VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

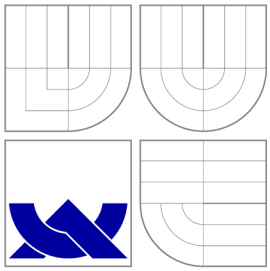
ANDROID APLIKACE PRO VIZUALIZACI DOPRAVNÍCH DAT ZE SYSTÉMU C2X

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

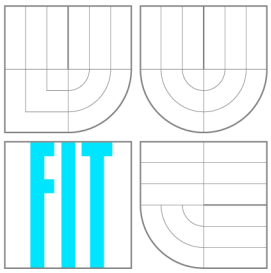
AUTOR PRÁCE
AUTHOR

Bc. TOMÁŠ CHLÁPEK

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ANDROID APLIKACE PRO VIZUALIZACI DOPRAVNÍCH DAT ZE SYSTÉMU C2X

ANDROID APPLICATION FOR TRAFFIC DATA VISUALIZATION FROM CAR2X SYSTEMS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. TOMÁŠ CHLÁPEK

VEDOUCÍ PRÁCE
SUPERVISOR

doc. RNDr. JITKA KRESLÍKOVÁ, CSc.

BRNO 2015

Abstrakt

Tato práce přináší aplikaci pro mobilní zařízení umístěné v automobilech. Slouží řidičům ke zlepšení orientace a zvýšení přehlednosti při jízdě. Nabízí grafické zobrazení aktuálních stavů křižovatek a světelného značení. V řešení bylo použito metod využívaných v inteligentních dopravních systémech. Vychází z komunikace mezi zařízeními umístěnými podél silnic a mobilním zařízení ve vozidle. Podstatou je výměna zpráv s informacemi o topologii křižovatek a stavy semaforů. Vytvořené řešení poskytuje možnosti grafického vizualizace křižovatek se světelným značením a přechody pro chodce na zařízeních s platformou Android ihned po přiblížení se k externímu zařízení. Hlavní přínos je pro vývojáře systému C2X, kteří za pomocí aplikace testují správnost jimi nastavených informací ve fyzických zařízeních nainstalovaných v těsné blízkosti podporovaných křižovatek.

Abstract

This thesis brings application for mobile devices placed in cars. It is used for drivers for enhancing orientation and driving clarity. It is based on communication between roadside units and devices in vehicle. Application offers graphical visualization of intersection states and traffic lights. Created solution provides graphical visualization of traffic data designated for system Android. Main benefit is for C2X developers who test correctness of information, which were inserted to physical devices near supported intersections.

Klíčová slova

c2x, android, its, siemens, map, spat, mobil, doprava

Keywords

c2x, android, its, siemens, map, spat, mobile, traffic

Citace

Tomáš Chlápek: Android aplikace pro vizualizaci dopravních dat ze systému C2X, diplomová práce, Brno, FIT VUT v Brně, 2015

Android aplikace pro vizualizaci dopravních dat ze systému C2X

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením paní doc. RNDr. Jitky Kreslíkové, CSc.

.....
Tomáš Chlápek
27. května 2015

Poděkování

Děkuji své vedoucí, paní doc. RNDr. Jitce Kreslíkové, CSc., za odbornou přípravu a vedení, panu Ing. Janu Vernerovi za profesionální přístup a panu Ing. Pascalovi Cassinellimu za konzultace ve společnosti Siemens s.r.o.

© Tomáš Chlápek, 2015.
Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod **3**

2 Analýza inteligentních dopravních systémů **4**
2.1 Životní cyklus dopravního systému 5
2.2 Problémy v dopravě 6
2.3 Případy užití 6
2.4 Architektura 7
2.5 Komunikační technologie 12
2.6 Ukládání prostorových dat 15
2.7 UML diagramy 15

3 Návrh komunikačního rozhraní **17**
3.1 Zadavatelská společnost 17
3.2 Platforma 18
3.3 Proces vytváření křižovatky 19
3.4 Typ zpráv 21
3.5 Syntaktický rozbor zprávy 22
3.6 Přenos zpráv 22
3.7 Topologie 23
3.8 Struktura aplikace 24

4 Implementace aplikace pro vizualizaci dopravních dat **26**
4.1 Vývojové prostředí 26
4.2 Inicializace 26
4.3 Hlavní vlákno 29
4.4 Zpracování polohy 31
4.5 Datová struktura 35
4.6 Ukládání dat 38
4.7 Vizualizace dat 39
4.8 Zobrazení dynamických stavů 42
4.9 Uživatelská nastavení 43

5 Testování **45**
5.1 Zpracování zpráv 45
5.2 Komunikace s jednotkou 45
5.3 Simulátor 46
5.4 Správa verzí 47
5.5 Ladící mód 47

5.6 Výsledky testování 48

6 Závěr **49**

A Obsah CD **52**

B Zkratky **53**

C Správa verzí **56**

Kapitola 1

Úvod

Android aplikace pro vizualizaci dopravních dat ze systému C2X přináší pomocníka určeného pro aktivní účastníky silničního provozu. Účelem bylo vytvoření mobilní aplikace, která slouží řidičům, po zasazení jejich mobilních zařízení do vozidla, ke zlepšení orientace na vozovce, včasné zobrazení blížící se křižovatky a vizualizaci proměnlivých aspektů, jakými jsou například světelná značení. Grafická podoba těchto informací je primárně přinášena pro koncové uživatele - řidiče, kteří využijí rychlý náhled na blížící se situaci na vozovce bez nutnosti složitého čtení přijatého nestrukturalizovaného kódu. Aplikace přináší výhody pro návrháře a testery inteligentního dopravního systému C2X. Umožňuje jim validaci a verifikaci jimi navržených dat, jenž byly vloženy do vysílačů podél cest a křižovatek. Důležitým aspektem je rovněž zvýšení povědomí o výhodách inteligentních dopravních systémů mezi veřejnost prostřednictvím graficky přívětivé prezentace nových technologií skrze rozšířenou mobilní platformu.

Cílem je vytvořit mobilní aplikaci, se kterou se uživatel po vstupu do vozidla připojí k palubní jednotce systému C2X. Po přiblížení ke světelné křižovatce integrované do systému dorazí data z venkovní jednotky do jednotky v automobilu, která je přepošle mobilnímu zařízení. Aplikace tato data rozpozná a na jejich základě vykreslí topologii křižovatky s graficky odlišenými jízdními pruhy, přechody a dalšími dopravními komponentami. Po výběru pruhu se uživateli zobrazí aktuální světelný stav přiřazených semaforů a časy, které zbývají do přepnutí stavů příslušných lamp.

Diplomová práce je rozdělena do pěti kapitol. V úvodní kapitole je motivace, proč byla práce vytvořena. Kapitola 2 analyzuje inteligentní dopravní systémy. Zaměřuje se na systém C2X, který popisuje z hlediska vývoje, využití, architektury a komunikačních technologií. Obsahem kapitoly 3 je návrh komunikačního rozhraní. V úvodu kapitoly je charakterizována společnost, pro kterou byla tato práce vyhotovena, a je zmíněn důvod výběru platformy Android. V dalších sekcích je vylíčen proces vytváření křižovatky dopravními inženýry a komunikace prostřednictvím zpráv. Kapitola 4 pojednává o implementaci aplikace. V jednotlivých sekcích je popsáno navázání komunikace mezi mobilním zařízením a externí jednotkou, zpracování polohy, specifikování uživatelských nastavení, ladění, uložení přijatých dat a jejich grafická reprezentace. Kapitola 5 se zabývá testováním aplikace. Zejména pak rozborem zpráv, analýzou paketů a nástroji vyvinutými pro testování aplikace mimo dosah aktivních jednotek systému C2X. Závěrečná kapitola 6 shrnuje dosažené výsledky.

Tato práce plynule navazuje na semestrální projekt. Jelikož byla vypracovávána čtyři semestry, v rámci semestrálního projektu již byl implementován prototyp aplikace, jenž na základě přijatých dat vykresloval topologii křižovatky. Ze semestrálního projektu byla převzata a upravena kapitola 2. Taktéž byla modifikována a rozšířena kapitola 3.

Kapitola 2

Analýza inteligentních dopravních systémů

Inteligentní dopravní systémy *ITS (Intelligent Transportation System)* představují informační zobrazení dopravních procesů v radioelektronickém, telekomunikačním a telematickém prostředí. Aplikace z oboru telekomunikací, skladebných a procesních informačních systémů jsou spojeny s navigačními systémy a digitálními obrazy světa. Široká škála komunikačně založených aplikací zvyšuje bezpečnost při cestování, zlepšuje správu provozu, minimalizuje dopady na prostředí a maximalizuje benefity z jízdy, jak pro komerční uživatele, tak pro širokou veřejnost.

ITS pokrývá všechny způsoby dopravy a zahrnuje všechny elementy transportního systému - automobily, infrastrukturu a dynamicky komunikující účastníky provozu. Základním jádrem jsou informace. Ty jsou statické či získané v reálném čase. Systémové nástroje jsou založeny na sběru, zpracování, integraci a poskytování dat. Zpracovaná data zachytávají aktuální situaci, umožňují plánování cest veřejnou i soukromou dopravou a zlepšují celkový přehled o situaci na daném území.

Asistence umožňuje řidičům průběžné sledování bezpečné vzdálenosti a povolené rychlosti, dodržování jízdních pruhů, snižuje míru nebezpečných situací a zjednodušuje průjezdnost komplikovanými křižovatkami. Při využití komunikace vozidel mezi sebou a v rámci dopravní infrastruktury jsou tyto benefity mnohonásobně zvýšeny. To vedlo k vytvoření kooperativního ITS umožňujícího výše zmíněnou komunikaci. Tím se otevřely nové možnosti a služby pro aktivní i pasivní účastníky provozu, jenž vedou k efektivnější dopravě a vyšší bezpečnosti na cestách. Standardizace je zaměřena na bezdrátovou komunikaci systémů, redukci času potřebného k přepravě a snížení znečištění ovzduší vlivem emisí. Prioritou je zmenšení počtu dopravních nehod a zranění. Vzhledem k tomuto faktu se na vývoji ITS podílí velká škála společností automobilového průmyslu.

Drive C2X[17] je decentralizovaný systém zabývající se komunikací *C2C (Car-To-Car)* mezi vozidly navzájem a komunikací *C2I (Car-To-Infrastructure)* mezi vozidly, vozovkou a infrastrukturou probíhající v ad-hoc módu s využitím všesměrových zpráv. Síť je vybudována s ohledem na evropskou dopravní architekturu pro spolupráci řídících systémů. Technologie je založena na standardech vyvinutých evropským ústavem pro telekomunikační normy *ETSI (European Telecommunications Standards Institute)*. Při jejím vytváření bylo čerpáno ze zkušeností z projektu *PRE-DRIVE C2X (Preparation for Driving Implementation and Evaluation of C2X)*. Využívá jeho specifikací, prototypů, prostředí a simulačních nástrojů.

2.1 Životní cyklus dopravního systému

Životní cyklus nové technologie je rozdělen na 3 iterativní fáze zahrnující průběžná měření a vyhodnocování. S přílivem nových poznatků nebývá návrat k předešlé fázi ničím neobvyklým. Ve fázi výzkumu ITS program integruje výuku a komunikaci s partnery. Ve fázi vývoje spolupráci a testování s blízkými společnostmi. A ve fázi nasazení trénink se vzděláváním.

Fáze výzkumu

Fáze výzkumu začíná myšlenkou, na kterou navazuje plánování, získávání poznatků a validace eventuálně realizovatelných nápadů. Zde jsou podporovány návrhy, myšlenky a úsudky zainteresovaných osob. Hlavní body dle oficiálního strategického plánu[1] jsou:

- organizace programu okolo klíčových bodů zájmu
- zjištění požadovaných výsledků programu
- definice milníků

Fáze vývoje

Fáze vývoje je spojena s tvorbou prototypů, testováním, zajištěním bezpečnostních prvků, minimalizací rizik, standardizací a definováním množiny použitých technologií. Disciplíny vývoje:

- definice výstupů
- stanovení výsledného procesu vyhodnocení
- vývoj a testování prototypů a jejich znovuzačlenění
- analýza výsledků

Fáze nasazení

Fáze nasazení je finálním bodem programu. Po úspěšném testování a splnění všech požadavků přichází čas k převzetí technologie, její nasazení, zajištění podpory, začlenění do operačního prostředí a zaučení cílových uživatelů. Jednotlivé kroky zavádí:

- podpora pro pilotní uživatele
- maximální snaha o zajištění úspěchu v reálném prostředí
- sledování chování a snížení rizik v delších časových úsecích
- zacílení na správný trh

[1] `http://www.its.dot.gov/strategicplan/index.html`

2.2 Problémy v dopravě

Dopravní zácpy

Dopravní zácpy jsou hlavním problémem všech transportních sítí. Zefektivnění průjezdnosti je hlavním cílem ITS po celém světě. Toho je dosahováno zaváděním řídících systémů, jejich správou a monitorováním ve špičce i mimo ni. Služby zahrnují poradce pro přesměrování, kontroly rychlosti, měření průjezdnosti v závislosti na čase a detekci i prevenci kolizí.

Ochrana životního prostředí

Vozidla mají v současné době výrazný dopad na životní prostředí, čemuž nepomáhá fakt, že se silniční doprava na celém světě každým rokem více rozmáhá. Zlepšení dopravních systémů pozitivně ovlivňuje míru dopadu dopravy na přírodu. To je ovlivňováno aktivním monitorováním emisí v závislosti na poloze. Z těchto dat je vytvářen komplexní model emisí, do kterého jsou zahrnovány okolní aspekty, jako okolní teplota, stoupání vozovky a další proměnné. Za tímto účelem je využíváno geografického informační systému *GIS (Geographic Information System)*.

Produktivita

Lokalizace automobilů dle jejich geografické polohy umožňuje časové výpočty vedoucí k účelnému vybrání nejvhodnějších tras od aktuální pozice až do cílové destinace. Ve spojení s dynamicky získávanými informacemi o zácpách a zhoršených silničních podmínkách jsou tato data výrazným pomocníkem a šetřitelem časových i finančních zdrojů.

Komfort

Aktuální informace o výlukách, potvrzení vybraných tras, odhadovaná doba jízdy, hlášení nehod a vizualizace dopravní topologie značně zlepšuje život řidičům, snižuje stres a zvyšuje komfort, pocit sebejistoty a bezpečnosti na cestách. Tyto služby spojující řidiče a nabízející varování o možných nebezpečích navozují větší pohodlí při překonávání dlouhých vzdáleností. ITS zprostředkovává tyto informace v reálném čase řidiči i spolujedoucímu.

2.3 Případy užití

Záchranáři

Záchranářská vozidla blížící se ke křižovatce mohou explicitně vyžádat změnu světel semaforů. Na základě požadavku se uloží aktuální stav křižovatky a pro směr, ve kterém jede záchranářské auto, se rozsvítí zelená. Všechny ostatní směry dostanou v ten jistý okamžik červenou. Po projetí je množina semaforů náležících křižovatce vrácena do původního stavu před touto událostí [11].

Veřejná doprava

Vozidla veřejné dopravy mají oproti běžným vozům větší prioritu. Během jejich příjezdu ke křižovatce (monitorovaném *GPS (Global Positioning System)* lokátorem) se vypočítá odhadovaný čas příjezdu a systém ovládající světla semaforů přejde dle jeho vnitřní logiky do stavu, ve kterém bude minimalizovat dobu čekání na červené pro dané vozidlo.

Zelená vlna

Řidiči, blížícímu se k množině křižovatek podporujících tuto funkci synchronizace semaforů, jsou zasílány instrukce pro navázání na zelenou vlnu. Uživatel je dle informací o jeho poloze a aktuální rychlosti obeznámen s faktem, zdali se v zelené vlně nachází.

Semafory

Řidiči je zobrazována informace o aktuálním a následujícím stavu nadcházejícího semaforu. Tyto data jsou zpracovávány. Výstupem jsou doporučení ke zpomalení či zrychlení pro dosažení plynulého průjezdu. Uživatel taktéž vidí míru šance ke kolizi.

Odstávky

Řidičům jsou prezentovány informace o aktuálních problémech na vozovce, odstávkách a uzávěrkách silnic. Dle jejich polohy jsou vypočítávány nejoptimálnější přesměrování. Lze tak předejít zácpám způsobeným pracemi na cestách.

