



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

SIMULACE DOPRAVY ZA VYUŽITÍ DOPRAVNÍCH ŘADIČŮ

TRAFFIC SIMULATION USING TRAFFIC CONTROLLERS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. DAVID DRESSLER

VEDOUcí PRÁCE

SUPERVISOR

doc. RNDr. JITKA KRESLÍKOVÁ, CSc.

BRNO 2019

Zadání diplomové práce



19611

Student: **Dressler David, Bc.**
Program: Informační technologie Obor: Informační systémy
Název: **Simulace dopravy za využití dopravních řadičů**
Traffic Simulation Using Traffic Controllers
Kategorie: Softwarové inženýrství

Zadání:

1. Seznamte se s konfigurací dopravních řadičů společnosti Siemens, s.r.o.
2. Seznamte se se stávajícím simulačním systémem, konfigurací dopravních řadičů, doplňující konfigurací programu pro návrh křižovatek a s principy dynamického řízení křižovatek.
3. Po konzultaci s konzultantem zadávající firmy navrhnete rozšíření stávající aplikace tak, aby využívala logiku řízení křižovatky reálných dopravních řadičů.
4. Navržené a schválené vybrané řešení implementujte.
5. Použitelnost vytvořeného systému otestujte v simulovaném prostředí a demonstруйте na vhodně zvoleném vzorku dat vybraném po dohodě s vedoucí.
6. Zhodnoťte dosažené výsledky a diskutujte další možnosti rozšíření vytvořené aplikace.

Literatura:

- TP 81. *Navrhování světelných signalizačních zařízení pro řízení provozu na pozemních komunikacích*. Praha: Ministerstvo dopravy č. j. 122/2015-120-TN/2, 2015. (<http://www.pjpk.cz/viewFile.asp?file=1542>).
- BARCELÓ, J.: *Fundamentals of Traffic Simulation*, Springer Science & Business Media, 2011, ISBN: 9781441961426.
- Simulátor dopravních infrastruktur a situací. (<https://www.fit.vutbr.cz/study/DP/DP.php.cs?id=21216>)

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 3.

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Kreslíková Jitka, doc. RNDr., CSc.**

Konzultant: Verner Jan, Ing., Siemens CZ

Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2018

Datum odevzdání: 22. května 2019

Datum schválení: 30. října 2018

Abstrakt

Cílem této diplomové práce je implementovat návrh rozšíření stávajícího simulačního systému pro návrh křižovatek. Požadované rozšíření umožní využití logiky řízení dopravních řadičů sX společnosti Siemens, s.r.o. První část práce se věnuje problematice řízení křižovatek, konfiguracím dopravních řadičů a principům dynamického řízení křižovatek. Další kapitola je věnována popisu implementace stávajícího simulačního systému. Následující kapitola je věnována způsobu využití dopravních řadičů sX a také samotnému návrhu rozšíření stávajícího simulačního systému tak, aby mohl využívat logiku řízení těchto dopravních řadičů. Poté následuje kapitola popisující implementaci tohoto rozšíření. Poslední kapitola je věnována testování celého systému z hlediska funkčnosti a výkonu. Závěrem jsou také nastíněny další možnosti pro vývoj rozšířené aplikace.

Abstract

The aim of this master's thesis is to design an extension of the existing simulation system for designing traffic intersections. The required extension will allow the use of the Siemens sX traffic controllers. The introduction to the topic of traffic engineering, traffic controllers configuration and the dynamic traffic control is discussed first. The next part is dedicated to describing the implementation of the existing simulation system. The following chapter describes the use of the sX traffic controllers and also describes the design of the required extension of the existing simulation system, for a purpose of enabling the use of sX traffic controllers. This is followed by a chapter describing the implementation of this extension. The last chapter is devoted to testing the whole system in terms of functionality and performance. Finally, other possibilities for the future development are outlined.

Klíčová slova

doprava, dopravní řadič, SUMO, konfigurace řadiče, světelná křižovatka, sX advanced 3.0.1

Keywords

traffic, traffic controller, SUMO, controller configuration, traffic lights, sX advanced 3.0.1

Citace

DRESSLER, David. *Simulace dopravy za využití dopravních řadičů*. Brno, 2019. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. RNDr. Jitka Kreslíková, CSc.

Simulace dopravy za využití dopravních řadičů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením paní doc. RNDr. Jitky Kreslíkové, CSc. a mého konzultanta Ing. Petra Švani. Další informace mi poskytnul pan Bc. Martin Sláma. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
David Dressler
20. května 2019

Poděkování

Zde bych rád poděkoval své vedoucí, paní doc. RNDr. Jitce Kreslíkové, CSc., svému konzultantovi, panu Ing. Petru Švaňovi a panu Bc. Martinovi Slámovi ze společnosti Siemens, s.r.o. za cenné rady a odbornou pomoc při řešení této diplomové práce.

Obsah

1	Úvod	3
2	Konfigurace dopravních řadičů společnosti Siemens, s.r.o.	4
2.1	Dopravní řadič	4
2.2	Konfigurace dopravního řadiče	4
2.3	Principy dynamického řízení křižovatek	8
2.4	Doplňující konfigurace programu pro návrh křižovatek	10
3	Stávající simulační systém	11
3.1	Simulátor	11
3.2	Webový klient	15
4	Návrh rozšíření stávajícího simulačního systému	19
4.1	Využití dopravního řadiče sX	19
4.2	Požadavky na simulátor	20
4.3	Návrh rozšíření	21
5	Popis implementace	26
5.1	Implementace rozšíření	26
5.2	Změny ve stávajícím simulačním systému	28
5.3	Komunikace mezi simulátorem a dopravním řadičem sX	30
5.4	Rozdíly při komunikaci s reálným dopravním řadičem sX	32
5.5	Problémy při práci se simulačním systémem	33
6	Testování simulačního systému	34
6.1	Testování původních funkcí simulačního systému	35
6.2	Testování rozšíření simulačního systému	36
6.3	Výkonnostní testování	39
6.4	Shrnutí výstupů testování	42
7	Závěr	43
7.1	Námět na budoucí úpravy simulačního systému	44
	Literatura	45
A	Obsah přiloženého CD	46
B	Ukázky úpravy webového rozhraní	47

C Testovací případy pro Black-box testování	49
D Testovací případy pro White-box testování	52

Kapitola 1

Úvod

Světelné křižovatky již po několik desetiletí řídí provoz na frekventovaných dopravních uzlech. Postupem času vznikaly nové možnosti, jak jejich řízení zefektivnit. Stojí za tím zejména nástup moderních dopravních řadičů. Základem je však dynamické řízení dopravy, které umožňuje reagovat i na výkyvy dopravy v rámci trvání několika sekund. Dává nám tím nástroje pro reakci na tvoření dlouhých kolon vozidel, ale také pro zmírnění hlukové zátěže okolních obyvatel a zvýšenou kvalitu ovzduší. Nebo také například pro zvýšený komfort jízdy městskou hromadnou dopravou.

V této diplomové práci se věnuji problematice dynamického řízení dopravy a možnostem dopravních řadičů. Na základě již implementovaného simulačního systému navrhnu a implementuji jeho rozšíření tak, aby bylo možné využít logiky řízení dopravních řadičů sX společnosti Siemens, s.r.o. Tím bude simulační systém více přiblížen reálnému řízení křižovatek.

Průběh řešení zadání této diplomové práce je popsán v následujících kapitolách. Druhá kapitola se týká problematiky řízení křižovatek, konfigurací dopravních řadičů a principů dynamického řízení křižovatek. Tato část se týká spíše oboru dopravního inženýrství než informačních technologií, nicméně nejedná se o první obor, kde informační technologie hrají svou roli. Třetí kapitola je věnována popisu implementace stávajícího simulačního systému, jehož funkce budou nadále rozšiřovány za účelem více se přiblížit reálnému provozu na křižovatkách. Čtvrtá kapitola se věnuje způsobu využití dopravních řadičů sX společnosti Siemens, s.r.o. a také samotnému návrhu rozšíření stávajícího simulačního systému tak, aby mohl logiku těchto dopravních řadičů využívat. Pátá kapitola je věnována popisu implementace vytvořeného návrhu rozšíření stavajícího simulačního systému. V šesté kapitole následuje testování kompletního systému po implementaci rozšíření z hlediska funkčnosti a výkonu. Poslední kapitola shrnuje odvedenou práci při vytváření této diplomové práce. Také jsou zde uvedeny náměty na další možnosti rozšíření pro budoucí úpravy vytvořeného simulačního systému.

Kapitola 2

Konfigurace dopravních řadičů společnosti Siemens, s.r.o.

V této kapitole bude popsán dopravní řadič společnosti Siemens, s.r.o. Nejprve o jaké zařízení se jedná a jaký je jeho účel. Poté bude zejména popsána jeho konfigurace. Budou vymezeny a popsány její jednotlivé části, které definují chování dopravního řadiče. Důraz bude kladen zejména na signální plány, signální skupiny a detektory.

2.1 Dopravní řadič

Jako dopravní řadič se označuje hardwarové zařízení, které slouží k řízení křižovatky. Dopravní řadič je propojen se semaforem (SSZ¹) a případně také s detektory. Ty nemusí být v křižovatce zabudovány. U semaforů řídí zobrazování příslušných světelných signálů. Detektory jsou využívány zejména při dynamickém řízení křižovatky.

Dopravní řadič sX advanced 3.0.1

Jedná se o řadič vyvinutý společností Siemens, s.r.o. Nahrazuje předešlé verze dopravních řadičů C800 a C900. Novinkou je dostupné webové rozhraní, které slouží k přístupu k datům dopravního řadiče. Umožňuje nahrávání nových konfigurací, změny signálních plánů i sledování aktuální činnosti řadiče. Pro veškeré úkony není třeba vypnout řízení křižovatky. Navýšil se také pracovní rozptyl teplot na interval od - 40 °C do 60 °C. Řadič splňuje úroveň bezpečnostní integrity SIL3². Více informací o dopravním řadiči sX lze získat z dokumentu s jeho popisem [7] a z dokumentu s porovnáním verzí řadičů C800/C900 a sX advanced 3.0.1 [6], oba jsou dostupné na přiloženém CD.

