přečti číslo z proměnné	
	Zavěsit hovor	
volba_2	Přehrát soubor	
	Zavěsit hovor	

Obr. 4.2 Konfigurácia testovanej IVR aplikácie v pôvodnom rozhraní

Funkciou aplikácií na obr. 4.2 a obr. 4.3 je zdvihnúť hovor, prehrať volajúcemu úvodnú hlásku a načítať zvolenú DTMF voľbu do premennej (spracováva pravidlo „Načíst DTMF do proměnné“). Po uložení nastane porovnanie hodnôt. Ak volajúci stlačil číslo „1“, aplikácia toto číslo prečíta a zavesí hovor. V prípade, že volajúci uvedie voľbu „2“, prehrá sa hláska (v našom prípade význam vybranej hlásky nehrá žiadnu úlohu, slúži iba pre testovacie účely) a hovor je opäť ukončený. Vo všetkých ostatných prípadoch aplikácia počká po dobu 2 s a opakuje celý proces prehratia hlásky a načítania DTMF do premennej (nekonečná slučka).



Obr. 4.3 Konfigurácia testovanej IVR aplikácie v novom grafickom rozhraní

Už pri takejto jednoduchej aplikácii je vidno, že absencia návští umožňuje lepší pohľad na tok hovoru cez IVR systém. Užívateľ menej orientovaný v telekomunikačných technológiách a pojmoch dokáže v tomto prípade lepšie pochopiť princíp aplikácie, poprípade upraviť niektoré parametre blokov.

Testovanie aplikácií vo viacerých fázach dopomohlo k nájdeniu a oprave chýb týkajúcich sa samotnej funkčnosti scenárov. V niektorých prípadoch došlo aj k opravám nedostatkov a chýb, ktoré boli historicky obsiahnuté v pôvodom rozhraní.

# ZÁVER

V diplomovej práci bolo predstavené nové grafické rozhranie IVR editora pre spoločnosť IPEX. V úvodnej kapitole sme postupne prešli teoretickým popisom základných pojmov ako sú PBX, IVR, DTMF či dial plan. Vysvetlený bol princíp fungovania dial planu vo frameworku Asterisk a následná analógia pri konfigurácii IVR aplikácie. Analýza dostupných riešení IVR editorov obsahovala základný prehľad systémov od spoločností Twilio, Apstel a Voicent spolu s uvedenými obrázkami ich grafických rozhraní.

Druhá kapitola bola zameraná na analýzu doterajšieho konfigurátora IVR aplikácií. Odôvodnené boli nedostatky, ktoré podnietili vývoj a stali sa motiváciou pri vytváraní nového riešenia.

Tretia kapitola obsahovala návrh grafického rozhrania IVR štúdia, kde boli predstavené základné ovládacie prvky, zoznam implementovaných blokov spolu s popisom ich funkcionalít a výber použitých technológií. Veľký dôraz bol kladený na optimalizáciu výkonu, vďaka ktorej je aplikácia pripravená pre manipuláciu s komplexnejšími diagramami. Súčasťou kapitoly je aj predstavenie generátora dial planu, ktorého úlohou je prevod štruktúry dát grafického rozhrania do syntaxe konfiguračných súborov pre ústredňu Asterisk.

V poslednej kapitole boli popísané formy testovania. Vďaka rôznym testovacím úrovňam bolo odhalených viacero chýb, ktoré by v produkčnom prostredí spôsobovali neželané efekty a v niektorých prípadoch aj nefunkčnosť samotnej aplikácie.

V čase pred odovzdaním diplomovej práce bol nový grafický systém pre vytváranie IVR už k dispozícii zákazníkom v ich administračnom rozhraní virtuálnej ústredne. Cieľom, ktorým bolo vytvorenie užívateľsky prívetivejšieho rozhrania, tak začína nová kapitola v ére poskytovania zákazníckej podpory, ktorá so sebou prináša implementáciu moderných technológií ako sú napríklad rozpoznávanie a spracovávanie reči.