Řízení přístupu

V oblastech s monitorovaným přístupem (např. rozsáhlé parkoviště) je identita uživatele automaticky kontrolována ihned po jeho příjezdu. Systémová jednotka může zprostředkovávat platby mezi uživatelem a vnitřním systémem. Využívá se proto elektronická peněženka. Využito toho může být v půjčovnách aut, u pump nebo drive-in prodejen.

2.4 Architektura

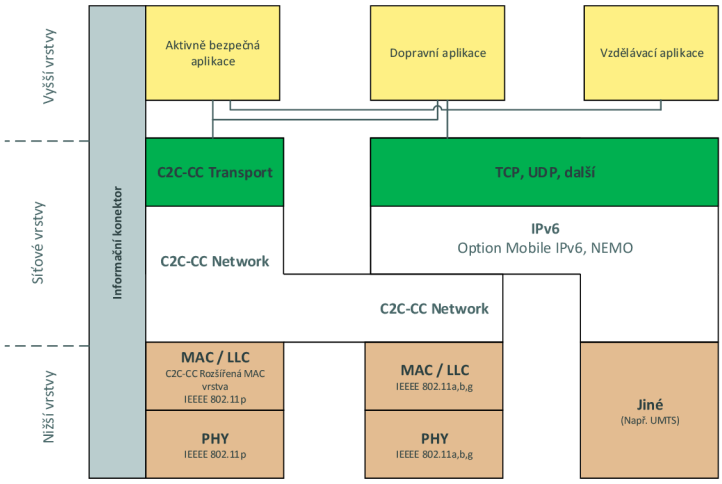
Komunikační architektura ETSI definuje 4 typy ITS stanic: Roadside, Central, Vehicle a Personal. Mobilní aplikační jednotka *AU (Application Unit)* je chápána jako rozhraní *HMI (Human-Machine Interface)*. To primárně slouží k vizualizaci přijatých dat na základě uživatelských požadavků získávaných hlasovým vstupem. Stanice mezi sebou navzájem komunikují skrze standardizované zprávy *DENM (Distributed Environmental Notification Message)*, zasílané komunikační a kontrolní jednotkou *CCU (Communication Unit)* prostřednictvím standardu IEEE 802.11p. Ta je typicky osazena operačním systémem Linux, *WLAN (Wireless LAN)* anténou a *CAN (Controller Area Network)* adaptérem. Její komponenty zajišťují veškerou komunikaci od fyzické až po síťovou vrstvu. Zprávy jsou používány k poskytnutí informací o prostředí, nehodách a dalších dopravních událostech. Každé vozidlo spravuje svou tabulku sousednosti, kvůli pravidelnému zasílání *CAM (Cooperative Awareness Message)* zpráv a uchování aktuálního stavu.

2.4.1 Vrstvy

Architektura systému je složena ze tří vrstev, jenž jsou níže popsány.

Datová vrstva

Stará se o sběr dat zachycených senzory, kamerou a nashromážděných informací z vozidel. Komunikace mezi vozidly je realizována prostřednictvím IEEE 802.11p.



Obrázek 2.1: Architektura [inspirováno [10]]

Komunikační vrstva

Zaopatřuje přenos dat mezi položkami infrastruktury a centrem. Data z vozidel dorazí k blízké stanici přes IEEE 802.11p a dále jsou transformována a posílána centrální stanici přes 3G[2].

Aplikační vrstva

Zpracovává nashromážděná data. Ukládá je do databází pro další použití. Dochází zde také ke správě *RSU (Road Side Unit)*.

2.4.2 Typy zpráv

V systému C2X je komunikace mezi vozidly a objekty infrastruktury realizována prostřednictvím zpráv. Ty jsou zasílány dle jejich využití skrze unicast či broadcast. Struktura zpráv se mění a není zatím pevně ustanovena. Je to dáno tím, že systém stále prochází fází vývoje. Stejně tak nebyla uzavřena množina zpráv. V následujícím souhrnu jsou popsány nejvyužívanější z nich. Tato práce se dále blíže věnuje zprávám *MAP (MapData)* a *SPAT (Signal Phase and Timing)*, jejichž data jsou vizualizována.

CAM

Zprávy *CAM (Cooperative Awareness Message)* slouží k periodickému zasílání informací o okolních vozidlech po kanálu *CCH (Control Channel)*. Zprávy se zasílají v intervalu 100 ms a užívají se k vybudování tabulky sousednosti. Každému odesilateli je vytvořen profil dle jeho profese. Uchovává informace o typu vozidla, rozměrech, stavu potkávacích i dálkových světel, historii poškození a dalších parametrech. Mohou také zasílat nouzový signál a nebo indikovat nehodu.

[2] Třetí generaci mobilní sítě

CAM zpráva			
Délka [byte]	Informace	Informace	Informace
1	messageID [CAM = 0 , DENM = 1]		
8	generationTime		
4	stationID		
1	stationInfo	mobileITS	
1		privateITS	
1		physicalRelevantITS	
8 | 8 | 4	referencePosition	lon | lat | elev	
4		heading	
32 | 4		streetName | roadSegID	
1		position	
1	camParameters	vehicleCommonParameters	vehicleType
2 | 2			length | width
4			speed
2			acceleration
1			accelerationControl [break, ACC, ..]
1			exterorLights
1			occupancy
1 | 1			crashStatus | dangerGoods

Obrázek 2.2: Struktura CAM [inspirováno [13]]

MAP

Zpráva MapData je využívána k obalení objektů spjatých se všemi typy map definovanými standardem. Jsou zahrnuty položky pro detailní popis křižovatek a jejich polohy, jízdních pruhů, skupin světelných znamení, přechodů a jednotlivých segmentů cest. Na jejich základě lze vygenerovat celou topologii křižovatek s veškerými statickými objekty. O jejich distribuci se stará jednotka RSU.

SPAT

Signal Phase and Timing zpráva zachycuje aktuální stav jednoho či skupiny světelných signalizačního zařízení. Obdobně jako zpráva MAP je vysílána jednotkou RSU a přijímána zařízením *OBU (On Board Unit)*, které je zabudované ve vozidle. Ve spolupráci se zmíněnou MAP zprávou dokáže určit signalizační fázi, prioritu a nadcházející očekávané stavy přiřazené konkrétním pruhům. Stavy jsou mapovány na specifické jízdní pruhy za použití identifikátorů, které odkazují na entity ve zprávě MAP.

SAM

SAM (Service Announcement Message) zprávy jsou všesměrově vysílané informace o službách poskytovaných RSU. Vysílány jsou skrze CCH.

DENM

DENM jsou zprávy lokálního charakteru založené na událostech. Využívají se k varování řidičů o snížené viditelnosti, nehodách, prudkém brzdění, pracích na vozovce a dalších akcích. Specifikuje se v nich daná situace a lokace. Po detekování události ITS stanice ihned zasílá informace o dění všem dalším stanicím v postižené oblasti. Zprávy jsou všesměrově zasílány po celou dobu, kdy je událost aktivní.

DENM zpráva			
Délka [byte]	Informace	Informace	-
1	messageID [CAM = 0 , DENM = 1]		
6	generationTime		
4	management	originatorID	
2		seqNum	
1		dataVer	
6		expiryTime	
1		frequency	
1		reliability	
1		isNegation	
1	situation	causeCode	
1		subCauseCode	
1		severity	
4	locationContainer	sitLat	
4		sitLon	
2		sitAlt	
4		accurancy	
N-40		relevenceArea	

Obrázek 2.3: Struktura DENM [inspirováno [13]]

2.4.3 Směrování

Důležitým aspektem kromě samotné struktury zprávy je použité směrování. To se volí dle cílových příjemců, rozsahu vzdáleností a požadované spolehlivosti.

- **Geograficky-zaměřený broadcast (GPC):** Využíván pro směrování zpráv geograficky spjatých s polohou definovanou kruhem se středem a poloměrem nebo obdélníkem s dvěma krajními body. Pokud je cílová stanice mimo oblast, zpráva je směrována metodou *direct-flooding*.
- **Topologicky-zaměřený broadcast (TSB):** Typ semínkového algoritmu omezeného maximální životností paketu danou počítadlem *TTL (Time To Live)*. TTL je maximální počet uzlů, kterými může zpráva projít. V každém uzlu je o jednotku snížen. Po dosažení nuly je zpráva vygenerována znova.
- **Single-Hop broadcast (SHB):** Zpráva určena pouze pro nejbližší sousední uzly. Implementačně je toho dosaženo snížením počítadla TTL na hodnotu jedna.
- **Unicast:** Zprávy jsou zasílané jedním odesílatelem přesně jednomu příjemci. Využívají se pro komunikaci mezi vozidlem a RSU.

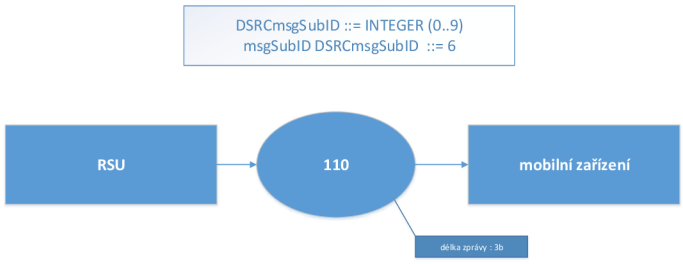
2.4.4 Kódování

ASN.1 (Abstract Syntax Notation One) je standard organizace *ITU (International Telecommunication Union)* pro formální popis datových struktur pro reprezentaci, kódování a přenos dat mezi komunikujícími systémy. Jeho hlavní uplatnění je v oblasti telekomunikací. ASN.1 definuje abstraktní syntaxi pro popis datových typů, nepředepisuje ale jak mají být konkrétní instance kódovány. K tomuto účelu se využívají tzv. kódovací pravidla, definovaná v některém z přidružených standardů. Doporučení ITU-T X.690 specifikuje tři základní přenosové formáty[16].

Základním kódováním je *BER (Basic Encoding Rules)* využívající triplet *TLV (Type Length Value)*. Typ popisuje druh tripletu. Délka určuje pozici, na které je triplet ukončen. Hodnota je popsána přímo nebo dalším tripletem. BER specifikuje jak by měla být data kódována pro přenos, nezávisle na operačním systému, programovacím jazyku, typu stroje a použité aplikaci. Ve vztahu k XML je definované kódování *XER (XML Encoding Rules)*, reprezentující ASN.1 hodnoty jako *XML (Extensible Markup Language)* data.

Posledním typem pravidel je *PER (Packed Encoding Rules)*. Na rozdíl od BER nezasílá typ. Délka s hodnotou nemusí být zasílána pokud je známa v obou stanicích. Tímto se výrazně šetří přenosové pásmo.

V systému C2X jsou ze zmíněných formátů využita dvě kódování. Pro přenos dat mezi jednotkami RSA a OBU je aplikováno kódování UPER. To bylo zvoleno z důvodu vysoké komprese, která má za následek malou velikost výsledné zprávy. Ta přímo ovlivňuje rychlost zasílání zpráv, jenž je v systému zásadní. Zejména pro složité křižovatky s velkým množstvím elementů. Zprávy zasílané z OBU do mobilního zařízení jsou standardně kódovány ve formátu XER. Kódování zpráv lze měnit v jednotce OBU. Toto nastavení je však ovlivněno verzí nahraného firmwaru.



Obrázek 2.4: PER [vlastní]

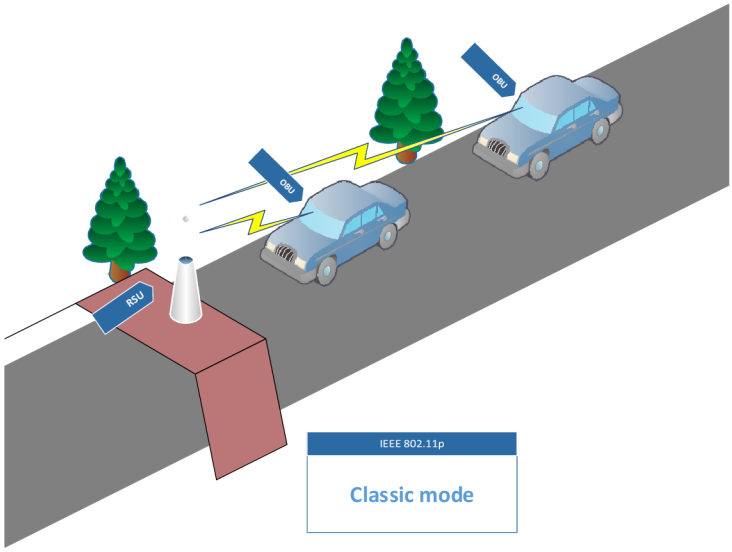
2.4.5 Jednotky

Road Side Unit

RSU je stacionární ITS stanice instalovaná u silnic. Zpravidla na semaforech, mostech a nadjezdech. Monitoruje úseky vozovky, sleduje aktuální stav počasí, zpracovává obraz a zvuk nahraný stereo kamerou. Může obsahovat laser pro detekci mlhy. V závislosti na využití může pracovat autonomně či spolupracovat s řídící stanicí. S tou může být propojena přímo či bezdrátově.

OnBoard Unit

Integrovaná univerzální palubní jednotka OBU tvořená vysílačem a přijímačem či pouze jedním z nich. Základní funkcí OBU je navázání spojení a výměna dat s RSU či jinou OBU umístěnou v blízkém vozidle. OBU proto musí být instalována na vozidle v místech s přímou viditelností (pod střechou za čelním sklem, pod zrcátkem apod.).



Obrázek 2.5: OBU - RSU [vlastní]

2.5 Komunikační technologie

2.5.1 Standard IEEE 802.11p

IEEE 802.11p je standard popisující protokol pro C2X komunikaci. Je využíván v hardwarových implementacích ITS stanic pro bezdrátovou komunikaci C2C a C2I. Vzdálenost pro přesnou lokalizaci vozidla jedoucího 50km/h byla dle předchozí analýzy určena na 50 metrů. Z toho vyplývá, že pro navázání spojení a prvotní měření je nutná vzdálenost 75 - 100 metrů. Z toho plynoucí čas na reakci při havárce je průměrně 100 - 500 milisekund. Je tedy brán velký zřetel na minimalizaci zpoždění.

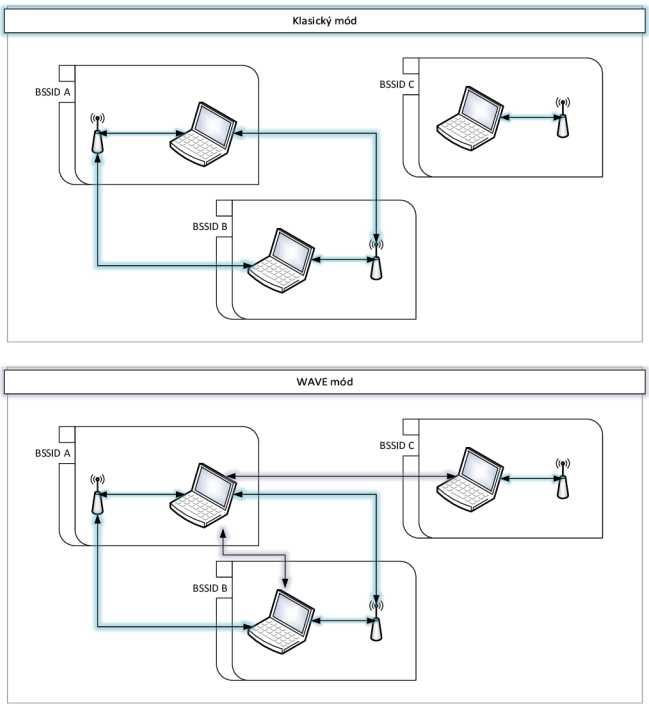
Fyzická vrstva je téměř totožná s IEEE 802.11a. Pracuje se širokopásmovou modulaci *OFDM (Orthogonal Frequency Division Multiplexing)*, využívající frekvenční dělení kanálu. Signál je vysílán na více nezávislých frekvencích, což vede k větší odolnosti proti interferencím. Subnosné vlny jsou modulovány modulací 16-QAM *(Quadrature Amplitude Modulation)*. Linková vrstva je IEEE 802.11a rozšířená o náhodnou *MAC (Media Access Control)* adresu, *QoS (Quality of Service)*, ad-hoc mód a další funkce. Uzly se připojují vždy k právě jednomu *BSS (Basic Service Set)*. Výjimkou je navázání spojení. K tomu je využito wildcard *BSSID (Basic Service Set Identification)* nastavené na hodnotu jedna. IEEE 802.11p přináší nový mód *WAVE (Wireless Access in Vehicular Environments)*, který nerozeznává rozdíl mezi uzlem a přístupovým bodem. Všechny uzly v tomto módu mají nastaveno stejné BSSID. Tím se ztrácí rozdíl mezi poskytovatelem a uživatelem. Poskytovatel při pokusu o navázání spojení zasílá *On-Demand Beacon* nesoucí všechny parametry potřebné k připojení. Při setkání vozidel s BSSID nastaveném na stejném kanálu dochází k okamžité komunikaci a výměně dat.