2.2 Konfigurace dopravního řadiče

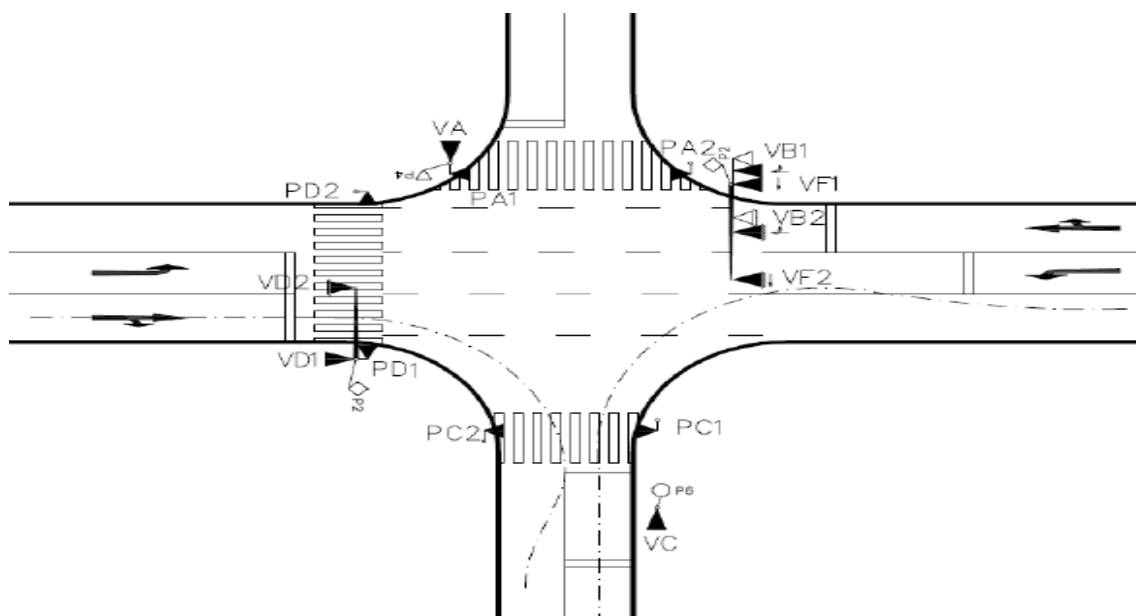
Konfigurace definuje chování dopravního řadiče. Jejím obsahem jsou zejména následující parametry, které v potřebné míře reflektují stavební uspořádání křižovatky a také ovlivňují její zobrazování světelných signálů:

¹světelné signalizační zařízení

²Ne více než jedna nebezpečná chyba v bezpečnostní funkci za 1000 let.

- signální skupiny,
- detektory,
- signální plány.

Při návrhu konfigurace dopravního řadiče se dbá zejména na bezpečnost všech účastníků provozu a průjezdnost dané křižovatky. Dopravní řadiče lze také zařadit do většího centrálního řízení dopravy. Poté lze sladěním signálních plánů sousedních křižovatek docílit lepší průjezdnosti. Například se jedná o koordinaci řízení zelená vlna. Jejímž cílem je, aby při dodržování určité rychlosti mohla vozidla projet co nejvyšším počtem křižovatek aniž by zastavila.



Obrázek 2.1: Nákres křižovatky, Vx jsou SSZ pro vozidla, Px pro chodce (Převzato z [4])

Signální skupina

Signální skupina je soubor návěstidel, která udávají v každém okamžiku pro jeden vjezd vozidel nebo vstup chodců na jeden přechod stejný signální obraz. Signální skupinu může tvořit i jediné návěstidlo [4]. Většinou je přítomno návěstidel více. Příkladem jediného návěstidla je doplňková zelená šipka pro odbočení vpravo, vyklizovací šipka, či návěstidlo pro tramvaje. Pro vozidla se využívají tříbarevné návěstidla. Jedná se o signál „Stůj!“ (červená barva), signál „Pozor!“ (žlutá barva) a signál „Volno!“ (zelená barva). Plné světelné signály mohou být upraveny směrovou šipkou, poté je možné pokračovat v jízdě pouze v povoleném směru. Důležitá je, krom doby trvání těchto signálů, také doba přechodu mezi stavy „Stůj!“ a „Volno!“. Tato doba odpovídá době trvání signálu „Pozor!“.

V konfiguraci dopravního řadiče musí být všechny signální skupiny definovány. Řadič typu sX uchovává informaci o typu signální skupiny. Možné typy jsou například následující:

- vozidla,
- odbočení vlevo,

- odbočení vpravo,
- autobus,
- tramvaj,
- jízdní kolo,
- pěší,
- nevidomé osoby,
- přerušované žluté světlo,
- jízdní kolo s pěší zároveň,
- speciální skupina.

Řadič typu sX také uchovává úhel signální skupiny vůči centru křižovatky. Pomocí tohoto úhlu lze identifikovat dopravní tok odpovídající signální skupině. Bez této informace by tedy nebylo možné identifikovat, o jaké rameno křižovatky se jedná. Pokud jsou pro signální skupinu využity semaforey se směrovými šipkami, je nutné uchovávat také informaci o povoleném směru jízdy.

Počet dopravních pruhů v daném směru jízdy a celková topologie křižovatky nepatří mezi uchované informace. Pro řadič nejsou tyto informace relevantní, řídí se na základě signálních skupin. Jestli je tedy například pro směr rovně jeden nebo dva dopravní pruhy není důležité, oba tyto případy by spadaly pod stejnou signální skupinu. Celková topologie křižovatky také není důležitá, protože světelné signály opět přísluší dané signální skupině. Například signální skupina 1 má signál „Volno!“ a signální skupina 2 má signál „Stůj!“.

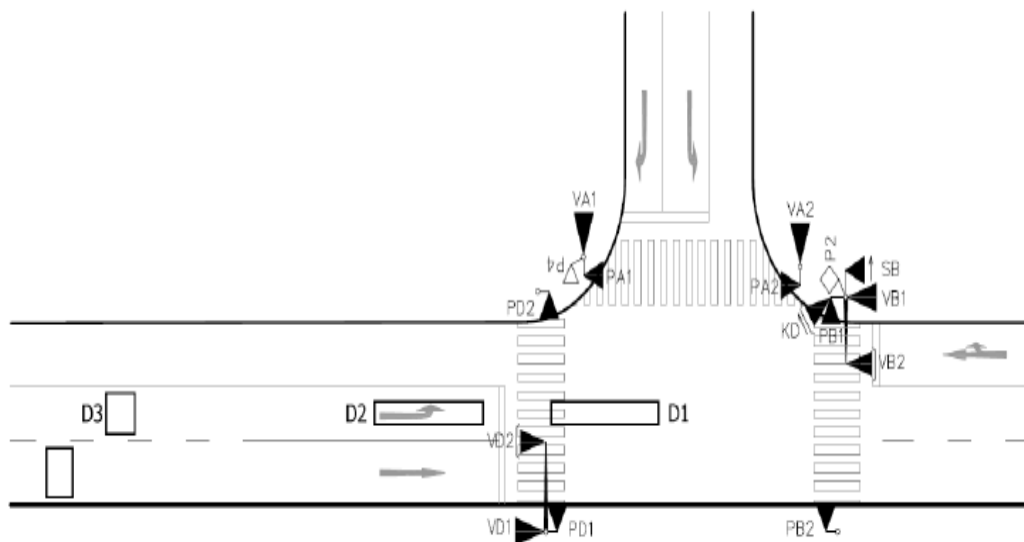
Detektor

Detektory slouží především k detekování projíždějících vozidel křižovatkou. Existuje více typů detektorů, liší se zejména využitou technologií při detekci. Každý typ nabízí jiné možnosti detekce, proto je běžně na křižovatce detektorů více a také různých typů. Využívají se například tyto typy detektorů [5]:

- Jednoduché indukční smyčky - pro detekci vozidel.
- Dvojité indukční smyčky - pro detekci vozidel, nabízí oproti jednoduchým indukčním smyčkám navíc například měření rychlosti, směru jízdy, či vzdálenosti mezi dvěma projíždějícími vozidly.
- Tlačítka - pro detekci chodců.
- Radarové detektory - pro detekci vozidel i chodců.
- Detektory městské hromadné dopravy - pro detekci vozidel MHD.
- Video detektory - pro detekci vozidel i chodců.
- PIR¹ detektory - pro detekci vozidel i chodců.

¹pasivní infračervený detektor

- Magnetické detektory - pro detekci vozidel.
- Rádiové detektory - pro měření doby jízdy.



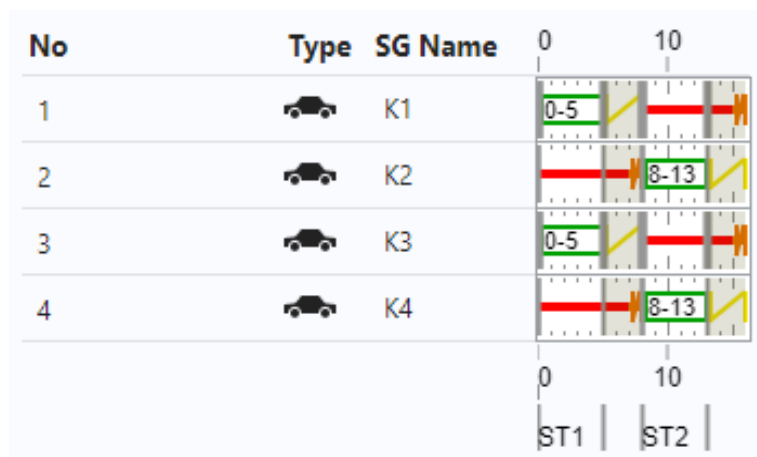
Obrázek 2.2: Schéma křižovatky s detektory typu indukční smyčky (Dx) (Převzato z [4])

Signální plán

Signální plán je program řízení světelného signalizačního zařízení (zkratka SSZ), který určuje pořadí a délku signálů volno jednotlivých signálních skupin [4]. Struktura signálního plánu se skládá zejména z následujícího:

- Rozdělení, počet a pořadí fází - fáze je časový interval, kdy má současně signál volno více navzájem nekolizních dopravních toků. Je nutné vytvořit fázové schéma, které přiřazuje fázím jejich pořadí a také definuje jejich počet. Existují také podmíněné kolizní dopravní toky, jedná se například o odbočení vlevo.
- Fázové přechody - jedná se o časové úseky mezi signály „Volno!“ skupin končící fáze a následné fáze.
- Mezičasy - jedná se o časový interval od konce signálu po začátek signálu „Volno!“ kolizní signální skupiny. V této době musí poslední vozidlo bezpečně opustit kolizní plochu křižovatky.
- Doby signálů „Volno!“ - jedná se o rozdělení celkové doby signálu „Volno!“ mezi jednotlivé směry dopravních toků.
- Délka cyklu - jedná se o součet dob signálů „Volno!“ a rozhodujících mezičasů.
- Okrajové podmínky - jedná se o nejnižší možné doby trvání světelných signálů.

Na obrázku 2.3 jsou fázové přechody podbarveny šedou barvou. Sloupce No, Type a SG Name označují číslo, typ a jména signální skupiny. Na časových osách v pravé části obrázku lze vidět posloupnost jejich fází (podbarveno bílou barvou) a fázové přechody (podbarveno šedou barvou). Zelená a červená barva značí příslušné světelné signály.



Obrázek 2.3: Ukázka signálního plánu (Snímek obrazovky)

Signální plány se dělí na pevné a dynamické. Pevné signální plány, nebo-li orientované na signální skupiny, mají fixně definované řízení. V některých případech jsou postačující a výhodou je jednoduchost tohoto řešení. Dynamické signální plány, nebo-li orientované na etapy, umožňují dynamické řízení křižovatek.