# LITERATÚRA

- [1] PEARCE, J. The History of the Call Centre [online]. [cit. 2019-12-08]. Dostupné z: <https://www.callcentrehelper.com/the-history-of-the-call-centre-15085.htm>
- [2] SMITH, E. The History of Call Center Technology [Infographic] [online]. [cit. 2019-12-08]. Dostupné z: <https://www.teledirect.com/history-call-center-technology-infographic/>
- [3] OPTIMA CALL, Čo je PBX a ako funguje [online]. [cit. 2019-12-08]. Dostupné z: <https://www.optimalcall.com/co-je-pbx/>
- [4] OOMA, The Pros and Cons of VoIP Services [online]. [cit. 2019-12-08]. Dostupné z: <https://www.ooma.com/home-phone/pros-and-cons-of-voip-services/>
- [5] VOIP-INFO, Asterisk introduction. [cit. 2019-12-08]. Dostupné z: <https://www.voip-info.org/asterisk-introduction/>
- [6] BRYANT, R., COLP, J. Asterisk Versions [online]. [cit. 2019-12-08]. Dostupné z: <https://wiki.asterisk.org/wiki/display/AST/Asterisk+Versions>
- [7] GERAGHTY, S. What is IVR and 6 Benefits of Using One In Your Call Center [online]. [cit. 2019-12-08]. Dostupné z: <https://www.talkdesk.com/blog/what-is-an-ivr-and-6-benefits-of-using-one/>
- [8] MITRANESCU, M. What IVR Definition and Benefits [online]. [cit. 2019-12-08]. Dostupné z: <https://aircall.io/blog/call-center/interactive-voice-response/>
- [9] ŘEHÁK, J. DTMF - fámy a skutečnost [online]. [cit. 2019-12-08]. Dostupné z: <https://vyvoj.hw.cz/teorie-a-praxe/dokumentace/dtmf-famy-a-skutecnost.html>
- [10] VAN MUGGELEN, J., BRYANT, R., MADSEN, L. Asterisk: The Definitive Guide. 5th ed. O'Reilly Media, 2019. ISBN 9781492031604.
- [11] WANG, D. How To: Build a Santa Bot with Twilio Studio [online]. [cit. 2019-12-08]. Dostupné z: <https://www.twilio.com/blog/2017/12/build-a-santa-bot.html>
- [12] APSTEL, Visual Dialplan [online]. [cit. 2019-12-08]. Dostupné z: <https://www.apstel.com/visual-dialplan-professional/>
- [13] IVR Studio [online]. [cit. 2019-12-08]. Dostupné z: <http://voicent.com/ivr.php>



# ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

IVR	Interactive Voice Response, interaktívny hlasový asistent.
ACD	Automatic Call Distributor, automatický distribútor hovorov.
PBX	Private Branch Exchange, súkromná pobočková ústredňa.
PABX	Private Automatic Branch Exchange, automatizovaná súkromná pobočková ústredňa
VoIP	Voice over IP, prenos hlasu internetovým protokolom.
ISDN	Integrated Services Digital Network, digitálna sieť hlasových služieb.
LTS	Long Time Support, verzia s dlhoročnou podporou.
EOL	End Of Life, koniec podpory.
CRM	Customer Relationship Management, systém pre management vzťahov so zákazníkmi.
DTMF	Dual-tone Multi Frequency, tónová voľba.
CLI	Command Line Interface, príkazový riadok.
HTML	Hypertext Markup Language, hypertextový značkovací jazyk.
DOM	Document Object Model, objektový model dokumentu
SVG	Scalable Vector Graphics, škálovateľná vektorová grafika.
API	Application Programming Interface, rozhranie pre programovanie aplikácií.