Klasický mód:

- přístupový bod zasílá beacon
- autentizace
- připojení uzlu

WAVE mód:

- uzly nastaví své wildcard BSSID na hodnotu jedna
- jeden z uzlů zasílá druhému On-Demand Beacon
- komunikace



Obrázek 2.6: Srovnání módů [vlastní]

Na základě testů bylo zjištěno, že jedna stanice na $20m^2$ v místech dopravních zácp či jedna stanice na $60m^2$ ve městě je dostačující. Lokalizace vozidla je určena s využitím metrik **Time of Arrival** a **Angle of arrival**.

Time of Arrival (ToA)

ToF je skupina metod sloužících k měření času potřebného pro překonání vzdálenosti. Používá se měření rozdílu časů mezi zasláním a přijetím signálu. Takto je možné určovat i velké vzdálenosti. Převážně je využíván laser, který se zamíří a vyšle paprsek na požadovaný objekt. Monitoruje čas přijmutí odraženého paprsku.

Angle of Arrival (AoA)

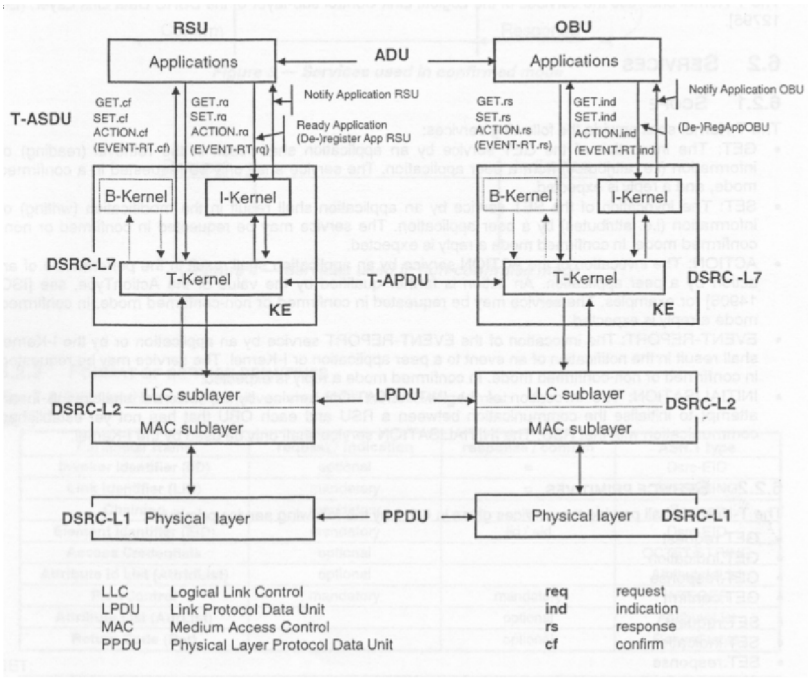
Metoda AoA využívá stanic s vybavením určujícím úhel přicházejícího signálu. Informace ze stanic jsou zasílány na mobilní přepínač, kde jsou analyzovány úhly pod kterými signál dorazil a z nich vypočtena přibližná pozice zařízení. Nevýhodou je nutnost rozšířit stanice o směrové antény.

Komunikace je rozdělena do rámců, ve kterých spolu páry uzlů komunikují a jsou lokalizovány. Kanál se dělí na tři fáze:

- detekce a správa sítě
- ToF měření (kritická na čas)
- AoA měření

2.5.2 Technologie DSRC

Dedikované spojení krátkého rozsahu *DSRC (Dedicated Short-Range Communications)* je komunikační technologie specificky navržená pro použití ve vozidlech. Umožňuje energeticky úspornou, vysoce spolehlivou a zabezpečenou datovou komunikaci mezi různými zařízeními na krátkou vzdálenost. Vozidlová jednotka OBU provádí výměnu informací se zařízením RSU, ke kterému se přiblížila. Dochází k výměně dat a lokalizaci vozidla. V Evropě a Japonsku je standardizovaná frekvence 5,8 GHz. V Severní Americe je využívána frekvence 5,9 GHz. DSRC spektrum je rozděleno na sedm 10 MHz širokých kanálů. Existují dva typy kanálů: kontrolní kanál CCH a služební kanál *SCH (Service Channel)*. Kontrolním kanálem proudí pouze zabezpečená komunikace. Služební kanál již zabezpečen být nemusí. V současnosti se technologie DSRC aktivně využívá převážně k výběru mýta.



Obrázek 2.7: Struktura DSRC [převzato [12]]

2.5.3 Bezpečnost

Přes veškeré výhody C2X jsou aplikace využívající zasílání zpráv náchylné k možným útokům. Padělání varovných zpráv může mít fatální následky pro bezpečnost na silnici. Všesměrové zasílání zpráv zjednodušuje útočníkům práci se vstupem do systému. Zvýšená bezpečnost je tedy v tomto ohledu nutná. Proto je zavedena v tomto odvětví množina bezpečnostních mechanizmů *PET (Privacy Enhancing Technologies)*[9]. Primárním cílem je zabezpečení přijímaných zpráv a zajištění privátnosti zasílaných dat. Hlavní bezpečnostní hrozby, slabiny a rizika jsou standardizovány institutem ETSI.

2.6 Ukládání prostorových dat

V různých odvětvích se naskytuje potřeba zpracovat geometrické, geografické či prostorové data. Pro tento přístup je charakteristická správa obrovských kolekcí relativně malých objektů, jako jsou polygony. Tuto správu zajišťují prostorové databázové systémy[8], které lze chápat takto:

1. prostorový databázový systém je databázový systém
2. nabízí prostorové datové typy ve vlastním datovém modelu a s vlastním dotazovacím jazykem
3. podporuje vlastní implementaci prostorových dat, indexování a efektivní algoritmy k prostorovému spojování

Prostorovými datovými typy chápeme například bod, úsečku či region. Tyto objekty mají své vlastnosti, jsou mezi nimi vztahy a mohou na ně býti aplikovány operace.

2.6.1 R* strom

Prostorové spojování je nejdůležitější operací pro kombinaci prostorových objektů, jenž mají navzájem různé vztahy. Z důvodů velkého množství objektů je efektivnost spojování kritickou vlastností. Za tímto účelem jsou nejpoužívanější R stromy, které jsou vhodné pro podporu prostorových dotazů[7].

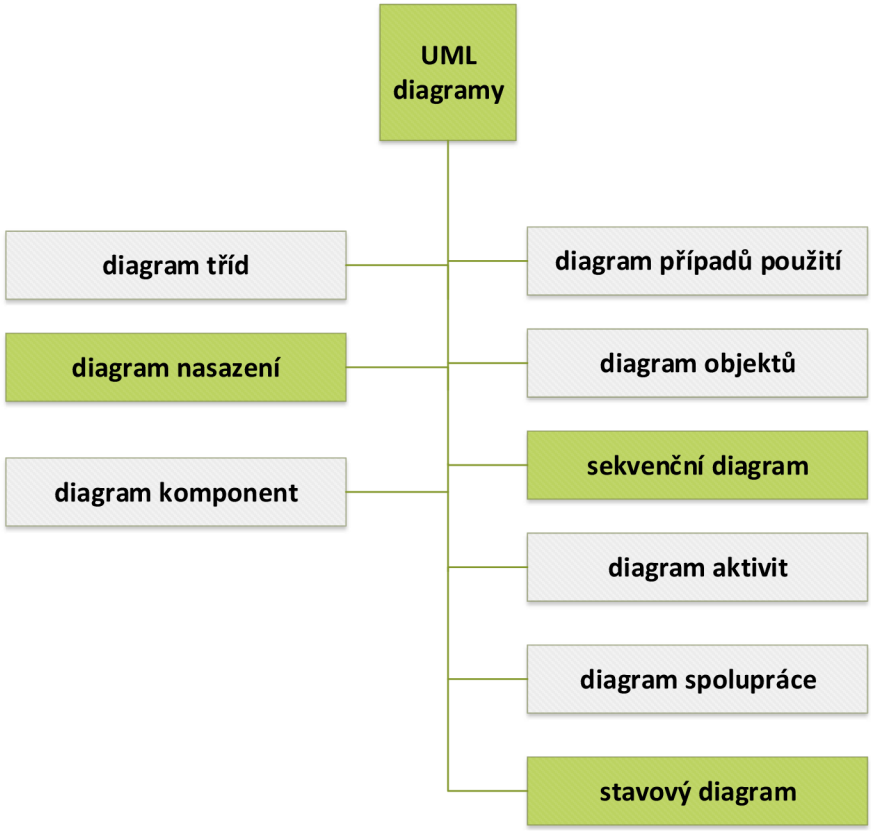
R stromy jsou hierarchické datové struktury založené na B^+ stromech[14]. Využívají se k dynamické organizaci množiny bodů, ohraničených hraničními obdélníky *MBR (Minimum Bounding Rectangles)*. Každému uzlu stromu odpovídá některý MBR. Listové uzly stromu obsahují ukazatele na databázové objekty. MBR, které zahrnují odlišné uzly, se mohou překrývat.

Jednou z nejefektivnějších odnoží rodiny R stromu jsou R* stromy[6]. R* strom používá sofistikovanější algoritmy pro vkládání a rozdělování, než klasický R strom. Datová struktura zůstává bez rozdílu. R* stromy byly zvoleny pro implementaci do aplikace.

2.7 UML diagramy

Jazyk *UML (Unified Model Language)* je univerzální jazyk pro vizuální modelování systémů. Byl navržen ke spojení nejlepších existujících postupů modelovacích technik a softwarového inženýrství. Poskytuje vizuální syntaxi, která byla dále využita při sestavování modelů. UML vytváří svět jako systém vzájemně se ovlivňujících objektů. Objekt je soudržné seskupení dat a metod [5].

Model UML popisuje statickou strukturu a dynamické chování. Základními stavebními bloky jsou předměty (elementy), vztahy (spojnice) a diagramy (pohledy). Celkem existuje 9 typů diagramů a všechny jsou znázorněny na obrázku 2.8. V této práci budou pro názornou ilustraci daných řešení využity stavové diagramy, diagramy nasazení a sekvenční diagramy.



Obrázek 2.8: UML diagramy [vlastní]

Kapitola 3

Návrh komunikačního rozhraní

Následující kapitola popisuje technologie, platformy a přístupy zvolené k realizaci grafické vizualizace topologie křižovatky pro mobilní zařízení s ohledem na měnící se dynamickou povahu dat. Z technologií popsaných v předešlé kapitole vybírá ty, které nejlépe vyhovují zadaným požadavkům, jsou dostupné pro potřeby vývoje a jdou využít v daných firemních podmínkách. Dále se zabývá již konkrétními návrhy dílčích částí výsledné aplikace.

První část kapitoly seznamuje s prostředím vývoje, vstupními podmínkami, zdroji a omezeními. Na základě toho popisuje využité technologie a důvody k těmto volbám. V druhé části se zaměřuje na struktury aplikace. Za tímto účelem jsou využity adekvátní diagramy UML.

3.1 Zadavatelská společnost

Tato práce byla zadána a vypracovávána pro společnost Siemens s.r.o.[1]. Hlavním požadavkem bylo vytvoření mobilní aplikace, jenž dokáže komunikovat s jednotkou RSU umístěnou v blízkosti křižovatky a s jednotkou OBU instalovanou ve vozidle. Zprávy obdržené z těchto jednotek dekóduje a získaná data vizualizuje na displeji připojeného zařízení. Tento proces nabídne zpětnou validaci a usnadní práci konstruktérům topologií křižovatek, jenž pracují se softwarem *Sitraffic*. Aplikace nabízí uživatelsky přívětivou grafickou demonstraci nově se rozvíjejícího systému C2X a v neposlední řadě slouží řidičům na dosavadních okruzích k testování nasazení ITS jednotek v reálném prostředí.

Společnost Siemens IT Solutions and Services je hlavním inovátorem a technologickým lídrem v oblasti softwarového vývoje C2C komunikací. Spojuje zkušenosti ze strojního a automobilového průmyslu se znalostmi z oboru komunikačních technologií a vestavěných systémů. S těmito předpoklady se Siemens IT Solutions an Services z většinové části podílí na vývoji C2C komunikačního rozhraní, jenž bude v budoucnosti efektivně spolupracovat s aplikacemi systému C2X. Komunikace je realizována radiovými technologiemi, které zaručují výměnu dat mezi vozidly a ostatními body dopravní infrastruktury v reálném čase. Technologie jsou založeny na mezinárodních standardech podporovaných výrobci automobilů a jejich dodavateli. Díky širokému spektru působení byla vytvořena kompletní podpora od prvotních konzultací až po finální nasazení systému a jeho dlouhodobou správu.

[1] `www.siemens.com`

Hlavní cíle:

- stabilní a bezpečná C2X technologie
  - bezpečnost (anonymita a důvěryhodnost přenášených dat)
  - nízká latence (okamžitý přenos informací)
  - vysoká spolehlivost (spolehlivá komunikace)
  - rychlost (neprodlení navázání komunikace)
- integrované služby založené na nové technologii
- cena (zachování nízké ceny za služby)
- standardizace

Pro účel vývoje aplikace byla k dispozici poskytnuta jednotka RSU a jednotka OBU. Jednotky se chovají jako směrovače a umožňují komunikaci s okolním světem prostřednictvím protokolů *TCP (Transmission Control Protocol)* a *UDP (User Datagram Protocol)*. Nenabízejí však již připojení k sítí Internet.

3.2 Platforma

Stěžejním bodem vývoje byl výběr podporované mobilní platformy. Z průzkumu podílu trhu pro třetí čtvrtinu roku 2014, vytvořeného organizací Strategy Analytics[2], vyplývá drtivá dominance platformy Android. Operační systém společnosti Google je nainstalován na 84% všech smartphonů. Kromě tohoto faktu společnost Google představila v červnu 2014 telematický standard Android Auto. Vzhledem k těmto skutečnostem byl jako cílový operační systém zvolen Android.

Global Smartphone Operating System Marketshare %	Q3 '13	Q3 '14
Android	81.4%	83.6%
Apple iOS	13.4%	12.3%
Microsoft	4.1%	3.3%
BlackBerry	1.0%	0.7%
Others	0.2%	0.1%
Total	100.0%	100.0%

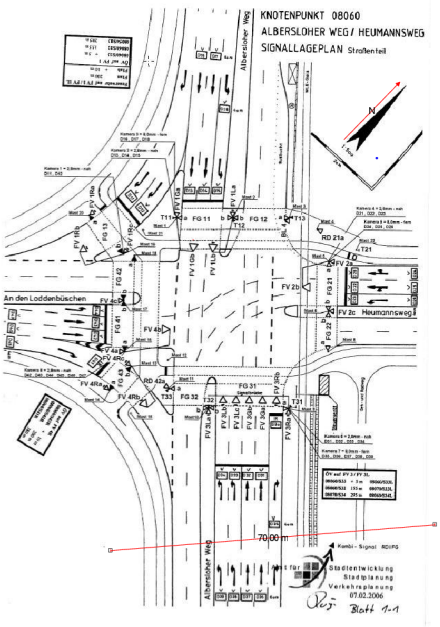
Obrázek 3.1: Tržní podíl [převzato [15]]

[2] `www.strategyanalytics.com`

3.3 Proces vytváření křižovatky

Proces tvorby křižovatky a nastavení jejich světelných lamp se skládá z dílčích části. Pro zjednodušení jsou v textu uvedeny pouze části, jenž lze graficky reprezentovat. Jako demonstrace byla použita vzorová křižovatka Knotenpunk 08060, zobrazující ulice Heumannsweg a Albersloher Weg. Ty jsou situovány v jihovýchodní část německého města Münster, jenž se nachází v severní části spolkové země Severní Porýní-Vestfálsko.