2.3 Principy dynamického řízení křižovatek

Při využití principu pevného řízení křižovatek se předpokládá ustálená intenzita provozu na křižovatce, tedy pouze s dlouhodobými změnami. Oproti principu pevného řízení křižovatek je u principu dynamického řízení křižovatek reagováno i na krátkodobé výkyvy (v jednotkách sekund). Reflektuje se tedy reálný aktuální stav provozu. Jsou pak ale také vyžadovány vyšší technické náklady, je totiž potřeba navíc zpracovávat údaje o aktuální dopravní situaci. Mezi získávané údaje patří:

- doba jízdy,
- délka vzdutí,
- intenzita provozu,
- jízdní rychlost,
- počet zastavení,
- doba zdržení.

Získávání těchto údajů závisí na technických možnostech konkrétní křižovatky. Mezi běžněji využívané nepatří poslední dva zmíněné. Při využívání se také musí stanovit jejich vzájemné priority, protože může vzniknout rozpor mezi jednotlivými údaji.

Na základě nastavení dopravního řadiče a případně vyhodnocení uvedených údajů probíhá řízení křižovatky. Rozhodování o změně může probíhat pouze následovně:

- Mimo průběh signálního plánu - rozhoduje se o volbě signálního plánu. Tato volba může být buď časově závislá nebo dopravně závislá. U časově závislé volby je předem nastaveno, jaké signální plány mají být aktivní v daném čase během dne a také lze nastavit různé signální plány podle kalendářního data. Dopravně závislé rozhodování probíhá podle aktuální dopravní situace.
- V průběhu signálního plánu - rozhoduje se o změně v signálním plánu v jeho průběhu. Změny se týkají délky signálu „Volno!“ a změny pořadí fází. Není dovoleno měnit mezičasy a minimální doby signálů volno kvůli bezpečnosti provozu.

Při změně doby signálu „Volno!“ se buď zkracuje délka cyklu, což prioritizuje jiný dopravní proud nebo je naopak prodloužena na úkor jiného dopravního proudu. Například pokud je detekováno vozidlo přijíždějící na křižovatku a je aktivní signál „Volno!“, může být prodlouženo jeho trvání o definovaný čas. Tato dodatečná doba má definované svoje minimální i maximální možné trvání. Je také omezena maximální délkou trvání fáze.

U změn fází se buď mění pouze pořadí předem daného sledu, což je účelné například v souvislosti s preferencí vozidel MHD, nebo lze do jejich sledu vložit fáze nová. Ta se vkládá při výzvě trvale se nevyskytujících dopravních proudů. Jedná se například o chodce, cyklisty, MHD a odbočující vozidla. Pokud není výzva zadána, může se čas využít pro ostatní účastníky provozu nebo se zkrátí délka cyklu. Pro chodce a případně cyklisty se k zadání výzvy využívají tlačítka nebo dotykové senzory. Vozidla se detekují pomocí indukčních smyček. Lze takto zařadit fázi pro vyklizovací šipku. Jednostopá vozidla a cyklisti pomocí šikmo umístěných indukčních smyček. Vozidla MHD musí provést včasné přihlášení.

Městská hromadná doprava

Při dynamickém řízení křižovatky lze prioritizovat jedoucí vozidla MHD. Pro jejich detekci se využívají i další specifické možnosti, například:

- Trolejové detektory - pro tramvaje a trolejbusy, poskytují jednobitovou informaci o přítomnosti.
- Datové indukční smyčky - nabízejí další informace oproti klasickým indukčním smyčkám.
- Vysílací a přijímací systémy - slouží pro komunikaci vozidel MHD s dopravním řadičem křižovatky. Jsou založeny na vysokefektivním přenosu, na infračervené technologii nebo na autonomním zjišťování polohy pomocí satelitů.

Důležité je zejména včasné detekování vozidel MHD. Pokud je zastávka od světelných semaforů vzdálená více než 150 m, může přihlášení proběhnout až 350 m před zastávkou. Pokud je zastávka v těsné blízkosti světelných semaforů probíhá detekce před příjezdem na zastávku nebo během zastavení MHD na zastávce. Mimo jiné například i pomocí informace o uzavření dveří.

2.4 Doplnující konfigurace programu pro návrh křižovatek

Doplňující konfigurace slouží k návrhu sítě křižovatek. Účelem je koordinovat světelné signály sousedních křižovatek a tím docílit efektivnějšího řízení dopravy. Je poté možné re-agovat i na události mimo rozsah řízení dané křižovatky, ale vznikne potřeba rozsáhlejší analýzy dopravy. Příkladem využití v praxi je například koordinace řízení - zelená vlna.

Zelená vlna

Koordinace řízení - zelená vlna je docilována sladěním signálních plánů sousedních křižovatek za účelem toho, aby vozidla při dodržování určité rychlosti mohla projet co nejvíce křižovatkami bez zastavení. Při řízení zelené vlny je potřeba brát ohled na různé skupiny účastníků dopravy a zvýhodňovat hlavní dopravní směry. Tento způsob řízení se uplatňuje pouze při vzdálenostech světelných signalizačních zařízení do 750 metrů, výjimečně do 1000 metrů. Předpokladem pro koordinaci v zelené vlně je také totožná délka cyklu na všech křižovatkách, krátkodobé odchylky v délkách cyklu se musí vyrovnávat. Mezi druhy koordinace v zelené vlně patří:

- Progresivní systém - doba světelného signálu „Volno!“ na křižovatkách následujících za sebou na sebe navazuje tak, aby odpovídala vypočtené době jízdy od stopčáry ke stopčáře. Světelné signály „Volno!“ jsou tedy na křižovatkách aktivovány postupně tak, aby následující křižovatka signalizovala světelný signál „Volno!“ po uplynutí této vypočtené doby.
- Simultánní systém - v průběhu zelené vlny je stejný světelný signál současně na všech křižovatkách. Použití je vhodné při vzdálenostech křižovatek do 100 metrů. Tyto křižovatky tedy signalizují světelný signál „Volno!“ současně.

Mohou se také využívat rychlostní signály doporučující rychlost vozidlům. Důvody pro jejich využití jsou například:

- Vedení vozidel tak, aby přijela k následující křižovatce v době počátku světelného signálu „Volno!“.
- Překlenutí posunů nebo délek cyklu mezi sousedními křižovatkami.
- Zabránění rozvolnění kolony vozidel. Využívá se při větších vzdálenostech mezi křižovatkami.

Kapitola 3

Stávající simulační systém

V této kapitole bude popsán stávající simulační systém. Nejprve stručně jednotlivé technologie a nástroje, které jsou pro jeho chod vyžadovány. Poté zejména naprogramovaná logika a jeho uživatelské rozhraní. Tyto informace budou nadále využity při návrhu rozšíření stávající aplikace. O důvodech, proč byl stávající simulační systém navrhnut tímto způsobem, se lze více dočíst z diplomové práce „Simulátor dopravních infrastruktur a situací. [11]“.

3.1 Simulátor

V této části aplikace je vytvořena logika simulačního systému. Simulátor byl implementován v jazyce Java 8. Využit byl framework SpringBoot¹ ve verzi 2.0.0. Pro sestavení projektu, získání externích závislostí i běh projektu byl využit nástroj Gradle² ve verzi 4.5.1.

Využité technologie a nástroje

- SUMO [8] - volně dostupná „open-source“ simulační platforma, specializovaná na simulaci dopravy.
- TraCI - rozhraní pro SUMO umožňující vzdálený přístup k běžící simulaci.
- TraaS [2] - implementace TraCI rozhraní v jazyce Java. Pracuje jako webová služba.
- Duarouter - nástroj sloužící k vyhledávání cest v simulační síti.
- Netconvert - nástroj sloužící ke konverzi simulačních sítí do formátu pro SUMO.

Popis struktury projektu

Projekt byl rozdělen na pět následujících modulů:

- TrafficSimulatorService - spustitelný modul, který propojuje jednotlivé moduly. Obsahuje také HTTP rozhraní.
- Simulator - slouží ke spuštění a zastavení simulace, získávání informací za běhu a také změnu parametrů simulace.

¹<https://spring.io/projects/spring-boot/> [Online; navštíveno 19.5.2019]

²<https://gradle.org/> [Online; navštíveno 19.5.2019]

- Parser - slouží pro zpracování XML souboru s konfigurací křižovatky.
- Generator - z vnitřní reprezentace dat vygeneruje validní konfigurační soubory pro nástroj SUMO [1].
- Shared - obsahuje společné třídy modulů, výjimky, enumerátory a konstanty.

Při spuštění aplikace je vytvořena nová prázdná instance třídy „Situation“. Ta reprezentuje simulační situaci. Samotný běh aplikace byl poté rozdělen na přípravnou fázi a fázi samotné simulace.

Přípravná fáze

Do této fáze patří operace manipulující s elementy, které nejsou v průběhu simulace měněny. Jedná se zejména o křižovatky a dopravní pruhy. Tato fáze probíhá po spuštění aplikace a vždy, když je vyvolána změna v zobrazovaných křižovatkách. Takové požadavky vždy zastaví simulační fázi, provedou změnu a poté je možné simulaci znovu spustit. Přípravná fáze se dělí na:

- Zpracování vstupů a konfigurací - nejprve je vytvořena počáteční vnitřní reprezentace simulace na základě konfigurace řadiče sX ve formátu XML. Také dle polohy křižovatky a jejího úhlu. XML soubory se zpracovávají pomocí knihovny „org.w3c.dom“ jazyka Java. Z konfigurace jsou získávány informace o signálních skupinách, detektorech a signálních plánech.
- Generátor SUMO konfigurace - z vnitřní reprezentace se vytváří konfigurační soubory pro SUMO. Generátor využívá nástroje netconvert pro generování konečného souboru s definicí simulační sítě. Dále nástroje duarouter pro vyhledávání cest v dopravní situaci. Poté ještě probíhá úprava do konečného formátu vytvořeného souboru a uložení vygenerovaných tras do vnitřní reprezentace simulace.

Fáze simulace

Do této fáze patří operace prováděné za běhu simulace. Jedná se například o provedení simulačního kroku nebo přepnutí signálního plánu. Fáze simulace probíhá po přípravné fázi.

Nejprve proběhne spuštění programu SUMO s vytvořenými konfiguračními soubory. Poté je spuštěna samotná simulace pomocí rozhraní TraCI. Simulace přistupuje k vnitřní reprezentaci s informacemi o signálních programech, fázích, detektorech a vygenerovaných trasách.

Při inicializaci simulace se vytvoří pomocí „SimulationControlFactory“ instance třídy „SimulationControlImpl“, která umožňuje obsluhu běžící simulace. Příchozí požadavky na změnu v běžící simulaci jsou ukládány do fronty. Každý simulační krok je fronta kontrolována a vykonávají se akce v ní uložené. Za běhu simulace je možné měnit hustotu provozu a signální program.