## ZOZNAM OBRÁZKOV

Obr. 1.1 Operátorka switchboardu v roku 1971 .....	8
Obr. 1.2 Infraštruktúra možného riešenia VoIP (IP PBX).....	9
Obr. 1.3 Vzťah medzi konfiguráciou zariadení a kontextom dial planu .....	12
Obr. 1.4 Grafické rozhranie Twillio Studio (prevzaté z [11]) .....	13
Obr. 1.5 Grafické rozhranie Visual Dialplan Professional .....	14
Obr. 1.6 Grafické rozhranie Voicent IVR Studio (prevzaté z [13]) .....	15
Obr. 2.1 Administratívne rozhranie iPBX ústredne.....	16
Obr. 2.2 Popis prostredia konfigurácie IVR aplikácie v systéme iPBX.....	17
Obr. 2.3 Schéma komunikácie medzi administratívnym rozhraním a virtuálnou ústredňou.....	18
Obr. 3.1 Príklad členenia štruktúry React.js aplikácie na komponenty .....	19
Obr. 3.2 Blok Start aplikácie - vstupný bod IVR aplikácie .....	20
Obr. 3.3 Vzájomné prepojenie blokov IVR aplikácie .....	21
Obr. 3.4 Popis grafického rozhrania IVR .....	22
Obr. 3.5 Zoznam nakonfigurovaných IVR aplikácií .....	23
Obr. 3.6 Modálne okno pre vytvorenie novej aplikácie .....	23
Obr. 3.7 Editor blokov - pridanie nového bloku (vľavo), editácia pôvodného bloku (vpravo).....	24
Obr. 3.8 Ukážka bloku „Ověření dostupnosti linky“ .....	26
Obr. 3.9 Vytvorenie novej podmienky pre vetvenie toku hovoru .....	26
Obr. 3.10 Ukážka bloku „Podmínka“ .....	27
Obr. 3.11 Ukážka bloku „Tabulka času“ .....	28
Obr. 3.12 Ukážka bloku „Volat kanál“ .....	28
Obr. 3.13 Porovnanie dátových štruktúr – pole a mapa.....	30
Obr. 3.14 Test efektívnosti dátových štruktúr (pole vs. mapa).....	31
Obr. 3.15 Použitie React Context API pre data providing.....	32
Obr. 3.16 Profil prekreslených React komponentov pred optimalizáciou.....	32
Obr. 3.17 Profil prekreslených React komponentov po optimalizácii.....	33
Obr. 3.18 Prístup k IndexedDB z rozhrania Chrome DevTools.....	34
Obr. 3.19 Príklad validácie dát pri vytváraní bloku „Poslat SMS“ .....	35

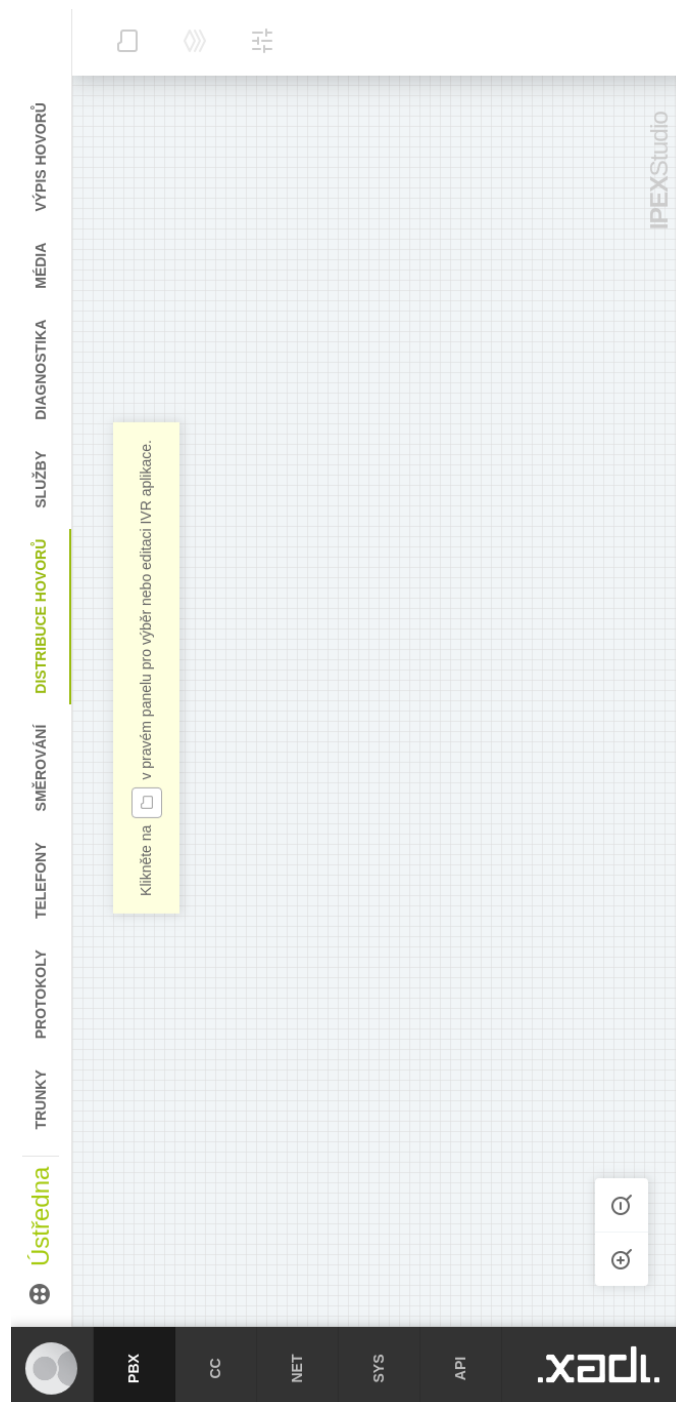
Obr. 3.20 Dátový model IVR aplikácie .....	36
Obr. 3.21 Algoritmus generovania dial planu z grafickej štruktúry .....	37
Obr. 3.22 Jednoduchá IVR aplikácia vytvorená v prostredí grafického štúdia .....	38
Obr. 3.23 Výsledný konfiguračný súbor IVR aplikácie v syntaxi dial planu .....	39
Obr. 4.1 Súbor dial planu s vyznačenými chybami .....	41
Obr. 4.2 Konfigurácia testovanej IVR aplikácie v pôvodnom rozhraní .....	42
Obr. 4.3 Konfigurácia testovanej IVR aplikácie v novom grafickom rozhraní.....	43