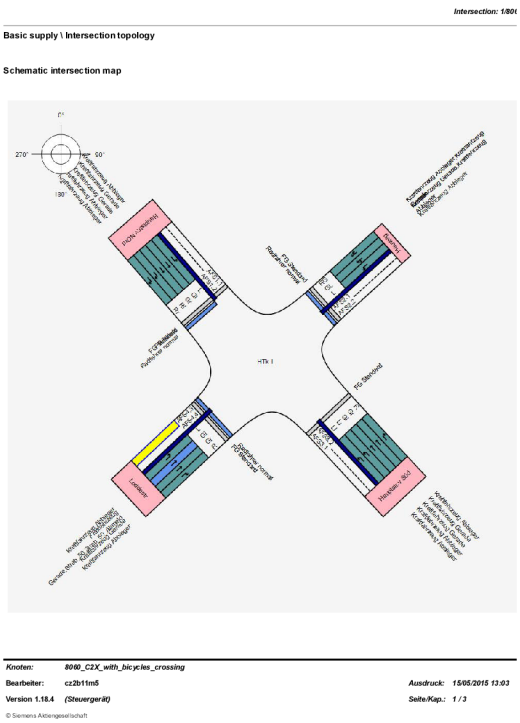
Prvním milníkem byl nákres topografie křižovatky. V něm jsou popsány jednotlivé přechody pro chodce a silniční pruhy, které jsou navzájem propojeny. Silnicím vstupujícím do křižovatky jsou přiděleny světelné semafory. Detaily jsou patrné na obrázku 3.2.



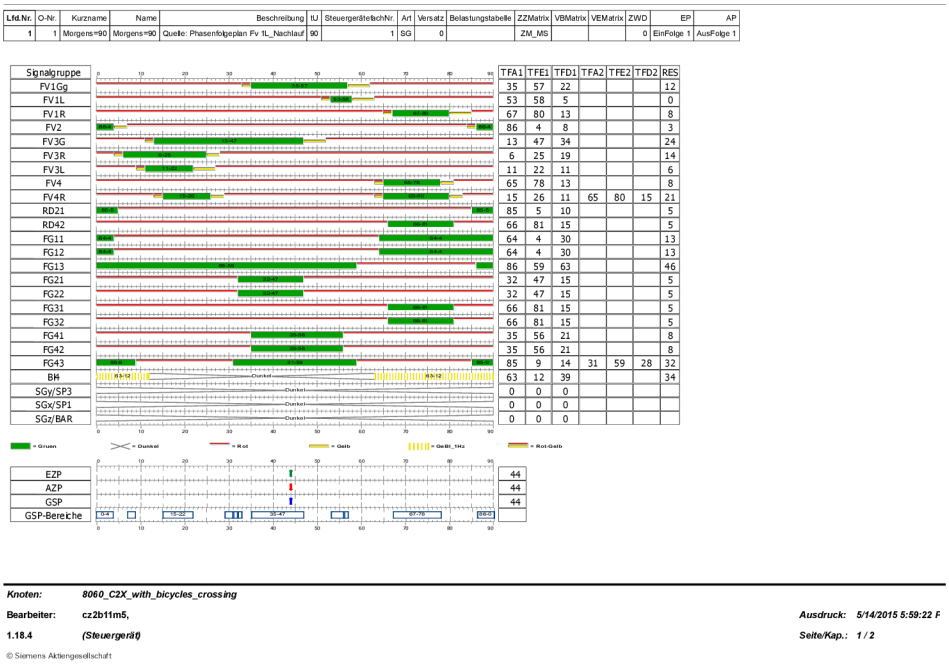
Obrázek 3.2: Návrh topologie křižovatky 08060 [výstup programu]

Informace získané z návrhu jsou vloženy do softwarového editoru křižovatek *Sitraffic*. Ten vytváří moduly pro kontrolu dopravního provozu. Náhled na takto editovanou křižovatku 08060 lze vidět na obrázku 3.3.

Editor *Sitetraffic* také nabízí možnost definice světelných lamp. Ty mohou býti definovány staticky či dynamicky. Při statickém definování je vytvořena sada scénářů pokrývajících různě části dne. Takto lze přidělit křižovatce jedno chování v době dopravní špičky a druhé chování uprostřed noci. Každý scénář popisuje sadu lamp a jejich stavů pro daný časový úsek. Příklad je znázorněn na obrázku 3.4.



Obrázek 3.3: Editor topologie křižovatky 08060 [výstup programu]



Obrázek 3.4: Signální plán křižovatky 08060 [výstup programu]

3.4 Typ zpráv

Z nabízených druhů zpráv distribuovaných jednotkami RSU a OBU[2.4.2] byly zvoleny zprávy typu MAP a SPAT. MAP detailně popisuje objekty dopravní infrastruktury. Uchovává informace o lokalitě pokryté oblasti, na které se nachází úseky silnic a křižovatky. Jeho reprezentace definuje statickou strukturu dat, jenž není časově kritická vůči změnám. SPAT dodává informace o aktuální situaci světelných křižovatek. Data se dynamicky mění a jsou zasílána zdrojovou jednotkou v krátkých časových intervalech. Oba typy zpráv je možné manuálně vytvořit a pro účely simulace přímo nahrát do jednotek RSU a OBU. Při vytváření je třeba věnovat pozornost jejich ASN.1 reprezentacím. Data jsou mezi uzly zasílána ve formátu XML. Pro účely vývoje nejsou přenášené informace dále kódovány a mají podobu prostého textu.

```
SPAT ::= SEQUENCE {
    msgID DSRCmsgDI, -- ID zprávy
    msgSubID DSRCmsgSubID OPTIONAL, -- subID zprávy
    name DescriptiveName OPTIONAL, -- jméno kolekce (pro debug)
    intersections SEQUENCE (SIZE(1..32)) OF
                        IntersectionState,
                        -- množina SPAT dat (pro každou křižovatku zvlášť)
    regional RegionalSPAT OPTIONAL,
                    -- regionální data
    ..., -- lokální data
    }
```

Zdrojový kód 3.1: ASN.1 reprezentace zprávy SPAT

```
MapData ::= SEQUENCE {
    msgID DSRCmsgDI, -- ID zprávy
    msgSubID DSRCmsgSubID OPTIONAL, -- subID zprávy
    msgIssueRevision MsgCount, -- číslo revize
    layerType LayerType OPTIONAL, -- typ vrstvy
    layerID LayerID OPTIONAL, -- ID vrstvy
    intersectionCount Count OPTIONAL, -- počet křižovatek (požadavek Japonska)
    intersections SEQUENCE (SIZE(1..32)) OF
                        Intersection OPTIONAL,
                        -- definice křižovatek
    roadSegments SEQUENCE (SIZE(1..32)) OF
                        RoadSegment OPTIONAL,
                        -- popisy segmentu silnice
    dataParameters DataParameters OPTIONAL,
                    -- metadata a˜volitelné parametry
    restrictionList RestrictionClassList OPTIONAL,
                    -- omezení
    regional RegionalMapData OPTIONAL,
                    -- regionální data
    ..., -- lokální data
    crc MsgCRC OPTIONAL
                    -- kontrolní součet
    }
```

Zdrojový kód 3.2: ASN.1 reprezentace zprávy MAP

3.5 Syntaktický rozbor zprávy

Ke zpracování dat v aplikaci je využíván parser. Jde o program, který kontroluje syntaktickou správnost dokumentu. Pro čtení XML dokumentu existují dva základní přístupy, *DOM (Document Object Model)* a *SAX (Simple API for XML)*. DOM převádí celý dokument na strom objektů v paměti. SAX při načítání dokumentu generuje sérii volání funkcí.

DOM

Rozhraní DOM je standardní rozhraní pro zpracování dokumentů XML. Definuje způsob mapování dokumentu na hierarchii objektů v paměti. Každému elementu odpovídá v paměti jeden objekt. Název elementu, obsah a další informace zjišťujeme pomocí dostupných metod a vlastností.

SAX

Na rozdíl od rozhraní DOM se hodí ke čtení rozsáhlých XML dokumentů. Ty se načítají sekvenčně a tudíž nezaplňují paměť jako celek. Během čtení se průběžně předávají informace o tom co se v dokumentu nachází. SAX parser pro každý element vyvolává nadefinovanou událost, jejímiž parametry jsou jména značek, text, atributy apod. Během zpracování dokumentu dochází nepřímo k obsluze událostí.

Aplikace si klade za cíl průběžně zpracovávat přijaté zprávy s minimálním časovým zpožděním a malou paměťovou náročností. Z tohoto důvodu byl zvolen přístup skrze parser SAX, jehož výhody převládají nad DOMem.

3.6 Přenos zpráv

Dostupné jednotky RSU a OBU nabízí tři přístupy k distribuci dat do cílového mobilního zařízení. Každý z nich má své výhody, ale zároveň i stinné stránky, které budou v následujících sekcích diskutovány.

Síťový mód

V síťovém módu se mobilní zařízení po přiblížení k jednotce OBU na vzdálenost pokrytou Wi-Fi vysílačem připojí k jednotce. Zasláním úvodní zprávy si vyžádá registraci. V úvodní zprávě předá parametry požadované typy zpráv. OBU poté zasílá MAP a SPAT v pravidelných intervalech na zaregistrovaná zařízení. Celý proces bude blíže specifikován v kapitole 4.

Výhody:

- zařízení má volný Bluetooth kanál (lze připojit headset)

Nevýhody:

- zařízení nedokáže navázat simultánní spojení s jinou sítí (přetrvává v offline režimu)
- zobrazení mapových podkladů není podporováno

Bluetooth mód

Bluetooth mód využívá komunikace mobilního zařízení osazeného bluetooth čipem s jednotkou OBU nainstalovanou v palubní desce. Přenosné zařízení po nastoupení do vozu ustanoví spojení a spáruje se s OBU.

Výhody:

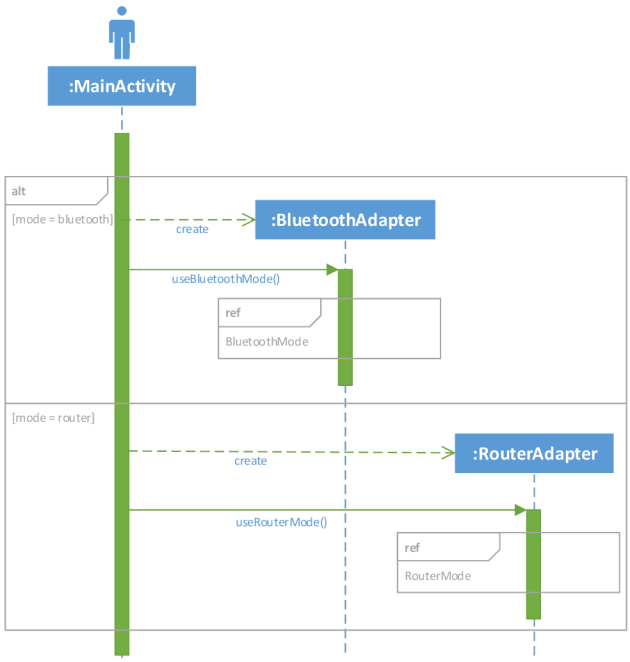
- během spojení může být mobilní zařízení online a dynamicky stahovat mapové podklady pro aktuálně zobrazované prostory

Nevýhody:

- OBU i mobilní zařízení musí mít přítomný a správně nakonfigurovaný bluetooth adaptér
- mobilní zařízení má zaneprázdněný Bluetooth kanál (nelze připojit headset)

Cloud mód

Cloud mód byl třetí vývojovou větví, skrze kterou aplikace přijímá data. Z důvodu komplikací při nasazování serverové části cloudu ze strany společnosti Siemens, nebyl mód ve finální verzi implementován.



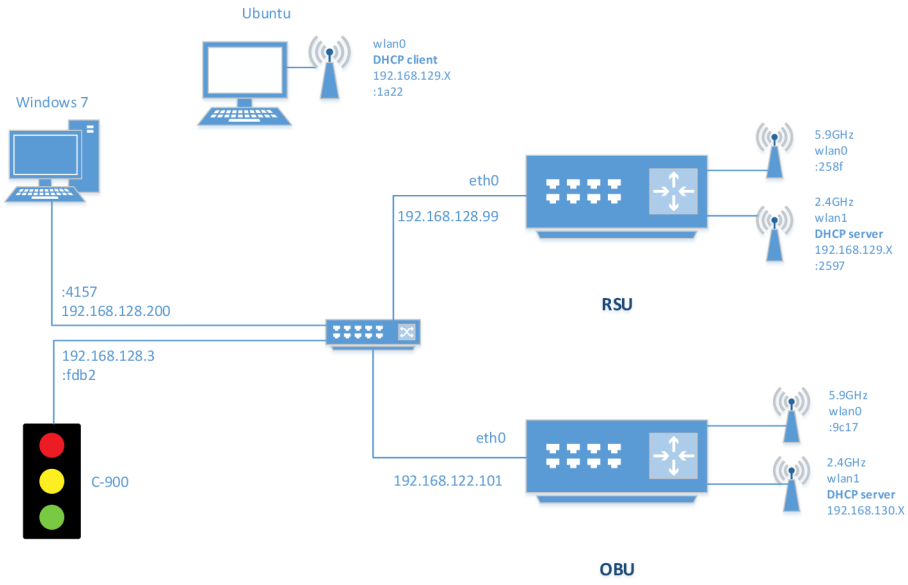
Obrázek 3.5: Sekvenční diagram výběru módu [vlastní]

3.7 Topologie

Za účelem testování nabízí zadavatelská společnost Siemens v jejích kancelářských prostorách jednotky RSU a OBU. Obě jednotky plní roli přístupového bodu k bezdrátové Wi-Fi

síti. Běží na nich DHCP server přidělující IP adresy. Jednotka RSU v sobě uchovává informace o topologii křižovatky, ke které je připojena. Stejně tomu tak je i v reálném dopravním prostředí.

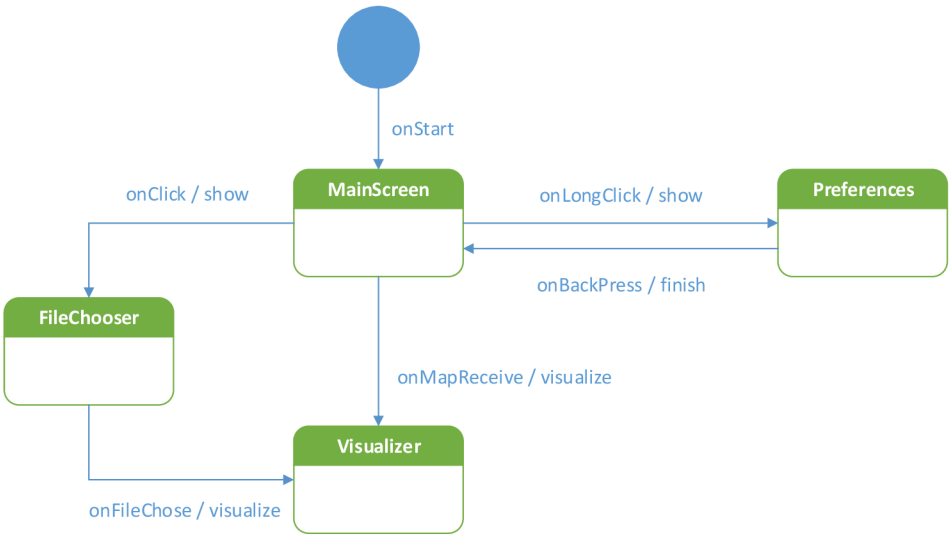
Jednotka OBU se připojuje k RSU a přijímá zprávy MAP a SPAT, které přeposílá na všechny IP adresy, jenž má uložené v seznamu. Adresy jsou do seznamu přidávány při registraci zařízení. Po vypršení přiděleného intervalu či restartu jednotky jsou adresy ze seznamu odstraněny. Pro účely simulace a testování jsou jednotky RSU, OBU a další objekty zapojeny dle topologie zakreslené na obrázku 3.6.