Dynamické řízení křižovatek je realizováno pomocí mechanismu „Traffic Actuation“. Ten umožňuje prodloužení aktivní fáze signální skupiny při detekování vysoké intenzity provozu. Vozidla se generují od startu simulace v závislosti na definované hustotě provozu. Generované jsou náhodně na vstupních bodech do simulace a je jim náhodně přiřazena jedna z vygenerovaných tras s počátkem v daném vstupním bodě. Implementované volby typu generovaných vozidel jsou:

- Automobil,
- Dodávka,
- Autobus MHD,
- Dálkový autobus,
- Nákladní automobil.

Simulace je ukončena až při požadavku na její ukončení nebo při potřebě vygenerování nových konfiguračních souborů pro SUMO při změně parametrů spadajících do přípravné fáze.

Konfigurace aplikace

Aplikace využívá souboru „application.properties“ sloužícího jako externí konfigurace. Nastavitelné parametry pomocí tohoto souboru jsou:

- Číslo portu pro HTTP komunikaci.
- Cesta k adresáři bin nástroje SUMO.
- Cesta k adresáři pro ukládání vygenerovaných konfiguračních souborů pro SUMO.
- Maximální počet zároveň přítomných vozidel v simulaci.
- Délka simulačního kroku.
- Vzdálenost křižovatek v simulaci v jednotkách metrů.
- Délka ramen křižovatek v simulaci v jednotkách metrů.
- Názvy dalších souborů, které jsou využívány při simulaci.

V průběhu simulace mohou vznikat kolize vozidel. Ty jsou pouze zaznamenány.

HTTP rozhraní

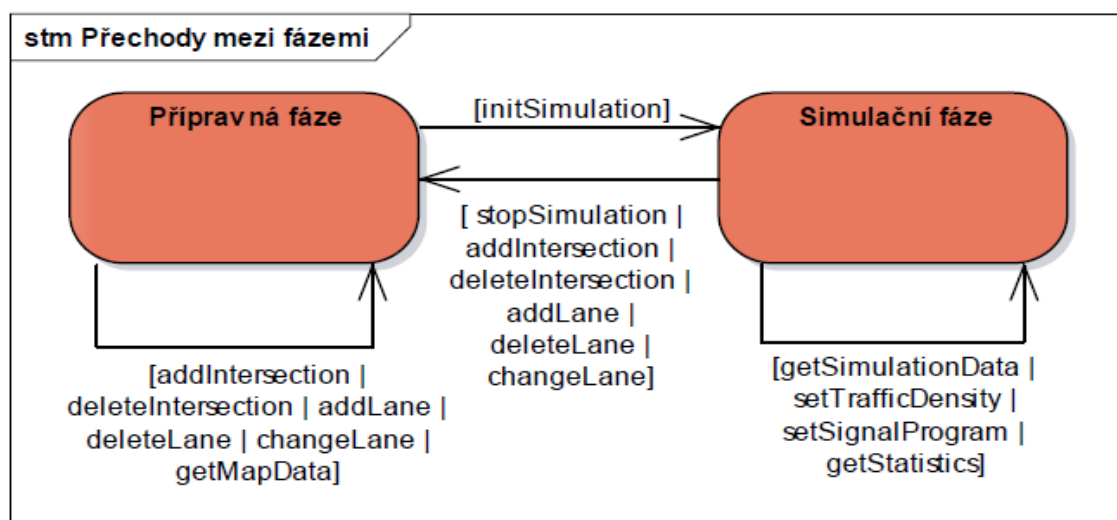
Dostupné požadavky jsou, až na jednu výjimku, založeny na HTTP [10] metodě GET. Odpovědi na požadavky klienta jsou instancí třídy „JsonMessage“ a obsahují číselný kód odpovědi, stavový text a mohou obsahovat také objekt s daty. Dostupné požadavky jsou reprezentovány přechody mezi vnitřními stavy aplikace konečným automatem na obrázku 3.1. Dostupné metody reprezentující požadavky tedy jsou:

- `getMap` - poskytuje klientovi všechny potřebné údaje pro vykreslení statického podkladu vizualizace. Z vnitřní reprezentace získá informace o všech křižovatkách a dopravních pruzích, získá také geometrická data pro vykreslení jejich přesných tvarů.
- `addIntersection` - slouží pro přidání křižovatky. Jediný požadavek aplikace založený na HTTP metodě POST. Parametrem je instance třídy „JsonReceivedConf“. Ta obsahuje booleovskou hodnotu značící, jestli se jedná o první přidanou křižovatku. Obsahuje také úhel natočení křižovatky zadaný uživatelem a obsah souboru s konfigurací. Pokud se jedná o první přidávanou křižovatku, zadávají se také typy generovaných vozidel.

Pokud se nejedná o první přidávanou křižovatku, zadává se křižovatka, ke které se má sousední křižovatka přidat a také pozice vůči ní (nahore, dole, vlevo nebo vpravo). První křižovatka je umístěna v souřadnicovém systému v bodě [0, 0]. Umístění ostatních křižovatek se vypočítá relativně k ní a kontrolují se případné kolize. Dále modul „Parser“ zpracuje konfiguraci nové křižovatky, křižovatka je přidána a pokud je to možné, jsou křižovatky automaticky propojeny. Poté modul „Generator“ vygeneruje nové konfigurační soubory pro SUMO.

- deleteIntersection - umožňuje vymazat křižovatku na základě jejího identifikátoru. Poté se také generují nové konfigurační soubory pro SUMO.
- addLane - slouží pro přidání nového dopravního pruhu k rameni křižovatky. Parametry jsou identifikátor křižovatky, identifikátor ramene křižovatky a zadané směry jízdy nového dopravního pruhu. Pořadí přidávaného pruhu je vypočítáno automaticky dle směrů všech pruhů daného ramene křižovatky. Probíhá kontrola, jestli je možné nový pruh přidat tak, aby nebyla ohrožena bezpečnost provozu. Poté se také generují nové konfigurační soubory pro SUMO.
- deleteLane - slouží pro vymazání dopravního pruhu na základě jeho identifikátoru. Alespoň jeden dopravní pruh musí zůstat v rameni křižovatky, což je kontrolováno před smazáním.
- changeLane - umožňuje změnu směrů jízdy z konkrétního dopravního pruhu. Parametrem je jeho identifikátor a také specifikované směry jízdy. Pokud je v rameni křižovatky více dopravních pruhů, proběhne jejich seřazení podle směrů jízdy. Poté se také generují nové konfigurační soubory pro SUMO.
- initSituation - slouží pro inicializaci simulace. Ověřuje se, jestli je možné simulaci provést kontrolou počtu křižovatek a validních cest. Při úspěšné kontrole se pomocí TraCI rozhraní spustí program SUMO.
- clearSituation - slouží pro smazání všech křižovatek.
- getDataMultipleStep - získává data pro simulaci. Parametrem je počet simulačních kroků, které má simulátor provést. Každý simulační krok nejprve zkontroluje, jestli je potřeba provést nějakou akci, provede mechanismus „Traffic Actuation“ (pouze pokud jsou přítomny detektory), vygeneruje nová vozidla a uloží data o simulaci. Poté také ukládá data, která se odešlou klientské aplikaci po provedení definovaného počtu simulačních kroků. Jedná se o souřadnice vozidel, úhel vozidel, data o rychlosti, ujeté vzdálenosti a aktuální doba zdržení na semaforu. Dále také data o nově vygenerovaných vozidlech, aktuálních světelných signálech signálních skupin a informace o simulačních krocích, signálních programech a fázích.
- setTrafficDensity - mění četnost vygenerovaných vozidel za jeden simulační krok na základě číselného parametru.
- setSignalProgram - slouží pro změnu signálního plánu křižovatky při běhu simulace. Kromě identifikátoru křižovatky je parametrem také řetězec identifikující nový signální plán.
- getStatistics - slouží pro získání statistik za dobu běhu celé simulace.

- stopSimulation - slouží pro zastavení simulace a přechod do počátečního stavu aplikace před inicializací.



Obrázek 3.1: Konečný automat s požadavky a interními stavy aplikace (Převzato z [11])

Statistická data simulace

Modul „Simulator“ po dobu běhu simulace shromažďuje data o spotřebě paliva, době čekání na semaforech, době strávené v simulaci a emisní data. Statistická data jsou rozdělena na:

- informace o všech vozidlech v rámci jednoho simulačního kroku.
- souhrné informace vozidla za průběh celé simulace.
- souhrné informace o celkovém stavu měřených parametrů všech vozidel za průběh celé simulace.

3.2 Webový klient

Tato část aplikace tvoří grafické uživatelské rozhraní. Webový klient je implementován v jazycích HTML, CSS a JavaScript. Využito je standardu ECMA verze 6, který podporuje definici tříd. Pro implementaci animací simulace je využito technologie SVG a pro práci s ní využito knihovny „Svg.js“¹ pro jazyk JavaScript. Pro vykreslení grafů statistických dat simulace je využito knihovny „Chart.js“² pro jazyk JavaScript. Využívá se také knihovna jQuery³, zejména pro zaslání HTTP požadavků.

Adresářová struktura

Klientská aplikace má vlastní adresářovou strukturu a byla vyvinuta separátně od simulační části. Kořenový adresář obsahuje:

¹<https://svgjs.com/docs/2.7/> [Online; navštíveno 19.5.2019]

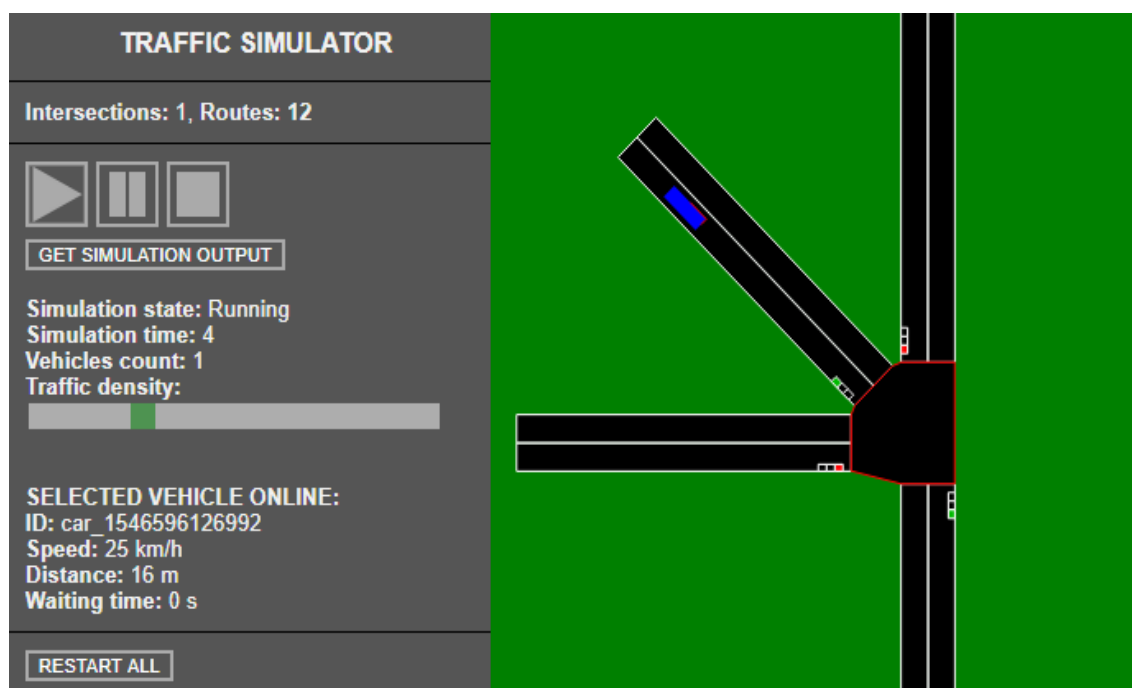
²<https://www.chartjs.org> [Online; navštíveno 19.5.2019]

³<http://jquery.com> [Online; navštíveno 19.5.2019]

- index.html - soubor pro přístup k webovému klientovi.
- css - adresář obsahující soubor „main.css“ s definicí vzhledu jednotlivých prvků.
- img - adresář obsahující ikony využívané v aplikaci.
- js - adresář se zdrojovými soubory v jazyce JavaScript.
- lib - adresář s externími JavaScript knihovnami.