## **ZOZNAM TABULIEK**

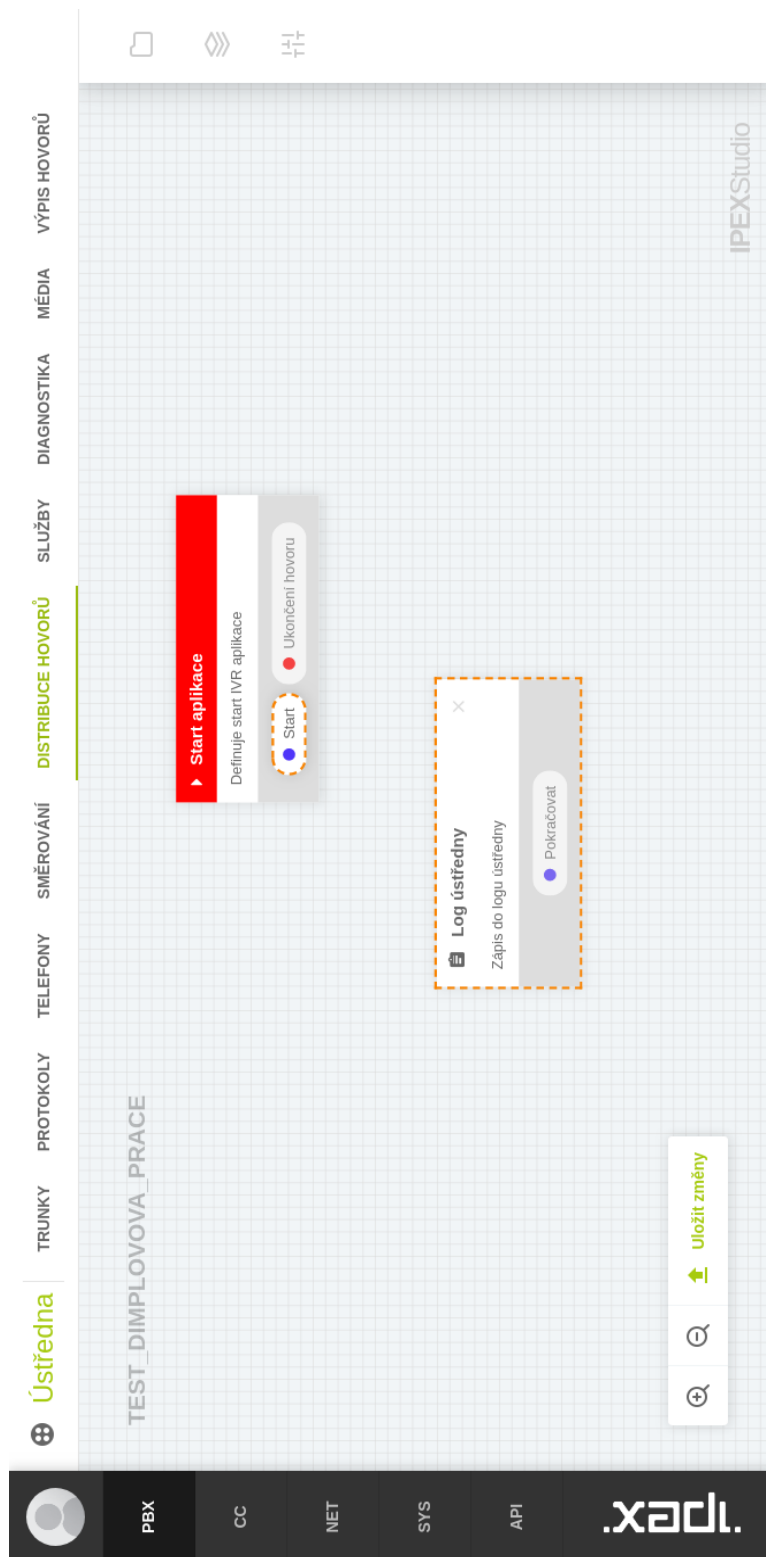
Tab. 1.1 Prehľad vydaných verzií Asterisk [6].....	10
Tab. 1.2 Prehľad frekvenčných párov pre jednotlivé DTMF voľby [9] .....	11