Obrázek 3.6: Topologie zapojení jednotek v prostorách společnosti Siemens [vlastní]

3.8 Struktura aplikace

Při návrhu aplikace byl kladen důraz na jednoduché, intuitivní a rychlé ovládání bez zbytečných prvků odvádějících pozornost. Proto byla snaha se v aplikaci vyvarovat komplexních nastavení a mnoha zanořených nabídek. Výsledný návrh byl po konzultaci s pracovníky společnosti Siemens koncipován do čtyř základních pohledů. Při startu aplikace so zobrazí uvítací obrazovka s logem zadavatelské společnosti. Ve výchozím nastavení program provede připojení k palubní jednotce a začne naslouchat. Po přijetí a zpracování zprávy s topologií se aplikace přepne do hlavního pohledu, jenž vizualizuje přijatá data. Tohoto pohledu lze taktéž dosáhnout kliknutím na logo společnosti na úvodní obrazovce. Tato akce nabídne ruční výběr zprávy již uložené v zařízení. Po dlouhém kliknutí na logo dojde k vyvolání čtvrtého pohledu, nabízejícího uživatelského nastavení. Jednotlivé přechody mezi pohledy vizualizuje schéma 3.7.



Obrázek 3.7: Stavový diagram mobilní aplikace [vlastní]

Kapitola 4

Implementace aplikace pro vizualizaci dopravních dat

Následující kapitola popisuje implementaci celé aplikace, použité konkrétní metody a přístupy a směr celého návrhu. Kapitola je rozdělena na dílčí sekce, popisující jednotlivé části aplikace.

Kompletní vývoj aplikace probíhal ve spolupráci se společností Siemens. Při první návštěvě společnosti byl ustanoven časový plán celého projektu. Během prvního roku bylo naplánováno vytvořit funkční prototyp aplikace, která bude vizualizovat statickou topologii křižovatky. Cílem bylo sestrojit prototyp, jenž by mohla společnost Siemens prezentovat zájemcům o systém C2X. Pro druhý rok bylo naplánováno zajistit plynulou komunikaci mezi všemi zařízeními a vizualizovat dynamická data (semafory, časovače), která bylo nutno propojit se stávající topologií.

Veškerá komunikace probíhala se specialistou na systém C2X, panem Pascalem Cassinellim. Během celého vývoje probíhali pravidelné mítinky. Ty byly od března 2014 zapisovány a zakládány. Kompletní zápisy z mítinků se nacházejí na přiloženém CD, odkazovaném v příloze A

4.1 Vývojové prostředí

V současné době vývoj mobilních aplikaci pro systém Android podporují dvě vývojová prostředí. Prvním z nich je *Eclipse IDE (Integrated Development Environment)*, rozšířené o zásuvný modul *ADT (Android Development Tools)*. V prosinci 2014 společnost Google publikovala oficiální software určený k vývoji aplikací, nazvaný Android Studio. Jako vývojové prostředí bylo zvoleno Eclipse, jelikož v prvním roce vývoje se jednalo o jediné podporované prostředí.

4.2 Inicializace

Po prvním spuštění se provádí prvotní inicializace vstupní konfigurace, uživatelských nastavení a načtení dat pro komunikaci se serverem. Tato perzistentní data jsou ukládána skrze obecné Android rozhraní `Shared Preference`[1]. Toto rozhraní je určeno pro ukládání a získávání párů klíč-hodnota primitivních datových typů. Uživatel zařízení může

[1] `http://developer.android.com/guide/topics/data/data-storage.html#pref`

skrze nastavení Androidu kdykoliv uloženou konfiguraci vymazat, vynutit opětovné načtení konfigurace a simulovat tím tak prvotní spuštění aplikace. Data jsou rovněž smazána po odinstalování aplikace. Z těchto důvodů byl pro ukládání nastavení a konfigurací zvolen tento přístup. Touto metodou jsou získány informace o poslední nastavené IP adrese, portu, času pro pokusy o připojení a nastavení výchozích grafických prvků a podkladové mapy. Rovněž lze aplikaci spustit ve vývojovém režimu, který poskytuje detailní informace o aktuálním stavu, fázi připojování a zpracovávaných datech formou kontrolních výpisů do protokolového mechanizmu Logcat[2], jenž přináší samotný systém Android.

4.2.1 Módy aplikace

V případě, že jde o první spuštění (data nejsou poskytnuta), se aplikace chová dle upraveného scénáře. Po načtení dat je zvolen pracovní mód. Na výběr je mezi *RouterMode* a *BluetoothMode*. Tyto módy určují jakým způsobem bude probíhat komunikace se serverem. V případě *RouterMode* zařízení komunikuje výměnou paketů skrze otevřený soket. V rámci *BluetoothMode* dojde ke spárování zařízení s jednotkou OBU, jenž je fyzicky přítomna v automobilu a data jsou přenášena bezdrátovou technologií Bluetooth. Oba tyto módy mají své výhody a nevýhody, které jsou popsány v sekci 3.6.

Na jednotce OBU, kde probíhalo majoritní testování, běžel po čas vývoje firmware, jež nepodporuje připojení skrze technologii Bluetooth. Z důvodu tohoto omezení byl primárně vyvíjen a testován režim *RouterMode*. Pokud nebude zmíněno jinak, tak následující text této práce bude popisovat činnost právě v režimu *RouterMode*.

4.2.2 Úvodní obrazovka

Po spuštění je uživatelovi zobrazena úvodní obrazovka s logem zadavatelské společnosti Siemens. Při zapnutém vývojářském módu je zobrazena aktuální verze aplikace a stav připojení.

Obrázek 4.1: Úvodní obrazovka [snímek obrazovky]

[2] `http://developer.android.com/tools/help/logcat.html`

4.2.3 Připojovací konfigurace

Při prvotní inicializaci jsou skrze rozhraní `Shared Preference` načteny informace o serveru, ke kterému byla aplikace naposled připojena. Data jsou uchovávána v objektu `NetworkData` a je k nim přistupováno skrze kontrolér `NetworkDataHolder`. Pokud jsou tyto informace zastaralé či jde o první spuštění, je nutné tato připojovací data získat z externích zdrojů.

Zisk IP adresy

Pro správný běh aplikace je nutné připojení k jednotce OBU, která zařízení přeposílá data přijatá ze semaforů. Jednotka standardně funguje jako přístupový bod bezdrátové sítě Wi-Fi. Na jednotce běží *DHCP (Dynamic Host Configuration Protocol)* server. Ten přidělí zařízení na omezenou dobu pomocí DHCP protokolu IP adresu, masku sítě a implicitní bránu. Aplikace přistoupí skrze systémový kontrolér `WifiManager` k informacím poskytnutým DHCP serverem a získá z nich výchozí bránu, která je rovněž IP adresou požadovaného serveru.

Zisk portu

Pokud není port specifikován v počáteční konfiguraci, je uživatel vyzván k ručnímu vložení čísla portu prostřednictvím dialogového okna. Dialog vyžaduje na svém vstupu až šestimístný numerický kód reprezentující port pro připojení.

4.2.4 Navázání komunikace s jednotkou

Jednotka OBU ve své standardní konfiguraci přijímá zprávy MAP a SPAT z blízkých jednotek RSU umístěných podél vozovky. Přijatá data přeposílá zaregistrovaným zařízením. Samotná registrace zařízení funguje ve třech krocích:

1. vytvoření registrační zprávy
2. zaslání zprávy jednotce OBU
3. přijetí potvrzení o úspěšné registraci

V prvním kroku je na cílovém zařízení vytvořena zpráva ve formátu XML. Ta má pevně danou strukturu. Obsahuje dva elementy, které jsou proměnlivé v závislosti na požadavcích aplikace. Element `seqno` definuje pořadové číslo zprávy a je pro každou zprávu inkrementován. Element `reg-msg` obsahuje šestibitové pole. Hodnota jednotlivých bitů určuje typ zpráv, pro které si zařízení registruje odběr. Pro účely aplikace je registrován odběr zpráv typu MAP a SPAT, reprezentovaný jako *001101*. Příklad zprávy můžeme vidět na obrázku 4.1.

```
<ItsXferMessage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="its.xsd">
<seqno>800</seqno>
    <content>
        <xfer-cfg-request>
                        <reg-msg>001101</reg-msg>
            </xfer-cfg-request>
    </content>
</ItsXferMessage>
```

Zdrojový kód 4.1: Příklad registrační zprávy

Pro ušetření času je sestavena vzorová XML zpráva. Ta je uložená v souborech aplikace. K navázání spojení je vytvořeno pracovní vlákno, které vytváří síťový soket a jednorázově zasílá zprávu na definovanou IP adresu. Jednotka OBU po přijetí zprávy ukládá IP adresu do privátního seznamu adres registrovaných k odběru a posílá nazpět potvrzení o registraci. Poté dle definovaných hodnot zasílá zařízením ze seznamu konkrétní zprávy v jimi nastavených intervalech. V průběhu vývoje aplikace se měnil a aktualizoval firmware nahraný v jednotce OBU. Z tohoto důvodu se původně povinná potvrzující zpráva stala volitelným parametrem. Aplikace kvůli zajištění zpětné kompatibility pracuje s oběma případy.

4.3 Hlavní vlákno

Hlavní činností aplikace je příjem dat z jednotky OBU a jejich grafická vizualizace. Tuto činnost zajišťuje samostatné vlákno. To naslouchá na otevřeném portu a přijímá zprávy zaslané jednotkou. Aplikace rozeznává dva typy zpráv, které přicházejí v odlišných intervalech. Zprávy jsou rozdělovány a syntakticky rozebírány na dílčí části, jenž se dále zpracovávají.

4.3.1 Příjem zpráv

Pro odběr zpráv je vlákno vytvořeno přímo jako instance třídy *Thread*, přičemž kontruktoru je předána instance třídy `RouterComm`, implementující rozhraní *java.lang.Runnable*. Toto rozhraní obsahuje pouze metodu *run()*. Vlákno opakovaně naslouchá a přijímá datagramy v soketu otevřeném při inicializaci. Pokud vlákno čekalo, je vyvolána výjimka *ThreadInterruptedException* a vlákno se rozběhne.

Obsah datagramu je uložen jako pole bajtů. To se převádí na řetězec reprezentující samotnou zprávu. Zpráva je prohledávána na výskyt párových elementů označovaných značkami <*map*> a <*spat*>. Ty určují jakým způsobem se bude dále se zprávou nakládat.

4.3.2 Zpracování zprávy MAP

Zpráva MAP je ve výchozím nastavení zasílána ve třicetisekundových intervalech. Pro snížení výpočetní náročnosti a za demonstračním účelem aplikace je zpracovávána jednorázově při prvním přijetí. Zpracování obstarává aktivita `Visualizer`.

Aktivitě je po jejím spuštění skrze objekt *Bundle* předán řetězec s přijatou zprávou. V případě, že je aktivita vyvolána staticky, je předána proměnná s definovanou absolutní cestou k XML souboru se zprávou MAP.

Uživatelské prostředí v platformě Android se skládá z hierarchie objektů - tzv. pohledů (*View*). Aktivitě je nastaven vlastní pohled `VisualizerView`, definovaný níže. Za testovacími účely je vytvořena instance třídy `StoplightSimulation`, jenž simuluje dynamicky se měnící světelné stavy křižovatek v testovacím módu aplikace, kdy není přítomna jednotka OBU. Pro správu doteků a gest je vytvořena instance třídy `MT` (MultiTouch), implementující rozhraní *OnTouchListener*, které implementuje metodu `onTouch(View a, MotionEvent event)`. Té jsou jejími parametry předány informace o dotecích a jejich souřadicích. Rozpoznané pozice doteků jsou porovnávány s pozicemi objektů uložených v databázi a v případě shody se aplikace chová dle scénáře předvoleného pro daný objekt.

Zpráva MAP je zpracovávána třídou `XMLHandler`, které je předávána jako parametr konstruktoru. Pro syntaktický rozbor je využito rozhraní SAX, umožňující sériový přístup ke XML, jenž je definováno v sekci 3.5. Využívá proudové zpracování, při kterém se dokument rozdělí na jednotlivé jeho části (počáteční a koncové značky, obsahy elementů, komentáře, atd.). Postupně se pak volají jednotlivé události, které ohlašují nalezení konkrétní části.

SAX parser postupně prochází dokument a za pomoci třídy `DataHandler` vyvolává metody :

- *startDocument()*: vytváří kořenový objekt `Data` 4.5.1
- *endDocument()*: kontroluje připojení k síti a vytváří instance tříd `GoogleTile` a `OSMTile`, zpracovávajících podkladové mapy
- *startElement(String namespaceURI, String localName, String qName, Attributes atts)*: nastavuje identifikátory označující aktuální zanoření v elementu
- *endElement(String namespaceURI, String localName, String qName)*: nastavuje identifikátorům negativní hodnoty
- *characters(char ch[], int start, int length)*: zpracovává data obsažená v elementech a utváří hiearchii objektů

```
@Override
public void characters(char ch[], int start, int length) {
  String chars = new String(ch, start, length);
 chars = chars.trim();

  if (_inHeader && _inProtocolVer) {
    _data.header.setProtocolVersion(Integer.valueOf(chars));
  } else if (_inHeader && _inMessageId) {
    _data.header.setMessageId(Integer.valueOf(chars));
  } else if (_inHeader && _inStationId) {
    _data.header.setStationId(Integer.valueOf(chars));
  }
}
```

Zdrojový kód 4.2: Příklad MAP parseru

Struktura zprávy MAP prošla za čas vývoje aplikace řadou zásadních změn, které bylo třeba reflektovat vydáním nové verze. Ta musela zpracovávat nejnovější formát avšak taktéž splňovat zpětnou kompatibilitu.

4.3.3 Zpracování zprávy SPAT

Zpracování probíhá obdobně jako u zpráv typu MAP. S rozdílem toho, že při zpracování kořenového elementu je vytvořeno pole objektů **MovementEvent** 4.5.7. To je definováno jako seznam *CopyOnWriteArrayList*, který je bezpečný z hlediska mezivláknové komunikace.

4.3.4 Správa gest

K zajištění interakce uživatele s aplikací formou dotyků je v aktivitě `Visualizer` vytvořena instance třídy `MT`, implementující rozhraní *OnTouchListener*. Její callback metoda *onTouch(View a, MotionEvent event)* navrací souřadnice bodů na displeji, kterých se uživatel dotknul. Při shodě souřadnic s blízkým okolím uzlu dojde ke zvolení daného uzlu a zobrazení jemu odpovídajících detailů v informačním okně. Taktéž se po zvolení začnou vykreslovat semafory svázané se signální skupinou, jenž spadá pod vybraný uzel. Více v sekci 4.8.

Správu gest zajišťuje instance třídy `ScaleListener` rozšiřující třídu *ScaleGestureDetector*. Ta umožňuje gesty přibližovat a oddalovat celou topologii. Ke změně úrovně přiblížení mapových podkladů byl vytvořen posuvník umístěný v levé části obrazovky.

4.4 Zpracování polohy

K informacím o poloze lze přistoupit jako k datům obsaženým mezi párovými elementy *<lat></lat>* a *<long></long>*, zanořenými v rodičovském elementu *<refPoint></refPoint>*. Pro korektní zpracování je třeba přijaté informace o souřadnicích zpracovat z poskytnutého řetězce do formátu *DMS (Degrees-minutes-seconds)*.

Řetězec poskytnutý mezi elementy je celočíselného datového formátu. Dle verze specifikace je třeba jej převést do formátu desetinných stupňů *DD (Decimal Degrees)*. Rozdělení se provádí na základě verze zprávy. V aktuální verzi má poskytnutý řetězec přesnost 7 desetinných míst. Se znalostí této informace lze pak řetězec rozdělit na celočíselnou část a desetinnou část. Dále je aplikován následující postup ke konverzi na stupně, minuty a sekundy dle rovnic 4.1, 4.2, 4.3 a 4.4.

Jeden stupeň (°) se rovná 60 minutám (’), což se rovná 3600 sekundám (”).

$$1^\circ = 60' = 3600'' \tag{4.1}$$

Stupně D se rovnají celočíselné části desetinných stupňů DD.

$$d = integer(dd) \tag{4.2}$$

Minuty M se rovnají rozdílu celočíselné části desetinných stupňů DD a stupňů D vynásobeném 60.

$$m = integer((dd - d) \times 60) \tag{4.3}$$

Sekundy S se rovnají rozdílu celočíselné části desetinných stupňů DD, stupňů D a podílu minut a 60, vynásobeném 3600.

$$s = (dd - d - m/60) \times 3600 \tag{4.4}$$

Po aplikaci těchto výpočtů získáme souřadnice referenčního bodu, ke kterému se bude vztahovat celá topologie. Po zpracování obdržené zprávy MAP využijeme tyto souřadnice na stažení podkladových map ze serverů společností Google a OpenStreetMap a jejich následné projekci.