Uživatelské rozhraní

Uživatelské rozhraní se dělí na dvě části. Vlevo je postranní řídicí panel, který obsahuje ovládací prvky a informace o simulaci. Zobrazované prvky se mění podle volby detailu. Například kliknutím na vozidlo v simulaci se zobrazí v detailu jeho informace. Při kliknutí na křižovatku se zobrazí možnost přidání sousední křižovatky. Při kliknutí na dopravní pruh se zobrazí možnost přidat sousední pruh a změnit směry jízdy. V pravé části se nachází vizualizace simulace. Operace uživatelského rozhraní se dělí na přípravnou fázi a fázi vizualizace simulace. Ukázka uživatelského rozhraní je zobrazena na obrázku 3.2.



Obrázek 3.2: Uživatelské rozhraní aplikace (Snímek obrazovky)

Přípravná fáze

V přípravné fázi probíhá překreslování statické části vizualizace. Provede se vždy při úspěšné změně, vkládání a mazání křižovatek nebo dopravních pruhů. Překreslení statické části je možné pouze při neběžící simulaci. Pokud simulace probíhá, je nejprve odeslán požadavek serveru na její zastavení a následně je restartována a začíná znovu od simulačního času 0.

Výběr existující křižovatky se provádí kliknutím myši na její střed. Označená křižovatka změní barvu svého středu a v postranním panelu se zobrazí její aktuální informace při běhu

simulace a také možnost smazání křižovatky nebo přidání sousední křižovatky. Výběr křižovatky proběhne pouze na straně klienta. Přidávání první křižovatky probíhá v modálním okně, které umožňuje nahrávání XML souboru s konfigurací křižovatky. Dále lze také zadat natočení křižovatky a typy generovaných vozidel v simulaci. Přidávání dalších křižovatek je možné pouze přidáváním sousední křižovatky k již existující vybrané křižovatce, a to ve čtyřech směrech. Typy vozidel se již nespecifikují. Kliknutím na potvrzovací tlačítko proběhne kontrola, jestli jsou vyplněny všechny povinné hodnoty formuláře a poté se data odešlou na server. Na serveru proběhne ověření přijetí dat a dále se otestuje, jestli je místo pro novou křižovatkou volné. Server vrací po zpracování odpověď na požadavek klienta. Při úspěchu se překreslí situace, při chybě se zobrazí chybová hláška. Při mazání křižovatky není brán ohled na okolní křižovatky, ty zůstanou beze změny. Pokud byla křižovatka propojena s jinými křižovatkami, jejich spojení zanikne.

Výběr dopravního pruhu probíhá podobně jako výběr křižovatky. V postraním panelu se kromě možnosti smazání zobrazí možnost změny směru jízdy z dopravního pruhu. Lze také přidat nový dopravní pruh sousedící s vybraným dopravním pruhem, který bude spadat pod stejnou signální skupinu. Upravovat lze pouze vstupní dopravní pruhy křižovatky. Alespoň jeden dopravní pruh musí v rameni křižovatky zůstat. Dopravní pruhy mají vliv na počet možných tras pro vozidla v simulaci.

Fáze vizualizace simulace

V této fázi probíhá vizualizace dynamické části simulace. Jedná se o pohyb vozidel a světelné signály semaforů. Simulační data jsou získávána od simulační části aplikace formou sekvence simulačních kroků a jsou uloženy do fronty. Při vizualizaci se vybere první prvek fronty. Dle informací o signálních skupinách se nastaví světelné signály. Nová vozidla se zobrazí na svých souřadnicích a pro již existující vozidla se vytvoří animace o délce trvání simulačního kroku. Animace probíhá od původního bodu souřadnic vozidla do nového bodu. Další simulační krok z fronty se vybere po době odpovídající délce simulačního kroku.

Při vizualizaci vozidel jsou ze simulátoru získávány souřadnice bodu uprostřed přední hrany vozidla ve směru jízdy. Okrajové body vozidla jsou dopočítány ze získaného úhlu, délky a šířky vozidla. Vozidlo je animované jako obdélník a navíc zobrazuje signalizaci brzdových a směrových světel. Každé vozidlo je přidáno do vizualizace při jeho prvním výskytu v simulačních krocích a odstraněno pokud již není obsaženo v následujícím simulačním kroku.

Při vizualizaci světelných signálů semaforů jsou semaforey přidány automaticky pro každý vstupní dopravní pruh, jejich pozice se v průběhu simulace nemění. Mění se pouze jejich světelné signály v závislosti na informaci v simulačních krocích.

Řízení simulace

Lze měnit stav simulace a získávat aktuální informace o vybraném prvku simulace. Délku simulačního kroku lze změnit pomocí konfigurace v souboru „application.properties“, aktuálně je nastavena na 0,5 s.

Spuštění simulace proběhne kliknutím na tlačítko „Start simulation“. Nejprve se odešle serveru požadavek na inicializaci simulace. Při úspěchu požadavku se začnou periodicky zasílat požadavky na provedení simulačních kroků a získávání simulačních dat. Požadavky jsou posílány s pěti simulačními kroky, pokud délka fronty se simulačními kroky klesne pod tři prvky, je zažádáno o další simulační data. Současně probíhá vizualizace simulace.

Pozastavení simulace proběhne kliknutím na tlačítko „Pause simulation“. Běžící simulace se pozastaví po ukončení animace nejbližšího simulačního kroku a přestanou se zasílat požadavky na další simulační data. Vozidla zůstanou na svých aktuálních pozicích a světelné signály semaforů zůstanou stejné. Při následném uvedení simulace do běžícího stavu animace plynule pokračuje.

Zastavení simulace proběhne kliknutím na tlačítko „Stop simulation“. Ať už u běžící nebo pozastavené simulace se ze situace odeberou všechna vozidla a semaforey jsou nastaveny do vypnutého stavu.

V postranním panelu jsou zobrazovány informace o aktuální běžící simulaci. Jedná se o její stav, simulační krok a počet vozidel. Při výběru křižovatky se zobrazují informace o její fázi a signálním programu. Signální program je možné změnit výběrem ze seznamu dostupných signálních programů křižovatky. Při výběru nové položky je odeslán požadavek na simulační část aplikace. Projevení změny není okamžité kvůli způsobu získávání simulačních dat. Při výběru vozidla se zobrazují informace o jeho rychlosti, ujeté vzdálenosti a době zdržení na semaforech. Dále je při běžící simulaci možné v postranním panelu pomocí posuvníku měnit hustotu dopravy navýšením rychlosti generování nových vozidel. Každá změna posuvníku odešle požadavek simulační části aplikace. Projevení změny také není okamžité.

Kapitola 4

Návrh rozšíření stávajícího simulačního systému

V této kapitole bude popsáno navržené rozšíření, které umožní stávajícímu simulačnímu systému využívat logiku řízení reálného dopravního řadiče od společnosti Siemens, s.r.o. Navržena bude také úprava uživatelského rozhraní.

4.1 Využití dopravního řadiče sX

Abych mohl navrhnout požadované rozšíření stávající aplikace, musel jsem po nastudování problematiky a seznámení se se stávajícím simulačním nástrojem zjistit, jak je možné s dopravním řadičem sX pracovat. Ze společnosti Siemens, s.r.o. jsem obdržel kopii virtuálního počítače, na kterém běží simulovaný dopravní řadič sX. Obdržel jsem také potřebnou dokumentaci pro instalaci a základní dokumentaci pro vývojáře. Pro virtualizaci bylo využito nástroje Oracle VirtualBox¹.

Po zprovoznění virtuálního počítače s řadičem sX jsem se seznamoval s jeho webovým rozhraním. To umožňuje nahrání konfigurace a také přístup k informacím o detektorech, signálních skupinách a signálních plánech. Také vizualizuje průběh řízení křižovatky, konkrétně průběh fází jednotlivých signálních skupin a zobrazuje informaci, který signální plán je aktuálně aktivní. Nabízí také možnost spuštění simulace na jednotlivých detektorech a tím simulovat detekci vozidel. Z hlediska použitelnosti pro vytváření požadovaného rozšíření je však nepraktické tyto úkony řešit pomocí tohoto webového rozhraní. Je potřeba provádět tyto akce na pozadí aplikace a ovládat je webovým rozhraním současného simulačního systému.

Dopravní řadič sX také umožňuje komunikaci s ním pomocí rozhraní nazvaného „C-Control“. K tomuto rozhraní je možné přistupovat pomocí kódu v jazyce Java. Pro ověření správnosti komunikace s dopravním řadičem jsem na základě dohody s mým konzultantem vytvářel malé demo aplikace. Ty ověřovaly správnost využití tohoto rozhraní. Nejprve bylo z programu napsaného v jazyce Java zajištěno ověření, že je dopravní řadič spuštěný. Pro nahrání jeho konfigurace rozhraní „C-Control“ využít nelze, existuje však externí nástroj configUploader, který nahrání konfigurace umožňuje. Poté již pomocí rozhraní „C-Control“ bylo možné zajistit následující úkony:

¹<https://www.virtualbox.org> [Online; navštíveno 19.5.2019]

- Simulaci detektorů křižovatek.
- Zásílání informace, že projelo vozidlo.
- Zjišťování stavu detektorů a světelných signálů.