# A KONCEPT GRAFICKÉHO ROZHRANIA

## A.1 Základné zobrazenie pracovnej plochy



## A.2 Mód vytvárania prepojení



## A.3 Editácia bloku

The screenshot displays the .xadi system interface for editing a block. The top navigation bar includes: Ústředna, TRUNKY, PROTOKOLY, TELEFONY, SMĚROVÁNÍ, DISTRIBUCE HOVORŮ, SLUŽBY, DIAGNOSTIKA, MÉDIA, VÝPIS HOVORŮ. The main workspace shows a workflow diagram titled "TEST\_DIMPLOVOVA\_PRACE" with three steps: "Start aplikace" (Define start IVR application), "Log ústředny" (Log call center), and "DTMF" (DTMF). A "Uložit změny" button is visible at the bottom right of the workspace.

The right-hand configuration panel, titled "UPRAVIT BLOK (QRTFJAQLE)", includes the following settings:

- Vyběr bloku:** DTMF
- Poznámka:** Bez poznámky
- Audio:** cz/agent-newlocation.wav
- VYBRAT HLÁSKY:** (empty)
- Název proměnné:** dtmf
- Maximální počet tísel:** 2

At the top of the configuration panel, there are icons for "ZPĚT NA NOVÝ BLOK", a folder icon, and a list icon.

## A.4 Výber hlások pre blok DTMF

The screenshot shows a web application interface with a top navigation bar containing the following menu items: Ústředna, TRUNKY, PROTOKOLY, TELEFONY, SMĚROVÁNÍ, DISTRIBUCE HOVORŮ, SLUŽBY, DIAGNOSTIKA, MÉDIA, and VÝPIS HOVORŮ. The main content area is partially visible, showing a search bar and a list of items. A modal dialog titled "Výběr souborů" (File Selection) is open, displaying a list of audio files. The dialog has a close button (X) in the top right corner. The selected files are marked with checkmarks in the first column.

Selected files:

- asistent\_away.wav (107.04 kB, 2020-05-14 21:39:19, VLASTNÍ)
- asistent\_busy.wav (171.06 kB, 2020-05-14 21:39:19, VLASTNÍ)

Other files in the list:

- / cz / ipbx /
- ...
- CHANGES-asterisk-core-cz-3.0 (2020-04-06 16:27:57, SYSTEMOVÁ)
- asistent\_telefonuje.wav (116.45 kB, 2020-05-14 21:39:18, VLASTNÍ)
- day-mod.wav (43.57 kB, 2020-05-14 21:39:18, VLASTNÍ)

Buttons: "POTVRDIT" (Confirm) and "Uložit změny" (Save changes).

## A.5 Uloženie aplikácie