4.4.1 Google Maps

Vytvoříme instanci třídy `GoogleTile`, rozšiřující generickou abstraktní třídu *AsyncTask*. Ta zajistí asynchronní spuštění kódu bez nutnosti ruční správy vláken. Třída přistupuje skrze rozhraní *Google Static Maps API v2* [3] ke statickým mapám uloženým na vzdálených serverech.

Rozhraní umožňuje vložení výstřižku z Google Maps přímo do aplikace, bez nutnosti využití JavaScriptu či jiných metod dynamického stahování. Služba vytváří mapu založenou na URL parametrech zaslaných jako HTTP požadavek a navrací zobrazitelný obrázek.

Parametry

Pro zisk správného obrázku je třeba korektně definovat povinné i volitelné parametry URL odkazu. Zvolené parametry:

- center: střed mapy definovaný ve formátu desetinných stupňů
- zoom: stupeň přiblížení
- size: rozlišení mapy
- scale: ovlivňuje počet navrácených pixelů
- maptype: typ mapy
- sensor: využití senzoru
- style: definuje vlastní styl
  - feature: rysy
  - element: podmnožina rysů
  - color: barva cest (32-bitová hexadecimální hodnota)
  - weight: šířka cesty
  - visibility: zobrazitelnost

Zvolená konfigurace

Konfigurace je dynamicky volena dle parametrů zařízení na kterém je aplikace spuštěna. Jako hlavní testovací zařízení byl zvolen tablet Nexus 7 (verze 2012). Pro něj bylo veškeré nastavení optimalizováno. V závorkách za textem jsou uvedeny použité hodnoty.

- center: pozice získaná z elementů *<lat></lat>* a *<lon></lon>* obsažených ve zprávě MAP

- zoom: za účelem co nejpřehlednějšího zobrazení bylo vybráno z rozmezí stupňů přiblížení 0 až 21+ (17)
- size: vybráno dle rozlišení displeje zařízení, jenž je děleno hodnotou *scale* (640 x 400)
- scale: zvolena hodnota, která násobí počet pixelů rozlišení (2)
- maptype: zvoleno satelitní zobrazení s popisem ulic a bodů hromadné dopravy (hybrid)
- sensor : aplikace nevyužívá vestavěný GPS senzor (false)
- style: aplikování prvního stylu
  - feature: aplikace na hlavní dopravní tepny (road.arterial)
  - element: aplikace jen na geometrické obrazce (geometry)
  - color: zvolena oranžová barva (0xff7043)
  - weight: zvolena šířka (2)
  - visibility: zobrazitelnost (true)
- style: aplikování druhého stylu
  - feature: aplikace na lokální silnice (road.local)
  - element: aplikace jen na geometrické obrazce (geometry)
  - color: zvolena zelená barva (0x00ff00)
  - weight: zvolena šířka (2)
  - visibility: zobrazitelnost (true)

Obrázek 4.2: Příklad zvolené konfigurace [snímek obrazovky]

Omezení

Služba je v době publikování této práce zdarma. Za předpokladu, že samotná aplikace nebude mít více než dvacet pět tisíc požadavků na stažení každý den po dobu více než devadesát dnů. Poté dojde ke kontaktování vývojáře aplikace. Toto omezení platí pouze v případě, že je aplikace podepsána *API (Application Programming Interface)* klíčem, čímž lze sledovat využití skrze portál Google Developer Console[3]. V opačném případě je horní limit 1000 stažených map z jedné IP adresy za 24 hodin a maximálně 50 požadavků na mapu za minutu. Délka URL je omezena na 2048 znaků.

4.4.2 Open Street Maps

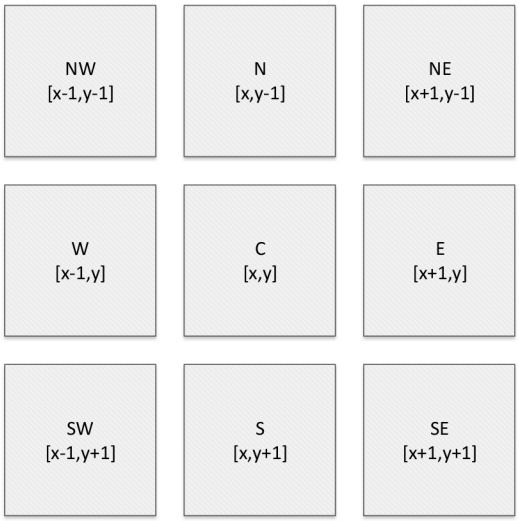
Vytvoříme instanci třídy `OSMTile`, rozšiřující generickou abstraktní třídu *AsyncTask*. Třída přistupuje skrze rozhraní *Open Street Map Editing API* [2] ke statickým mapám uloženým na vzdálených serverech.

Aplikace využívá mapových dlaždic, primárně určených pro webové rozhraní *Slippy-Map*[4] založené na *AJAX (Asynchronous JavaScript and XML)* knihovně *OpenLayers*.

Rozhraní umožňuje vložení výstřižku z Open Street Maps přímo do aplikace, bez nutnosti využití JavaScriptu či jiných metod dynamického stahování. Služba vyžaduje zadání validní URL ve formátu :

`http://tile.openstreetmap.org/zoom/x/y.png`

Jako výstup navrací odpovídající mapovou dlaždici s rozlišením 256 x 256 pixelů ve formátu *PNG (Portable Network Graphics)*.



Obrázek 4.3: Rozvržení bitmapové matice [vlastní]

Parametry

- zoom : celočíselná hodnota v rozmezí 0 až 18, kde 0 reprezentuje celý svět

[3] `https://console.developers.google.com/`

[4] strojově generuje mapy do rastrových obrázků, mezi kterými se dá posouvat gesty

- x : hodnota v rozmezí od 0 (levý roh je 180°W) do $2^{zoom} - 1$ (pravý roh je 180°E)
- y : hodnota v rozmezí od 0 (horní roh je 85.0511°N) do $2^{zoom} - 1$ (spodní roh je 85.0511°S) v Mercatorově zobrazení

Pro vygenerování map bylo třeba převést zeměpisné souřadnice do Marcatorova zobrazení. Převod je demonstrován rovnicemi 4.5 a 4.6.

$$x = \left\lfloor \frac{lon + 180}{360} \times 2^{zoom} \right\rfloor \tag{4.5}$$

$$y = \left\lfloor (1 - \frac{ln(tan(lat \times \frac{\pi}{180}) + \frac{1}{cos(lat \times \frac{\pi}{180})})}{\pi}) \times 2^{zoom-1} \right\rfloor \tag{4.6}$$

Jelikož jsou dlaždice generovány pouze v rozlišení 256 x 256, byl zvolen přístup, kdy je generována sada 9 dlaždic. Ta je poté programově skládána a dohromady tvoří mapový podklad. Každá z dlaždic je označena světovou stranou kterou reprezentuje vzhledem ke středové dlaždici. Složení je graficky reprezentováno na obrázku níže.

Obrázek 4.4: Grafické zobrazení dlaždic [vlastní]

4.5 Datová struktura

4.5.1 Data

Třída `Data` je kořenovou třídou, reprezentující celou topologii křižovatky. Instance této třídy je vytvořena při načtení zprávy typu MAP. V konstruktoru jsou inicializovány použité se-

znamy, čítačům jsou přiděleny nulové hodnoty, je vytvořena instance třídy `Header` a objekt *RStarTree*, který může držet až dvě stě uzlů. Třída deklaruje následující proměnné :

- lat : zeměpisná šířka v desetinných stupních
- lon : zeměpisná výška v desetinných stupních
- LatLon : seznam polí řetězců, držících informace o zeměpisných souřadnicích ve formátu DMS
- intersectionsCount : počet křižovatek
- tree : instance objektu, reprezentujícího R* strom
- header : hlavičkový objekt **Header** [4.5.2]
- interName : jméno křižovatky
- interId : identifikační číslo (dále jen id) křižovatky v hexadecimálním tvaru
- interIdDec : identifikační číslo (dále jen id) křižovatky v desítkovém tvaru
- laneWidth : šířka cesty
- laneSetCnt : počet cestovních množin (využito jen v Japonsku)
- crosswalkCnt : počet přechodů pro chodce
- arms : počet ramen křižovatky
- genericLaneList : seznam objektů `GenericLane` [4.5.3], držících informace o cestách
- googleTile : bitmapa zobrazující mapový podklad získaný ze serveru Google Maps
- osmC|N|S|W|E|NW|NE|SW|SETile : bitmapy zobrazující mapové podklady získané ze serveru Open Street Maps

4.5.2 Header

Třída **Header** uchovává informace získané z obdržené hlavičky zprávy typu MAP. V konstruktoru jsou proměnné inciovány na nulu. Třída deklaruje proměnné:

- protocolVersion : celočíselná verze využitého protokolu
- messageId : id zprávy
- stationId : id vysílací stanice

4.5.3 GenericLane

Třída `GenericLane` popisuje jednotlivé jízdní pruhy a přechody pro chodce. Uchovává informace o uzlech, kterými je pruh reprezentován, seznam spojení a nabídnuté manévry. Uzly definované v této třídě jsou svázány spojnicemi.

- laneNum : id jízdního pruhu či přechodu
- tree : instance objektu, reprezentujícího R* strom
- nodeList : seznam uzlů `Node` [4.5.6]
- connList : seznam spojení `Connection` [4.5.4]
- laneAttrs : objekt `LaneAttributes` [4.5.5]
- maneuvers : manévry

manévr	hodnota	popis
maneuverStraightAllowed	1	pohyb vpřed
maneuverLeftAllowed	2	odbočení doleva
maneuverRightAllowed	4	odbočení doprava
maneuverUTurnAllowed	8	obrat o 180 stupňů
maneuverLeftTurnOnRedAllowed	16	zastavení a odbočení doleva
maneuverRightTurnOnRedAllowed	32	zastavení a odbočení doprava
maneuverLaneChangeAllowed	64	změna pruhu
maneuverNoStoppingAllowed	128	zákaz zastavení
yieldAllwaysRequired	256	nutno dát přednost v jízdě
goWithHalt	512	zastavení, poté možné poračovat
caution	1024	nutná obezřetnost
reserved1	2048	přiřazení 12-bitového pole

Tabulka 4.1: Manévry

4.5.4 Connection

Třída `Connection` zahrnuje jízdní pruhy a přechody do signálních grup. Skrze tyto grupy se zajišťuje svázání se stavy semaforů poskytnutými ve zprávách SPAT.

- connLane : id jízdního pruhu či přechodu
- signalGroup : id signální skupiny

4.5.5 LaneAttributes

Třída **LaneAttributes** drží konstantní informace o jízdním pruhu či přechodu.

- parentId : id jízdního pruhu či přechodu
- directionalUse : určuje směr použití
- sharedWith : určuje zda je cesta sdílena (motorizovaná vozidla, kola, vlaky)
- laneType : typ cesty

typ	hodnota	popis	výchozí barva
vehicle	0	motorizovaná vozidla	černá
crosswalk	1	přechod pro chodce	modrá
bikeLane	2	cyklostezka	zelená
sidewalk	3	chodník	žlutá
median	4	kanalizace, škarpa	růžová
striping	5	dopravní značení	červená
trackedVehicle	6	vlaky a tramvaje	fialová
parking	7	parkoviště	šedá

Tabulka 4.2: Typy cest

4.5.6 Node

Třída `Node` uchovává data o konkrétních uzlech, jenž se budou vykreslovat na plátno. U každého z uzlů je zjišťováno, zda pro něj byla specifikována jako první souřadnice x či souřadnice y. Po vložení obou z nich je vytvořen nový objekt typu `java.awt.Point`, kterému jsou v konstruktoru předány souřadnice uzlu. Pro každý uzel je vytvořen ohraničující obdélník, jehož levý horní roh je shodný s pravým dolním rohem. Těmto rohům je přiřazen zmíněný objekt `java.awt.Point`, který je přidán do odkazovaného stromu.

- x : souřadnice x
- y : souřadnice y
- z : souřadnice z (nepovinná)
- tree : instance objektu, reprezentujícího R* strom
- parentId : id jízdního pruhu či přechodu
- inTree : označuje zda je uzel obsažen v objektu `tree`
- entity : řetězec uchovávající informace o vztahu uzlu k nadřazeným objektům

4.5.7 MovementEvent

Třída `MovementEvent` nese informace o dynamicky se měnících stavech křižovatky. Ty jsou přijímány ve zprávách SPAT. Třída deklaruje proměnné:

- signalGroup : id signální skupiny
- spatType : světelný stav semaforu
- minEndTime : minimální čas zbývající do přepnutí stavu

4.6 Ukládání dat

Jednotlivé uzly nesoucí informace o souřadnicích bylo třeba ukládat. Ukládání do datové struktury se však neukázalo jako nejlepší nápad, jelikož bylo příliš časově náročné a neumožňovalo provádění hromadných operaci nad skupinami bodů. Za účelem správy prostorových dat nejvíce vyhovuje prostorová databáze, která je popsaná v sekci 2.6. Jako konkrétní přístup byl zvolen R* strom.

stav	popis
STOP_THEN_PROCEED	blikající červená
STOP_AND_REMAIN	červená
PRE_MOVEMENT	příprava na zelenou
PERMISSIVE_MOVEMENT_ALLOWED	zelená
PROTECTED_MOVEMENT_ALLOWED	zelená s výstrahou
PERMISSIVE_CLEARANCE	žlutá
PROTECTED_CLEARANCE	žlutá s výstrahou
CAUTION_CONFLICTING_TRAFFIC	blikající žlutá
NONE	nedefinováno

Tabulka 4.3: Světelné stavy semaforů

SQLite

SQLite je přenosná, kompaktní, jednoduchá, efektivní, spolehlivá, relační databáze, která je vedena jako open source[4]. Jedná se o embedded databázi[5], jenž využívá operační systém Android.

SQLite standardně podporuje modul R* strom. Jeho implementace umožňuje indexaci až pěti dimenzí dat. Tabulky obsahují sloupec s celočíselnými primárními klíči následovanými jedním až pěti páry sloupců. To vede k tabulkám s lichým počtem sloupců (3-11). Data musí být vkládána v párech. Při vkládání bodů se pro minimální i maximální komponentu vkládají stejné hodnoty.

Zdrojový kód pro modul R* strom je v knihovně SQLite zahrnut, avšak není ve výchozím nastavení aktivován. Pro zapnutí je třeba databázi zkompilovat s definovaným makrem *SQLITE_ENABLE_RTREE*. Pro práci s takto zkompilovanou knihovnou je v systému Android třeba využít pracovních nástrojů *Android NDK (Native Development Kit)*. Tímto způsobem však poté dostáváme do systému dvě databáze SQLite.

Pro správu prostorových dat byla se svolením autora využita externí knihovna *cz.muni.fi.bc.rstartree*.

4.7 Vizualizace dat

Veškerá vizualizace je prováděna v pohledu `VisualizerView`, rozšiřujícím pohled *SurfaceView*. Třída implementuje rozhraní *Runnable*. Po spuštění, voláním metody *Thread.start()*, je vytvořena uzavřená smyčka, ve které běží veškerá vizualizace. K samotnému pohledu se přistupuje skrze instanci objektu `SurfaceHolder`.

SurfaceView nabízí kreslící paletu vloženou do hierarchie pohledů, které jsou umisťovány na ose Z, tj. poslední přidaný pohled překrývá pohledy umístěné před ním. Z tohoto důvodů jsou jednotlivé komponenty vykreslovány systematicky za sebou. K samotnému kreslení je využito plátno *Canvas*, které je v každém smyčky překreslováno. V rámci této sekce budou popsány jednotlivé kroky vizualizace.

Pozadí

Po uzamknutí plátna je vykresleno pozadí aplikace. Jako barva byla zvolena neutrální bílá. Na toto pozadí je z mapových dat, získaných ze vzdálených serverů, sestrojena bitmapa.