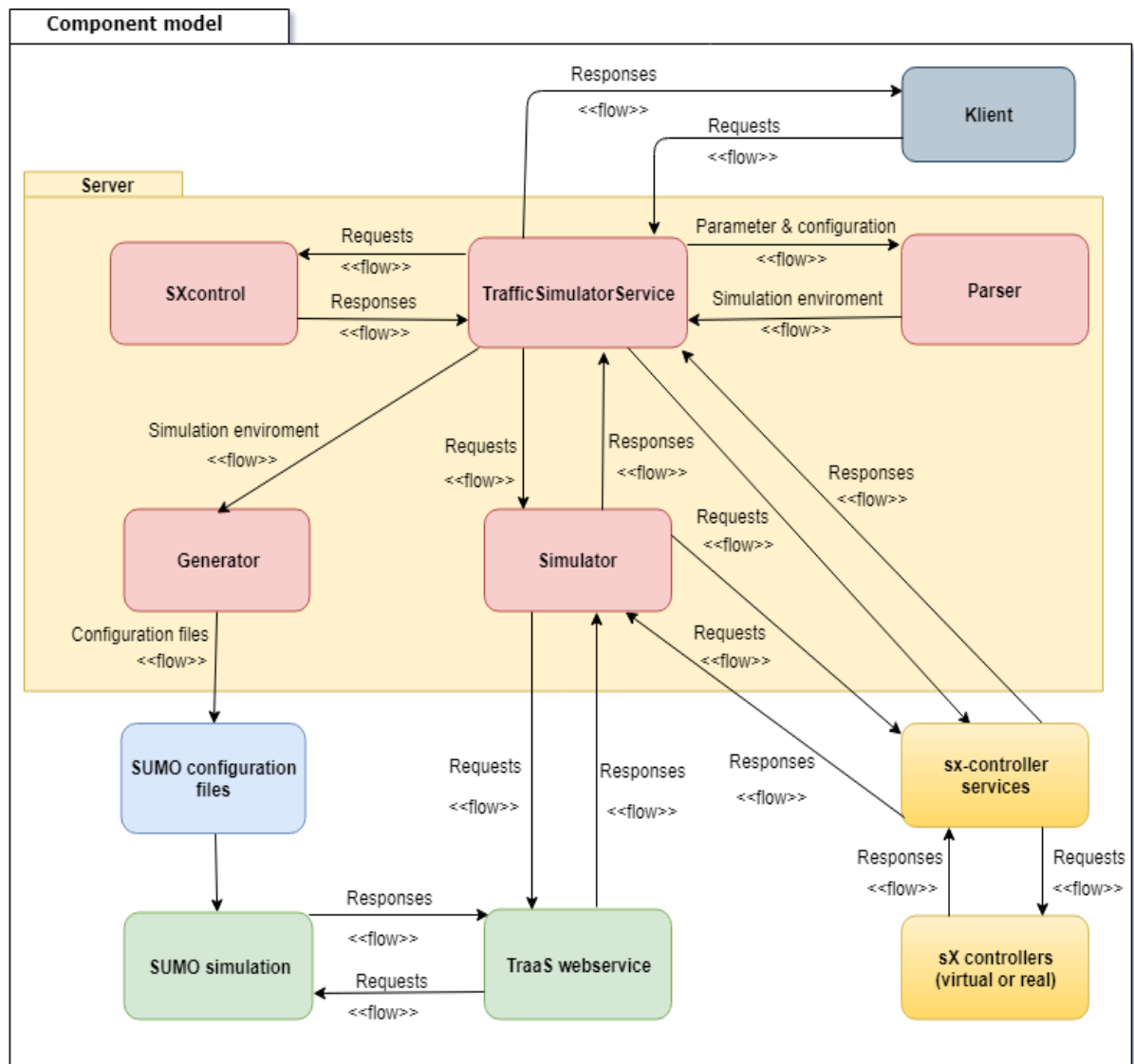
Bylo také potřeba zajistit korektní využití více virtuálních strojů s dopravním řadičem sX zároveň na jednom hardwaru. Je pevně definováno síťové rozhraní, které virtuální stroj má, obvykle na jednom hardwaru běží pouze jeden. Síťové rozhraní je možné změnit pouze po spuštění virtuálního počítače a musí zůstat v definovaném rozsahu sítě s 255 adresami. Počet virtuálních počítačů je ale omezen kvůli hardwarovým nárokům pro jejich běh.

Je také možné využít reálných dopravních řadičů sX, které mají totožné rozhraní jako jejich virtualizované verze.

4.2 Požadavky na simulátor

Po konzultaci s mým konzultantem byly určeny tyto požadavky pro rozšíření simulátoru o využívání dopravních řadičů sX:

- Zachování podpory křižovatek, které mají 2 až 4 ramena.
- Zachování zohlednění detektorů.
- Zachování automatického spojování křižovatek, pokud jich je více než jedna.
- Zachování možnosti parametry simulace měnit z webového rozhraní. Změna parametrů může mít za následek restart simulace.
- Pro samotnou simulaci se zachová využití nástroje SUMO. U křižovatek bude možné zvolit, jestli má být řízena nástrojem SUMO nebo dopravním řadičem sX (virtualizovaným nebo reálným).
- Uživatel bude moct simulaci konfigurovat následujícími možnostmi:
 - Přidáním nové křižovatky do simulace, definováním její pozice vzhledem k existující infrastruktuře v simulaci a volbě, jestli má být křižovatka řízena nástrojem SUMO nebo dopravním řadičem sX.
 - Přidáním jízdního pruhu do ramena křižovatky a definováním možných směrů jízdy z tohoto jízdního pruhu. Platí pro obě varianty křižovatek.
 - Otočením libovolné křižovatky dle zadaného úhlu.
 - Vybráním předdefinovaných typů vozidel a nastavením hustoty dopravy.
 - Zachování optimalizace provozu mechanismem „Traffic Actuation“.
 - Dynamicky ovlivňovat simulaci aktivací předdefinovaných situací.
 - Systém si parametry určí automaticky, pokud nejsou zadány uživatelem.



Obrázek 4.1: Návrh architektury s rozšířením (vlastní)

4.3 Návrh rozšíření

Návrh rozšíření vychází z implementovaného návrhu současného simulačního nástroje. Přibude jeden modul a některé stávající moduly obsáhnou nové funkce. Nově přibude také správa virtuálních strojů s dopravními řadiči sX.

Vstupem programu budou zadané konfigurace sX řadičů. Pro křižovatky řízené nástrojem SUMO zůstanou ve formátu XML, nicméně při řízení dopravním řadičem sX je potřeba konfigurace ve formátu „Sitraffic smartCore project file (.c10)“. Jedná se o formát souboru, který je výstupem nástroje „Sitraffic sCore 2.1.1“. Ten slouží pro vytváření konfigurací pro dopravní řadiče sX.

Konfigurace ve formátu XML jsou zpracovávány do vnitřní reprezentace programu, ze které jsou poté vygenerovány konfigurační soubory pro nástroj SUMO a využívá se také pro další ovládání simulace pomocí rozhraní TraCI. Konfigurace ve formátu „.c10“ je po-

třeba také zpracovat do této vnitřní reprezentace. Při zpracování do vnitřní reprezentace programu je potřeba konfiguraci ve formátu „.c10“ rozbalit jako archiv, v něm se poté nachází i samotný soubor ve formátu XML, který je možné zpracovat již implementovaným způsobem. Navíc je však potřeba konfiguraci pomocí nástroje configUploader nahrát na příslušný virtuální počítač s dopravním řadičem sX nebo na reálný dopravní řadič sX.

Jednotlivé moduly programu a jejich funkce jsou:

- Parser - obdrží na vstup parametry a vstupní konfigurace řadičů ve formátu XML. Pomocí těchto informací vytvoří vnitřní reprezentaci konfigurace simulačního systému.
- Generator - z vnitřní reprezentace vygeneruje konfigurační soubory pro nástroj SUMO.
- SXcontrol - umožňuje spouštění a využití dostupných virtuálních počítačů s dopravními řadiči sX nebo reálných dopravních řadičů sX. Dále spouští služby pro řízení jednotlivých virtuálních počítačů.
- Simulator - ovládá simulaci pomocí rozhraní TraCI a zasílá informace o projetí vozidel křižovatkou příslušným službám pro řízení dopravních řadičů sX.
- TrafficSimulatorService - slouží jako spustitelný modul, přijímá HTTP požadavky od klientské aplikace a odesílá odpovědi na požadavky. Přijímá HTTP požadavky a odesílá odpovědi také na požadavky od jednotlivých spuštěných služeb pro řízení dopravních řadičů sX.

Pro rozhraní TraCI je i nadále využito knihovny TraaS jako webové služby. Pomocí ní se získávají informace o pohybu vozidel a zasílají se do webového klienta, který zobrazuje a ovládá simulaci. Navíc se ale informace o pohybu vozidel zasílají také do příslušných služeb pro řízení dopravních řadičů sX (pokud budou využívány nějakou křižovatkou). Služba sx-controller obsahuje rozhraní „C-Control“, které komunikaci s dopravními řadiči sX umožňuje. Dopravní řadiče sX na základě informací o pohybu vozidel řídí svou příslušnou křižovátku v simulaci. Pokud byla konfigurace ve formátu „.c10“, zajistí nahrání konfigurace na příslušný dostupný dopravní řadič sX. Nástroj SUMO se spouští s vygenerovanými soubory na základě požadavku z klientské aplikace.

Komunikační rozhraní

Komunikační rozhraní zůstává v principu totožné, jako doposud. Využívá se standardní HTTP protokol ve verzi 1.1 a jeho metody GET a POST. Komunikaci iniciuje klientská webová aplikace, která na základě svého stavu zasílá požadavky. Modul TrafficSimulatorService zůstane v roli serveru a tyto požadavky obsluhuje prostřednictvím třídy „HttpProxy“. Ta je zodpovědná za správné zpracování požadavku, sestavení příslušné odpovědi a správné odeslání odpovědi klientské webové aplikaci. Je však potřeba komunikaci rozšířit o možnosti týkající se dopravních řadičů sX. Je potřeba server informovat o potřebě využití dopravního řadiče sX a získat jeho konfiguraci. Vznikly tedy nové požadavky, kterým bude určeno jejich URN¹, pomocí kterého se jednoznačně identifikují všechny požadavky.

¹Uniform Resource Name

Správa virtuálních počítačů

Virtuální počítače s dopravními řadiči sX je zapotřebí evidovat. Je potřeba uchovávat seznam dostupných virtuálních počítačů a také uchovávat informace o jejich využití. V souboru „application.properties“ s externí konfigurací přibyla volba počtu dostupných virtuálních počítačů. Virtuální počítače jsou identifikované názvem, ten bude vyžadován ve formátu „SXx“, kde „x“ je celé číslo začínající číslem jedna. Číslovat je zapotřebí v posloupnosti. IP adresy pro komunikaci s dopravními řadiči sX je potřeba nakonfigurovat. Virtuální počítač SX1 má IP adresu „192.168.128.3“, která je defaultní po instalaci. Další virtuální počítače budou mít IP adresu s číslem posledního oktetu navýšeným o jedna.

Pro manipulaci s virtuálními počítači se využije nástroj VBoxManage [3], který je obsáhnut v základní instalaci nástroje Oracle VirtualBox. Ten nabízí následující:

- Zapnutí virtuálního počítače podle jeho názvu.
- Vypnutí virtuálního počítače podle jeho názvu.
- Pozastavení a opětovné uvedení do chodu virtuálního počítače podle jeho názvu.
- Zobrazení seznamu zapnutých virtuálních počítačů.

Problém při využití virtualizovaných dopravních řadičů sX je dlouhá doba jejich spuštění. Připravené k řízení křižovatky jsou až po uplynutí doby v řádech jednotek minut. Nahrání nové konfigurace může trvat podobně dlouhou dobu. Při práci s virtuálními stroji je tedy potřeba minimalizovat jejich vypínání a zapínání je pokud možno v předstihu. První virtuální počítač je zapnut při přístupu ke klientské webové aplikaci, pokud není již zapnutý. Připravení k použití je možné otestovat prostřednictvím dostupnosti pomocí IP adresy. Pokud je dopravní řadič sX připraven k použití, ale není vyžadováno jeho využití, přepne se do pozastaveného stavu, ze kterého ho lze téměř okamžitě uvést do běžícího stavu. Pozastavený virtuální stroj minimalizuje využití prostředků hardwaru počítače, na kterém běží. Při přidání křižovatky, která má být řízena virtualizovaným dopravním řadičem sX, se virtuální počítač přepne z pozastaveného stavu do běžícího stavu a je možné ho využívat. Tím se minimalizuje doba čekání. Pokud je dostupný další virtuální počítač, při využití již zapnutého virtuálního počítače, tak se spustí a po naběhnutí opět pozastaví a bude takto připraven k rychlému využití pro případné řízení další křižovatky. Pokud se z aktuální simulační situace vymaže křižovatka, která využívala virtualizovaný dopravní řadič sX, tak se příslušný virtuální počítač pouze uvede do pozastaveného stavu a je zaevidováno, že ho lze použít pro řízení jiné křižovatky.

Tyto akce s virtuálními počítači tedy nemusí nutně provádět uživatel. Může se o ně postarat simulační část aplikace pomocí nástroje VBoxManage.

Využití reálného dopravního řadiče sX

Návrh počítače také s možností využití reálných dopravních řadičů sX místo virtualizovaných. Komunikace s nimi se neliší od jejich virtualizovaných verzí. Při jejich využívání se nebudou provádět akce týkající se správy virtuálních počítačů. Přístup k nim probíhá také pomocí příslušné IP adresy (a případně portu). Ta bude potřeba zadat při přidávání křižovatky, která má reálný dopravní řadič sX využít.

Řízení simulace klientskou aplikací

Simulace se může nacházet ve třech stavech:

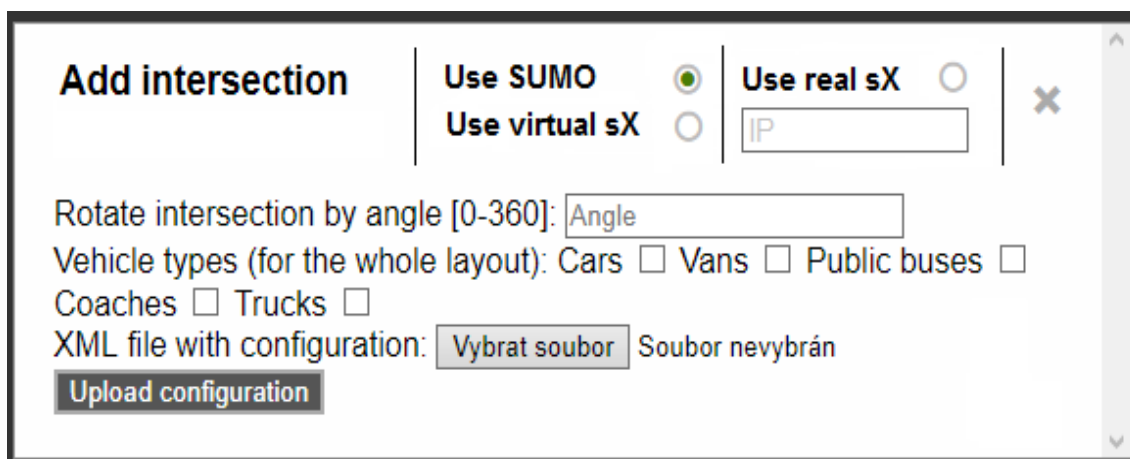
- Běžící simulace.
- Pozastavená simulace.
- Zastavená simulace.

Při pozastavené simulaci se aktivně nezískávají data a animace je pozastavena. Prvky zůstávají v posledním známém stavu. Při využití virtualizovaných dopravních řadičů sX je potřeba navíc pozastavit jejich příslušný virtuální počítač. Důvodem je, že kdyby zůstal dopravní řadič spuštěn a nebyly by mu zasílány informace o projetých vozidlech, ovlivnilo by to jeho rozhodování při úpravě řízení křižovatky. Tím by se simulace stala nepřesnou a znehodnotily by se její výsledky. Při pokračování simulace se virtuální počítač opět uvede do běžícího stavu.

Při zastavení simulace se virtuální počítače pozastaví a dopravní řadiče sX jsou označeny za volné k využití.

Úprava uživatelského rozhraní

Úpravy uživatelského rozhraní se týkají zabudování možnosti přidávat křižovatky, které bude řídit dopravní řadič sX. Na obrázku 4.2 je znázorněna úprava dialogového okna pro přidávání první křižovatky do simulace. Přibyla možnost volby mezi přidáním křižovatky řízené nástrojem SUMO, řízením virtualizovaným dopravním řadičem sX a řízením reálným dopravním řadičem sX. Pokud se vybere volba využití virtualizovaného dopravního řadiče sX, místo popisu „XML file with configuration:“ u výběru souboru bude popis „.c10 file with configuration:“. Při využití reálného dopravního řadiče sX se navíc přidalo pole pro zadání IP adresy, pomocí které se k němu přistupuje.



Obrázek 4.2: Návrh úpravy přidávání křižovatky (Snímek obrazovky)

Obdobně je provedena úprava dialogového okna pro přidávání sousedních křižovatek.

Délka simulačního kroku

Délka simulačního kroku byla určena na 0,5s na základě kompromisu mezi výkonem a přesností z následujících faktorů:

- Přesnost simulace s ohledem na vizualizaci.
- Objem přenesených dat.
- Časová náročnost generování jednoho simulačního kroku.

Dalším faktorem nyní bude využití dopravních řadičů sX. Navýší se časová náročnost generování jednoho simulačního kroku kvůli komunikaci s nimi. Zvolená hodnota 0,5s tedy nemusí vyhovovat a bude případně potřeba najít novou vyhovující hodnotu délky simulačního kroku.

Kapitola 5

Popis implementace

V této kapitole bude popsána implementace rozšíření stávajícího simulačního systému. Popsány budou také nutné změny ve stávajícím simulačním systému, které bylo potřeba provést, aby toto rozšíření bylo možné využívat. Závěrem této kapitoly bude popis komunikace mezi simulátorem a službou řízení dopravního řadiče sX.

5.1 Implementace rozšíření

Rozšíření stávajícího simulačního nástroje umožňující využití dopravních řadičů sX je rozděleno na dvě části:

- Správa virtuálních počítačů.
- Řízení dopravního řadiče sX.

Správa virtuálních počítačů se týká evidence dostupných virtuálních počítačů a jejich stavu. Také operací jejich spouštění, pozastavení a vypínání. Řízení dopravního řadiče sX poskytuje úkony týkající se dopravního řadiče, který na virtuálních počítačích běží. Jedná se zejména o přístup k informacím o světelných signálech křižovatky a simulaci detektorů křižovatky při průjezdu vozidla křižovatkou ve spuštěné simulaci. Struktura a logika řízení těchto dvou částí je popsána níže.

Správa virtuálních počítačů

Pro správu virtuálních počítačů byl vytvořen nový modul SXcontrol, který je přidán do původní struktury simulačního nástroje a je spouštěn zároveň s ním. Ze souboru externí konfigurace „application.properties“ je získán počet dostupných virtuálních počítačů a také cesta k nástroji „VBoxManage“ v souborovém systému. Dále je získán také port, na kterém simulační nástroj aktuálně běží. Tyto tři informace jsou využity jako vstupní parametry při vytváření instance třídy „SXvirtualboxManager“. Ta uchovává seznam poskytnutých virtuálních počítačů, včetně jejich stavu. Virtuální počítač se může nacházet v následujících třech stavech:

- Dostupný - jedná se o spuštěný virtuální počítač, který není využit pro řízení žádné křižovatky v simulaci.
- Využitý - virtuální počítač je spuštěný a aktuálně se využívá pro řízení křižovatky v simulaci.

- Vypnutý - vypnutý virtuální počítač, který může být spuštěn, když je potřeba. Spuštěním nabývá stavu „Dostupný“.

Dále může být virtuální počítač také pozastaven. Při pozastavení je zachován jeho stav. Stav jednotlivých virtuálních počítačů jsou kontrolovány již při spuštění. Důvodem je, že uživatel může virtuální počítače spustit explicitně, ještě před spuštěním simulačního nástroje. Při vytváření seznamu virtuálních počítačů je tedy zaznamenán jejich stav, který v danou chvíli může být pouze „Dostupný“ nebo „Vypnutý“.

Modul SXcontrol slouží také ke spuštění služby řízení dopravního řadiče sX. Při přiřazení virtuálního počítače k křižovatce jsou poté u virtuálního počítače uchovány také informace o identifikátoru křižovatky v simulaci a číslo portu, na kterém řídicí služba běží.

Třída „SXvirtualboxManager“ obsahuje také metody, které umožňují manipulaci s virtuálními počítači tak, jak bylo navrženo v části 4.3 týkající se správy virtuálních počítačů.