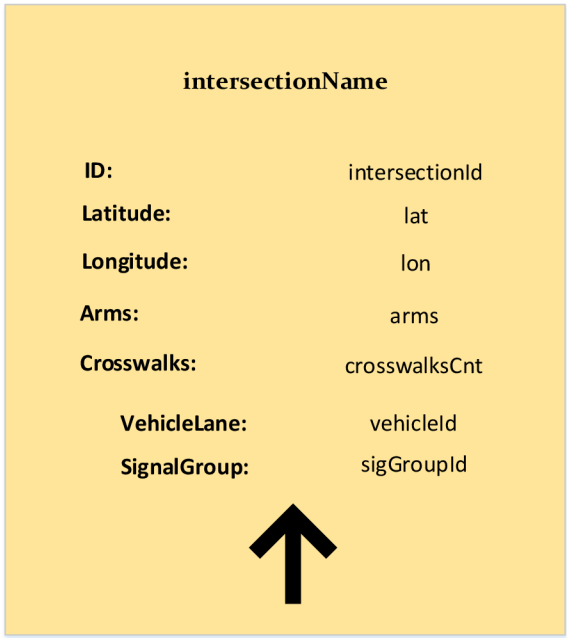
[5]databáze běžící v adresovém prostoru aplikace, která ji využívá

Ta se vykreslí na pozadí a zobrazí uživatelovi digitální pohled na místo, kterým projíždí. V případě zvolené Open Street Map se vykreslí sada devíti bitmap zasazených do matice 3x3.

Informační okno

Pro zobrazení informací o aktuální křižovatce je vykresleno do pravého horního rohu informační okno. Jeho rozměry jsou vypočítány na základě aktuálního rozlišení displeje. Okno zobrazuje následující informace, které jsou přístupné skrze objekt `Data`:

- intersectionName: jméno křižovatky
- id: identifikační číslo křižovatky
- latitude: zeměpisná šířka ve formátu DMS
- longitude: zeměpisná výška ve formátu DMS
- arms: počet ramen křižovatky
- crosswalkCounter: počet přechodů pro chodce
- actualLaneId: identifikační číslo aktuálně vybraného pruhu či přechodu
- actualSignalGroupId: identifikační číslo skupiny semaforů, pod kterou spadá aktuální pruh či přechod



Obrázek 4.5: Informační okno [vlastní]

Dopravní topologie

Pro vizualizaci topologie křižovatky je volána metoda *drawConnectionTopology()*. Z kořenového uzlu odkazovaného stromu jsou voláním metod *getTopLeft()* a *getBottomRight()* získány krajní body, jenž utváří obdélníkovou oblast, ve které se všechny ostatní body nachází. Nad kořenovým uzlem se provádí prohledávání všech uzlů celého stromu, které leží v dané oblasti. Pole takto nalezených uzlů, reprezentovaných obdélníky, je postupně procházeno. Každý z bodů je explicitně prohledáván na výskyt entity, která má podobu řetězce rozděleného symboly "∼∼∼". Takto strukturovaná entita označuje body, jež jsou svázány spojnicemi.

Uzly

Pozicování uzlů prošlo během vývoje zásadními změnami. V první implementované verzi zprávy MAP byly pozice všech uzlů vztaženy ke středovému bodu, na který bylo odkazováno zeměpisnými souřadnicemi. Od toho bylo v dalších verzích zpráv upuštěno a byl zaveden nový model. V rámci něj je pouze první uzel cesty vztažen ke středovému bodu. Ostatní se již vztahují vždy k bodu předcházejícímu. Pro vykreslení uzlu je v každém cyklu vykreslovacího vlákna volána metoda *drawCircle(float cx, float cy, float radius, Paint paint)* s následujícími parametry:

```
canvas.drawCircle(
   DEVIATION_X + // definovaná x-ová odchylka
   this.getWidth() / 2 + // středový bod displeje na x-ové souřadnici
   (int) f.getTopLeft().getX() / ZOOM_LEVEL, // x-ová souřadnice bodu upravená
       dle úrovně přiblížení

   -DEVIATION_Y + // definovaná y-ová odchylka
   this.getHeight() / 2 - // středový bod displeje na y-ové souřadnici
   (int) f.getTopLeft().getY() / ZOOM_LEVEL, // y-ová souřadnice bodu upravená
       dle úrovně přiblížení

   CIRCLE_RADIUS, // průměr vizualizovaného uzlu
   paint // kreslící nástroj s˜přednastavenou barvou
   );
```

Zdrojový kód 4.3: Vykreslení uzlu

Spojnice

Každý uzel uchovává proměnnou `entity`, jenž definuje index nadřazené třídy `GenericLane` 4.5.3. Uzly jsou rozděleny do skupin dle indexu entity. Tyto skupiny jsou cyklicky procházeny. Mezi sousedními uzly jsou vykreslovány spojnice. Dle hodnoty proměnné `laneType`, třídy `LaneAttributes` 4.5.5, je kreslícímu nástroji nastavena odpovídající barva. Pro vykreslení spojnice je volána metoda *drawCircle(float cx, float cy, float radius, Paint paint)* s následujícími parametry:

```
canvas.drawLine(
/*X1*/(DEVIATION_X + // definovaná x-ová odchylka
      getWidth() / 2 + // středový bod displeje na x-ové souřadnici
      (int) Integer.valueOf(nodeList.get(vehPos).x) / ZOOM_LEVEL),
      // x-ová souřadnice aktuálně procházeného uzlu

/*Y1*/(-DEVIATION_Y + // definovaná y-ová odchylka
      getHeight() / 2 - // středový bod displeje na y-ové souřadnici
      (int) Integer.valueOf(nodeList.get(vehPos).y) / ZOOM_LEVEL),
      // y-ová souřadnice aktuálně procházeného uzlu

/*X2*/(DEVIATION_X +
      getWidth() / 2 +
      (int) Integer.valueOf(nodeList.get(vehPos - 1).x) / ZOOM_LEVEL),
      // x-ová souřadnice předešlého uzlu

/*Y2*/(-DEVIATION_Y +
      getHeight() / 2 -
      (int) Integer.valueOf(nodeList.get(vehPos - 1).y) / ZOOM_LEVEL),
      // y-ová souřadnice předešlého uzlu

      paint // kreslící nástroj s˜přednastavenou barvou
      );
```

Zdrojový kód 4.4: Vykreslení spojnice mezi uzly

4.8 Zobrazení dynamických stavů

Při počáteční inicializaci aplikace je vytvořen statický pohled `gui_visualizer`, který definuje pole množin semaforů umístěných od levého horního rohu obrazovky po informační okno. Každá množina obsahuje devět semaforů, reprezentujících jednotlivé světelné stavy. Jednotlivé semafory jsou složeny z bitmap signalizačních světel a manévrovacích ukazatelů. Tyto semafory jsou na sebe vrstveny, přičemž zobrazen je objekt, který je v dané hierarchii nejvýše.

Po výběru některého z uzlů je vykreslovací vlákno o této akci informováno. Informace o vybraném uzlu jsou předány informačnímu oknu, jenž zobrazí atributy uzlu a jeho totožnost. Vykreslovací vlákno prochází poslední načtený seznam stavů poskytnutý třídou `MovementEvent`. Při shodě s identifikačním číslem signální skupiny vyvolá vlákno v každém cyklu metodu na vykreslení semaforů dle následujícího postupu:

1. skrytí všech předešle zobrazených semaforů
2. zjištění počtu signálních skupin, do kterých spadá uzel
3. zobrazení množin semaforů ve výchozím stavu, které odpovídají počtu signálních skupin
4. zjištění, zda jde o množinu semaforů určenou pro řidiče či pro chodce
5. iterace nad každou zobrazenou množinou semaforů
   (a) zobrazení semaforu s odpovídajícím světelným stavem

(b) nastavení manévrovacího ukazatele

(c) skrytí ostatních semaforů

(d) zjištění, zdali je k dispozici informace čase zbývajícím ke přepnutí semaforu

   i. výpočet souřadnic pro umístění kruhu s časovačem

   ii. vykreslení informativního kruhu s barvou odpovídající aktuálnímu světelnému stavu

   iii. vykreslení hodnoty časovače

Po přijetí nové SPAT zprávy a jejím vložení do datové struktury aplikace jsou v následujícím cyklu vykreslovacího vlákna vizualizovány nově získaná data.

4.9 Uživatelská nastavení

Při dlouhém kliknutí na úvodní obrazovku aplikace dojde ke spuštění aktivity `UserSettings`. Ta umožňuje uživatelům změnu základních nastavení ovlivňujících vizualizaci. Mezi nabízené možnosti patří:

- Node size: změna velikosti uzlu
- Node color: změna barvy uzlu typu *VEHICLE*
- Crosswalk color: změna barvy uzlu typu *CROSSWALK*
- Touch tolerance: změna velikost plochy okolo uzlu, ve které je uzel po dotyku vybrán
- Numeral system: výběr mezi hexadecimálním a desítkovém zobrazení identifikačních čísel v informačním okně

4.9.1 Ladění

Pro zobrazení ladících informací uživatel musí v systémovém nastavení operačního sytému Android pod položkou **Vývojářské menu** povolit možnost **Ladění USB**. Vývojářské menu je z bezpečnostních důvodů ve výchozím nastavení pro běžného uživatele skryté. Jeho vyvolání se liší dle výrobce a verze systému. Přístup do menu bývá popisován na oficiálních stránkách výrobce. Pro zobrazení kontrolních výpisů je třeba mít zařízení připojené k počítači, na kterém běží výše zmiňovaná aplikace **Logcat**. V rámci této aplikace jsou zaznamenávány veškeré kontrolní výpisy všech běžících procesů. Pro zobrazení námi požadovaných dat je třeba vytvořit filtr, kterému se nastaví filtrování dle jména balíku aplikace, *com.siemens.c2xdemo*. Poté jsou zobrazována pouze relevantní data.

Obrázek 4.6: Aplikace s podkladovou vrstvou *GM (Google Maps)* [snímek obrazovky]

Obrázek 4.7: Aplikace s podkladovou vrstvou *OSM (Open Street Maps)* [snímek obrazovky]

Kapitola 5

Testování

Tato kapitola popisuje testování prováděné na aplikaci. Testování aplikace se po dobu jejího vývoje měnilo, čemuž bylo potřeba přizpůsobovat testovací komponenty. Dle daných fází je kapitola rozdělena do příslušných sekcí.

5.1 Zpracování zpráv

První částí vývoje, kterou bylo třeba průběžně testovat, bylo zpracování zpráv ve formátu XML. V této části byl implementován modul, ve kterém byl vytvořen SAX parser XML zpráv. Pro jeho správný chod bylo třeba nadefinovat veškeré elementy obsažené ve zprávách MAP a SPAT. V první fázi byl společností Siemens poskytnut výňatek z dokumentu [1]. Ten obsahuje příklad zprávy MAP, jež se měla zpracovat. Oproti současné verzi zde nebyl ani jeden shodný element. Taktéž přístup zpracování uzlů byl zcela odlišný. Hlavním rozdílem bylo jejich umisťování. Pozice všech uzlů byly vztaženy ke středovému bodu, na rozdíl od současného přístupu, ve kterém je pozice každého bodu vztažena k bodu předešlému. Toto byla na dlouhou dobu jediná specifikace, ze které se vycházelo.

Po několika měsících byly společností Siemens uvolněny tři testovací zprávy MAP. Ty se však již od předešlé specifikace výrazně lišily. Datová struktura proto musela býti navrhnuta znovu od počátku. Z důvodu zastaralosti předešlé verze zpráv, kterou již jednotky nepodporovaly, nebylo nutné zajišťovat zpětnou kompatibilitu.

Tři testovací zprávy MAP se tak staly na dlouho dobu jediným možným vstupem. Na doporučení kolegů ze společnosti Siemens nebyly uměle vytvářeny komplexnější zprávy, jelikož byly očekávány větší změny v datové struktuře, z důvodu blížícího se veletrhu inteligentních dopravních systému, na kterém byly představeny novinky.

Do implementovaného modulu byla přidána možnost ručního výběru XML souboru s danou zprávou. Výběr se vyvolá kliknutím na logo Siemens na úvodní obrazovce. Pro snazší testování a hledání chyb byl tento přístup ponechán i do dalších verzí.

5.2 Komunikace s jednotkou

Po ustálení formátu přijímaných zpráv byla potřeba zprovoznit komunikaci mezi mobilním zařízením a jednotkou OBU, jenž je fyzicky přítomna v prostorách společnosti Siemens v Brně. Jednotka OBU vytváří přístupový bod Wi-Fi, ke kterému se může připojit kdokoliv v jejím dosahu. Po zaslání registrační zprávy začíná jednotka zasílat zprávy MAP a SPAT.

5.2.1 Analýza paketů

Kontrola formátu zasílaných zpráv probíhala skrze počítač připojený k přístupovému bodu, poskytnutým jednotkou. Stroj, s nainstalovaným operačním systémem Ubuntu, využíval za tímto účelem program Wireshark. Wireshark je protokolový analyzér a paketový sniffer, sloužící k analýze a ladění programu v počítačových sítích. Pro zajištění zobrazení požadovaných dat byl vytvořen skript *c2x.lua*, filtrující přijaté pakety. Skript byl napsán v jazyce *Lua*[1] a je možné jej importovat do všech verzí programu Wireshark s podporou tohoto jazyka.

5.2.2 Simulace jednotky

Pro testování dynamického příjmu dat, mezivláknové komunikace a nahodilých výpadku sítě bylo potřeba být fyzicky přítomný v okolí testovací jednotky. To se však s přibývajícími specifikacemi ukázalo jako velmi časově náročné a neefektivní. Řešením této komplikace byla implementace mobilní aplikace *SPATSender*, simulující jednotku OBU. Simulátor je popsán v kapitole 5.3.

5.3 Simulátor

Simulátor jednotky OBU byl vyvinut jako samostatná mobilní aplikace určená pro operační systém Android. Jejím primárním cílem bylo simulovat chování jednotky OBU v reálném čase. Hlavní nároky byly kladeny na využitelnost a jednoduchost. Bylo proto vytvořeno pouze prosté grafické prostředí s možností interakce s uživatelem. Uživatel aplikace je zároveň jejím vývojářem. Díky tomuto faktu je množina odesílaných zpráv měnitelná pouze úpravou programových konstant. Po každé editaci je třeba program zkompilovat a opětovně spustit. Jelikož simulátor slouží ke sledování dlouhodobých procesů a nahodilých výpadků, je tento fakt přípustný.

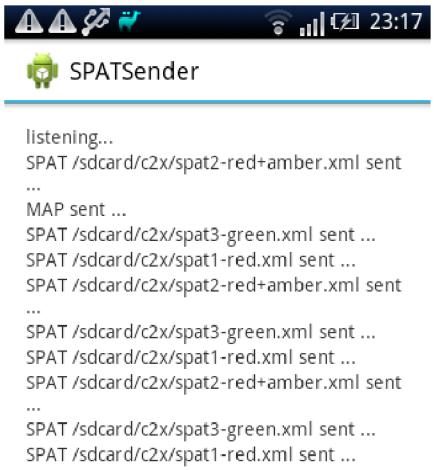
Po spuštění aplikace je vytvořeno dialogové okno vyzývající uživatele zadat základní parametry nutné k ustanovení spojení. Aplikace neumožní spojení bez vložení všech požadovaných hodnot v korektním formátu. Požadované údaje:

- IP adresa mobilního zařízení, na kterém běží aplikace C2X
- port aktivního zařízení, na kterém se sestrojuje soket
- interval odesílání zpráv MAP (v sekundách)
- interval odesílání zpráv SPAT (v sekundách)

Pro zajištění korektního běhu je třeba, aby byly obě aplikace spuštěny ve stejném čase na zařízeních připojených v jedné síti. Po zadání všech údajů dojde k navázání spojení. V nastavených časových intervalech aplikace *SPATSender* zasílá zprávy MAP a SPAT aplikaci na vzdáleném zařízení. Pro reálnou demonstraci komunikace jsou zprávy k odeslání vybírány z předdefinované množiny, nad kterou je iterováno. Odeslání každé zprávy je demonstrováno informativním výpisem.

Simulátor oproti klasické jednotce nepodporuje ustanovení komunikace zasláním registrační zprávy. Ta však není prakticky nutná, jelikož zpráva, potvrzující registraci, není povinná.

[1] imperativní, procedurální programovací jazyk



Obrázek 5.1: Simulátor [snímek obrazovky]

5.4 Správa verzí

Veškerý pokrok při vývoji aplikace byl zaznamenáván a ukládán. Pro zpřehlednění vývoje, zjednodušení přístupu a zálohu dat byla aplikace nasazena do systému *SVN (Subversion)*. Systém umožňuje správu a verzování zdrojových kódů. Taktéž bylo díky systému možné zdrojové kódy sdílet s pracovníky společnosti Siemens, kteří se podílejí na tamějším vývoji systému C2X. Na základě jejich požadavků byly rovněž jednotlivé prototypy aplikace verzovány. V příloze C jsou vypsány informace k vybraným verzím.

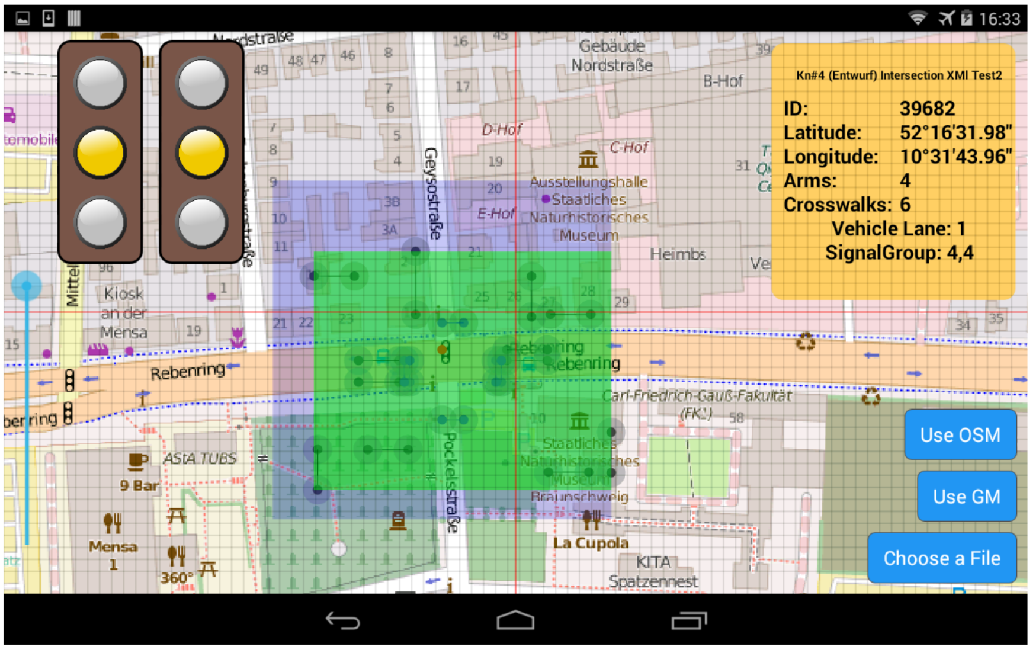
5.5 Ladící mód

Za testovacími účely byl vyvinut ladící mód aplikace, jenž graficky zobrazuje střed aktivní části displeje, informace o stavu připojení a hraniční obdélníky. Ty umožňují jednoduše rozpoznat MAP zprávy se špatně zadefinovanými souřadnicemi uzlů, které jsou umístěny mimo zobrazitelnou oblast displeje. Příkladem může být mapa *Kn#4 (Entwurf)*, jejíž uzly jsou náhodně rozmístěny po mapě. Navíc i přidělení uzlů do signálních skupin není již na první pohled validní. Tyto informace mohou pomoci lidem, kteří testují nově nasazené jednotky. Příklad topologie, která není validní je demonstrován obrázkem 5.2.

V ladícím módu byly při zobrazování křižovatky navíc poskytnuty tyto přídavné grafické informace

- linie dělící displej
- hraniční obdélníky MBR
- plochy ohraničující okolí uzlů, kam lze kliknout
- bod označující střed

- verze aplikace
- dodatečné textové informace a výpisy



Obrázek 5.2: Příklad invalidní topologie [snímek obrazovky]

5.6 Výsledky testování

V poslední fázi vývoje byla aplikace podrobena závěrečné zkoušce. Testování probíhalo v kancelářských prostorech společnosti Siemens, kde byly fyzicky přítomné jednotky OBU a RSU. Jednotka RSU měla uloženou topologii německé křižovatky 08060, umístěné ve městě Münster. Ta je zapojena do systému C2X a v době závěrečného testování vysílala data o její topologii a světelných stavech do svého okolí. Od kolegů z německého oddělení společnosti Siemens byla tato data získána a nahrána do jednotky RSU umístěné v Brně. Zprávy vytvořené z těchto dat byly kódovány dle formátu UPER a dále přeposílány jednotce OBU.

Jednotka RSU byla ovládána kontrolérem, který umožňoval měnit signální plány křižovatky. Ty určují v jakém pořadí a po jakou dobu budou svítit semafory umístěné nad křižovatkou. Signální plány se mění v závislosti na čase, průjezdnosti a vnějších aspektech. Tímto je možné nakonfigurovat odlišné chování ve špičce a uprostřed noci. Výměnu plánu lze ručně vynutit kontrolérem připojeným k RSU. Změna je aplikována do devadesáti sekund. Mezitím časovače zobrazují výchozí hodnotu 255 a stav lamp se nemění.

Mobilní aplikace se po přiblížení a připojení k bezdrátovému bodu jednotky OBU automaticky zaregistrovala a začala přijímat zprávy zakódované jako XER. Informace získané z přijatých zpráv ihned vizualizovala. Tato data byla v reálném čase porovnávána s výstupem programu Wireshark, jenž běžel na počítači připojeném do stejné sítě. Změny světelných stavů byly porovnávány s výstupem softwaru Sitraffic. Data získaná z výstupů softwarů odpovídala grafické vizualizaci mobilní aplikace.

Kapitola 6

Závěr

V práci je analyzována, navržena a implementována mobilní aplikace pro vizualizaci dopravních dat ze systému C2X. Během analýzy byl prostudován dopravní systému C2X. Dále pak architektura systému s jeho komponentami a komunikační protokoly. Pro výměnu dat byly vybrány zprávy MAP a SPAT uchovávající informace o topologii a světelné signalizaci.

Aplikace implementuje komunikaci s jednotkami prostřednictvím XML zpráv. Přijaté zprávy analyzuje, parsuje a ukládá do datových struktur. Odesílané zprávy vytváří či načítá z úložiště zařízení a zasílá vzdálené jednotce umístěné ve stejné bezdrátové síti. Ke komunikaci využívá protokol UDP. Vizualizace podkladové vrstvy je v aplikaci implementována prostřednictvím map získaných ze serverů Google Maps a Open Street Maps. Mapové dlaždice jsou ze serverů získávány na základě specifických dotazů, které jsou vytvořeny dle pozic křižovatek, získaných analýzou přijatých zpráv.

Aplikace byla po konzultacích ve společnosti Siemens navržena s ohledem na jednoduchost ovládání, stabilitu a plynulou funkčnost. Tím je zajištěna rychlá a snadná odezva pro řidiče. Hlavními úspěchy jsou vizualizace topologie a dynamicky se měnících dat. Implementováno je zobrazení a grafické odlišení vstupujících a vystupujících jízdních pruhů křižovatky, přechodů pro chodce, stezek pro cyklisty a dalších dopravních komponent spolu s informacemi o křižovatce. Při výběru některého ze vstupních pruhů jsou zobrazeny semafory pro každý z odpovídajících manévrů. Světelné stavy se na semaforech mění dle aktuální situace. Ta je v krátkých časových intervalech přijímána zprávami SPAT.

Aplikace byla během čtyřech semestrů vývoje testována na jednotkách RSU a OBU umístěných v prostorách společnosti Siemens. Veškeré testy probíhaly na reálných datech získaných z testovacích okruhů ve vybraných evropských městech, kde je již systém C2X zaveden. V poslední fázi proběhlo testování aktuální konfigurace aplikované na okruhu v německém městě Münster. Testy dokázaly správnou funkčnost aplikace při zpracování dat z reálného prostředí. Je možné je shlédnout v přiloženém videu.

Z hlediska vývoje by bylo možné rozšířit aplikaci o příjem dalších typů zpráv systému C2X. Nejpřínosnější jsou zprávy CAM, prostřednictvím kterých je možné získávat detailní informace o okolních vozidlech, a kterými lze mezi těmito vozidly komunikovat. Využití taktéž přináší zprávy SAM a DENM, které vysílají informace o službách jednotek a varují řidiče před nebezpečím na cestách. Tato témata však již nebyla předmětem vývoje a jejich vývoj se odvíjí od diskuze se společností Siemens.

Literatura

[1] Intersection VLSA 02094. 2013.

[2] API - Open Street Map Wiki. [cit. 2014-12-20].
URL `http://wiki.openstreetmap.org/wiki/API`

[3] Static Maps API V2 Developer Guide. [cit. 2015-03-02].
URL `https://developers.google.com/maps/documentation/staticmaps/`

[4] Allen, G.; Owens, M.: *The Definitive Guide to SQLite*. Berkely, CA, USA: Apress, druhé vydání, 2010, ISBN 14-302-3225-0.

[5] Arlow, J.; Neustadt, I.: *UML 2 a unifikovaný proces vývoje aplikací*. Brno: Computer Press, vyd. 1. vydání, 2007, ISBN 978-80-251-1503-9.

[6] Beckmann, N.; Kriegel, H.-P.; Schneider, R.; aj.: The R*-tree: An Efficient and Robust Access Method for Points and Rectangles. In *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, SIGMOD '90, ACM, 1990, ISBN 0-89791-365-5, s. 322–331, doi:10.1145/93597.98741.
URL `http://doi.acm.org/10.1145/93597.98741`

[7] Brinkhoff, T.; Kriegel, H.-P.; Seeger, B.: Efficient Processing of Spatial Joins Using R-trees. *SIGMOD Rec.*, ročník 22, č. 2, Červen 1993: s. 237–246, ISSN 0163-5808, doi:10.1145/170036.170075.
URL `http://doi.acm.org/10.1145/170036.170075`

[8] Güting, R. H.: An introduction to spatial database systems. *The VLDB Journal*, ročník vol. 3, č. issue 4, 1994: s. 67–100, ISSN 1066-8888.
URL `http://link.springer.com/10.1007/BF01231602`

[9] by Hagen Stübing: *Multilayered Security and Privacy Protection in Car-to-X Networks Solutions from Application down to Physical Layer*. Wiesbaden: Springer Fachmedien Wiesbaden, 2013, ISBN 978-3-658-02530-4.

[10] Hartenstein, H.; Laberteaux, K.: *VANET*. Chichester, U.K.: Wiley, 2010, ISBN 04-707-4062-0.

[11] ISO: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications. Technická Zpráva 17572-3:2014, International Organization for Standardization, Geneva, Switzerland, 06/2009.

[12] ISO: Dedicated Short-Range Communication - Application layer. Standard 15628:2013, International Organization for Standardization, Geneva, Switzerland, 11/2000.

[13] Kosch, T.: *Automotive internetworking*. Hoboken, N.J.: Wiley, 2012, ISBN 978-047-0749-791.

[14] Manolopoulos, Y.; Nanopoulos, A.; Papadopoulos, A. N.; aj.: *R-Trees: Theory and Applications (Advanced Information and Knowledge Processing)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005, ISBN 1852339772.

[15] Mawston, N.: Android Captures 84% Share of Global Smartphone Shipments in Q3 2014. [cit. 2014-10-31].
URL `http://9to5mac.com/2014/10/31/android-vs-ios-market-share-3q-2014/`

[16] Mlýnková, I.; Pokorný, J.: *XML technologie*. Praha: Grada, první vydání, 2008, ISBN 978-80-247-2725-7.

[17] WWW stránky: Drive C2X technology [online]. [cit. 2013-07-12].
URL `http://www.drive-c2x.eu/technology`

Příloha A

Obsah CD

- Technická zpáva ve formátu pdf - /thesis-pdf/
- Zdrojové kódy technické zprávy v LaTeX - /thesis-tex/
- Zdrojové kódy aplikace - /src-c2x/
- Zdrojové kódy simulátoru - /src-sim/
- Skript pro wireshark - /src-wireshark/
- Zápisy z mítinků - /meeting-minutes/
- Manuál - /manual/

Příloha B

Zkratky

ITS	Intelligent Transportation System
API	Application Programming Interface
C2X	Car-To-X
C2C	Car-To-Car
C2I	Car-To-Infrastructure
ITS	Intelligent Transport Systems
GIS	Geographic Information System
IDE	Integrated Development Environment
ETSI	European Telecommunications Standards Institute
IT	Information Technology
AU	Application Unit
HMI	Human-Machine Interface
ITU	International Telecommunication Union
USB	Universal Serial Bus
AoA	Angle of Arrival
ToA	Time of Arrival
OBU	On Board Unit
RSU	Road Side Unit
CCH	Control Channel
SCH	Service Channel
ADT	Android Development Tools

DSRC	Dedicated Short-Range Communications
DENM	Distributed Environmental Notification Message
PreDriveC2X	Preparation for Driving Implementation and Evaluation of C2X
MAP	Map Data
SPAT	Signal Phase And Timing Message
BAC	Blood alcohol content
WLAN	Wireless Local Area Network
CAN	Controller Area Network
CAM	Cooperative Awareness Message
SAM	Service Announcement Message
PET	Privacy Enhancing Technologies
OFDM	Orthogonal Frequency Division Multiplexing
QAM	Quadrature Amplitude Modulation
MAC	Media Access Control
QoS	Quality of Service
BSS	Basic Service Set
BSSID	Basic Service Set Identification
UML	Unified Model Language
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
XML	Extensible Markup Language
ASN.1	(Abstract Syntax Notation One)
BER	Basic Encoding Rules
XER	XML Encoding Rules
PER	Packed Encoding Rules
UPER	Unaligned Packed Encoding Rules
TLV	Type Length Value
OSM	Open Street Map
SAX	Simple API for XML

DOM	Document Object Model
DMS	Degrees-minutes-seconds
DD	Decimal Degrees
GPS	Global Positioning System
PNG	Portable Network Graphics
MBR	Minimum Bounding Rectangles
NDK	Native Development Kit
SVN	Subversion

Příloha C

Správa verzí

```
Po, led 20 2014 17:25:47
Změny:

    zvětšení vykreslené topologie
    přídáno zobrazení ID křižovatky
    opravení nekorektního zobrazení GPS souřadnic
    zvětšen počáteční uzel jizdního pruhu
       (na klikatelnou velikost - testováno na Nexusu 7)
    po kliknutí na počáteční uzel dojde k vypsání jeho ID
```

```
Út, led 21 2014 12:49:41
Změny:

    upravena topogie, nyní už sedí osy x a y
    konečná úprava zobrazených GPS souřadnic z DD formátu do DTS
       (teď sou již dohledatelné na mapách)
    grafické a textové vyznačení přechodů
```

```
Po, dub 07 2014 14:47:00
Změny:

        aplikace převedena do nového formátu XML (Kn667.xml a Kn8060.xml)
        vyřešen problém se zamrzáním (změnou hodnot přímo v canvasu)
        zavedeno grafické rozlišení uzlů na základě Signal Group
           (společných skupin semfaorů)
        objekty rozděleny do samostatných tříd
        rozdělení infrastruktury do balíčku pro zlepšení přehlednosti
        přidán vertikální SeekBar pro podporu zoomování
        přidána možnost výběru XML souboru skrze dialog přímo v aplikaci
```

```
Čt, dub 17 2014 01:07:00
Změny:

       přidána možnost přepínat mezi podkladovými vrstvami
          (GoogleMaps a OSM)
       úprava zoomů
       přidáno zoomování OSM vrstvy
       úprava aplikace pro podporu novější verze XML
          (které nevyžaduje položku LaneNumber)
       OSM vrstva složená z 9 bitmap (C,N,S,W,E,NE,NW,SE,SW)
       nevyrovnaný zooming
       nezobrazování korektního počtu aprroaches
       špatné zobrazení mapové vrstvy pro Kn_8060_3
          (chyba souřadnic)
```

```
Ne, kvě 04 2014 23:57:00
Změny:

       file chooser after start added
       preferences enabled
       definition of node size, node color,
          crosswalk color and touch tolerance
       default google map type changed to satellite
       green info box changed color to yellow
          (better contrast on satellite google maps)
       version info added on start sceen
```