Řízení dopravního řadiče sX

Služba řízení dopravního řadiče sX je vytvořena jako samostatný projekt „sx-controller“. Stejně jako simulátor je implementovaná v jazyce Java verze 8 jako SpringBoot aplikace (3.2.1) ve verzi 2.0.0. Služba je vždy spuštěna pro jeden konkrétní dopravní řadič sX. Její strukturu lze rozdělit dle poskytovaných funkcí na tyto části:

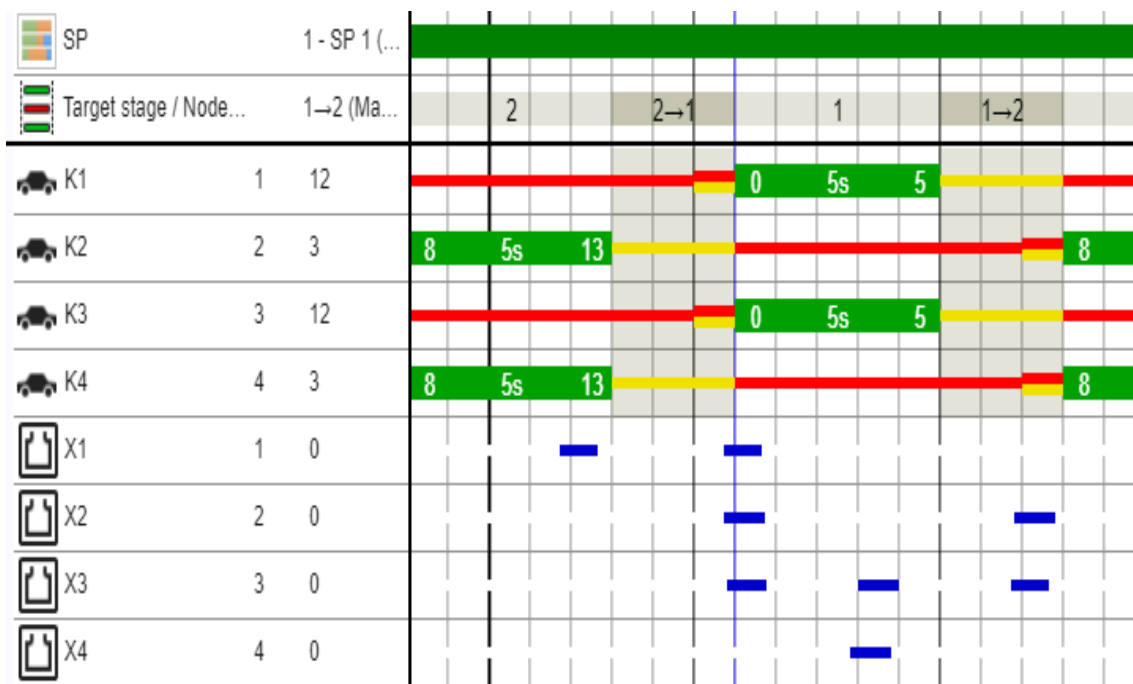
- Komunikační rozhraní - obdobné jako komunikační rozhraní simulátoru. Přijímá HTTP požadavky metody GET, na základě kterých se provádějí požadované akce.
- Rozhraní „C-Control“ - rozhraní umožňující komunikaci s dopravním řadičem sX.
- „configUploader“ - nástroj umožňující nahrání konfigurace dopravního řadiče sX.
- JMX - technologie využitá při zasílání informace o průjezdu vozidla dopravnímu řadiči sX.

Komunikační rozhraní implementované ve třídě „SxRestController“ nabízí následující metody, pomocí kterých se komunikuje s dopravním řadičem sX:

- setDetectorState - metoda spustí simulaci konkrétního detektoru umístěného v některém rameni křižovatky. Tímto způsobem je dopravnímu řadiči sX předávána informace o průjezdu vozidla. V grafickém uživatelském rozhraní dopravního řadiče sX je poté možné vidět průjezd vozidla jako modrou čáru na časové ose detektoru, jak znázorňuje obrázek 5.1. Každá modrá čára v rádcích „X1 - X4“ reprezentuje jednu detekci vozidla.
- setTrafficLightsOfSumoIntersection - při vyvolání metody se začne simulátoru posílat příkaz k nastavení světelných signálů křižovatky, ke které služba řízení přísluší. Rozhraní „C-Control“ je využíváno pomocí třídy „ControlCenter“ a umožňuje přístup k informacím poskytovaným dopravním řadičem sX. Využito je informací o tom, ve které fázi se signální program nachází a také informací o aktuálních zobrazovaných světelných signálech. Tyto informace jsou získávány při každé změně a zobrazované světelné signály jsou ukládány do seznamu, který je postupně aktualizován. Pro každou signální skupinu je tedy v seznamu uložen aktuální světelný signál. Při každé změně fáze (včetně fázového přechodu) je simulátoru odeslán příkaz pro nastavení

aktuálních světelných signálů uchovaných v tomto seznamu. Na obrázku 5.1 lze krom simulace průjezdu vozidel také vidět fáze, fázové přechody a světelné signály zobrazované signálními skupinám. Aktivní fáze nebo fázový přechod signálního plánu je zobrazena číslem v řádce „Target stage“. Zobrazovaný světelný signál dané signální skupiny zobrazují řádky „K1 - K4“.

- stopSendingTrafficLights - metoda zastaví zaslání příkazu nastavení světelných signálů dané křižovatky. Služba řízení dopravního řadiče sX uchovává informaci o tom, zdali má zasílat příkaz k nastavení světelných signálů křižovatky. Obnovení zaslání je provedeno opětovným vyvoláním metody „setTrafficLightsOfSumoIntersection“.
- uploadConfiguration - metoda vyvolá transakci nahrávání konfigurace na příslušný dopravní řadič sX pomocí nástroje configUploader. Tento úkon bohužel nelze plně automatizovat a je vyžadováno potvrzení transakce v grafickém uživatelském rozhraní dopravního řadiče sX. Jako parametr je vyžadována cesta k souboru typu „.c10“ s konfigurací.



Obrázek 5.1: Znázornění detekce vozidel v průběhu řízení křižovatky (Snímek obrazovky)

5.2 Změny ve stávajícím simulačním systému

Změny v implementaci si vyžádalo zejména komunikační rozhraní stávajícího simulačního systému. Také bylo potřeba upravit logiku řízení křižovatek a implementovaný mechanismus „Traffic Actuation“.

Změny v komunikačním rozhraní stávajícího simulačního systému

Komunikační rozhraní stávajícího simulačního systému bylo popsáno v kapitole 3.1. V této sekci jsou popsány změny v původních metodách a také nově přidané metody. Jedná se o následující metody:

- `getDataMultipleStep` - tato původní metoda pro získávání simulačních kroků pro vizualizaci simulace nyní také zruší pozastavení virtuálních počítačů ve stavu „Využitý“, které byly pozastaveny v důsledku pozastavení simulace.
- `clearSituation` - jedná se o původní metodu sloužící ke smazání všech křižovatek ze simulace. Nyní také zastavuje zaslání světelných signálů z využitých dopravních řadičů sX. Dále také nastaví všem spuštěným virtuálním počítačům stav „Dostupný“ a uvede je do pozastaveného stavu.
- `stopSimulation` - tato původní metoda sloužící k zastavení simulace bez vymazání všech křižovatek ze simulace nyní také pozastaví virtuální počítače ve stavu „Využitý“.
- `addIntersection` - tato původní metoda přidává křižovatku do simulace. Předávána je nově informace o tom, zdali se má pro řízení křižovatky využít dopravní řadič sX. Pokud ne, metoda pracuje beze změny. Pokud ano, nastaví se křižovatce příznak využívání dopravního řadiče sX. Dále se získá virtuální počítač ve stavu „Dostupný“, zruší se jeho pozastavení a přiřadí se mu řízení dané křižovatky. Křižovatce i virtuálnímu počítači se také přiřadí port, na kterém běží služba řízení dopravního řadiče sX a je změněn stav virtuálního počítače na „Využitý“. Odešle se také požadavek na nastavování světelných signálů křižovatky příslušné službě řízení dopravního řadiče sX. Pokud aktuálně neexistuje další virtuální počítač ve stavu „Dostupný“, vyvolá se metoda pro zapínání virtuálních počítačů.
- `sendConfigurationSX` - pomocí této nově zavedené metody je přijat z webového rozhraní soubor s konfigurací ve formátu „.c10“. Soubor je uložen a také je z něj extrahována konfigurace ve formátu XML. Ta je odeslána zpět webovému rozhraní. Tato akce je využívána při přidávání křižovatky řízené dopravním řadičem sX, kdy je potřeba konfigurace ve formátu „.c10“ pro dopravní řadič sX a konfigurace ve formátu XML pro simulaci.
- `startSX` - tato nově zavedená metoda slouží ke spuštění virtuálních počítačů s dopravními řadiči sX. Nejprve je zjištěno, které virtuální počítače běží. Pokud je ve stavu „Dostupný“ alespoň jeden virtuální počítač, další se nespouští. Další virtuální počítač se spustí v případě, kdy žádný není ve stavu „Dostupný“ a zároveň existuje další ve stavu „Vypnutý“.
- `getIntersectionLegs` - pomocí této nově zavedené metody získává řídicí služba dopravního řadiče sX řetězec určující aktuální povolené směry jízdy jednotlivých ramen křižovatky. Tuto informaci využívá pro vygenerování řetězce určujícího, jaké světelné signály mají být na křižovatce zobrazeny. Nejprve je získán seznam ramen příslušné křižovatky. Poté se postupně procházením tohoto seznamu sestavuje výsledný řetězec, který má tvar „<číslo ramene>_xxx-<číslo ramene>_xxx“. Délka řetězce závisí na počtu ramen a hodnota „x“ vyjadřuje hodnotou „1“ nebo „0“ povolené směry v pořadí jízda rovně, doleva, doprava. Číslo ramene je potřeba, protože není gerantováno správné pořadí vůči konfiguraci nahrané na příslušném dopravním řadiči sX.

- `setTrafficLights` - tato nově zavedená metoda umožňuje nastavovat zobrazené světelné signály příslušné křižovatky. Využívá se při řízení křižovatky dopravním řadičem `sX`. Parametry metody jsou číslo portu, na kterém běží příslušná řídicí služba a řetězec určující, jaké světelné signály mají být zobrazeny. Dle zadaného čísla portu se identifikuje příslušná křižovatka, u které se má změna provést.

Změny kvůli řízení dopravním řadičem `sX`

Úpravu komunikačního rozhraní stávajícího simulačního systému provází také úprava jeho dalších částí. V implementaci rozhraní „TraCI“ přibyla metoda „`setTrafficLightState`“ sloužící k nastavení světelných signálů křižovatky v běžící simulaci. Upraven byl také implementovaný mechanismus „Traffic Actuation“. Detekce projíždějícího vozidla zůstává obsluhována, pokud je ale křižovatka řízena dopravním řadičem `sX`, mechanismus „Traffic Actuation“ se neprovádí. Místo toho se pouze spustí simulace detektoru na příslušném dopravním řadiči `sX`. Ten na základě informací o průjezdu vozidel může reagovat změnou v programu řízení křižovatky, což se projeví na aktuálně zobrazovaných světelných signálech, které se následně v simulaci nastavují.

Změny na straně webového klienta

Změny byly nutné provést také na straně webového klienta. Zejména díky změně možností přidávání křižovatky do simulace. Upraveno bylo dialogové okno pro přidávání křižovatky. Přibyl výběr způsobu řízení křižovatky a podle toho se dynamicky mění prvky v dialogovém okně, zejména kvůli změny volby souboru s konfigurací. Při výběru řízení křižovatky nástrojem SUMO se prováděné akce nemění, pouze přibyla v požadavku odesílaném simulátoru při potvrzení tlačítka „Upload configuration“ informace o tom, že se nemá k řízení křižovatky využít dopravního řadiče `sX`. Při výběru řízení křižovatky dopravním řadičem `sX` se nejprve odešle požadavek s konfigurací ve formátu „.c10“ simulátoru. Ten v odpovědi na tento požadavek vrátí konfiguraci ve formátu XML. Ta je dále poskytnuta původní logice obsluhy konfigurace ve formátu XML s informací, že se dopravní řadič `sX` k řízení křižovatky využije. Ukázky změn webového klienta lze nalézt v příloze [B](#).

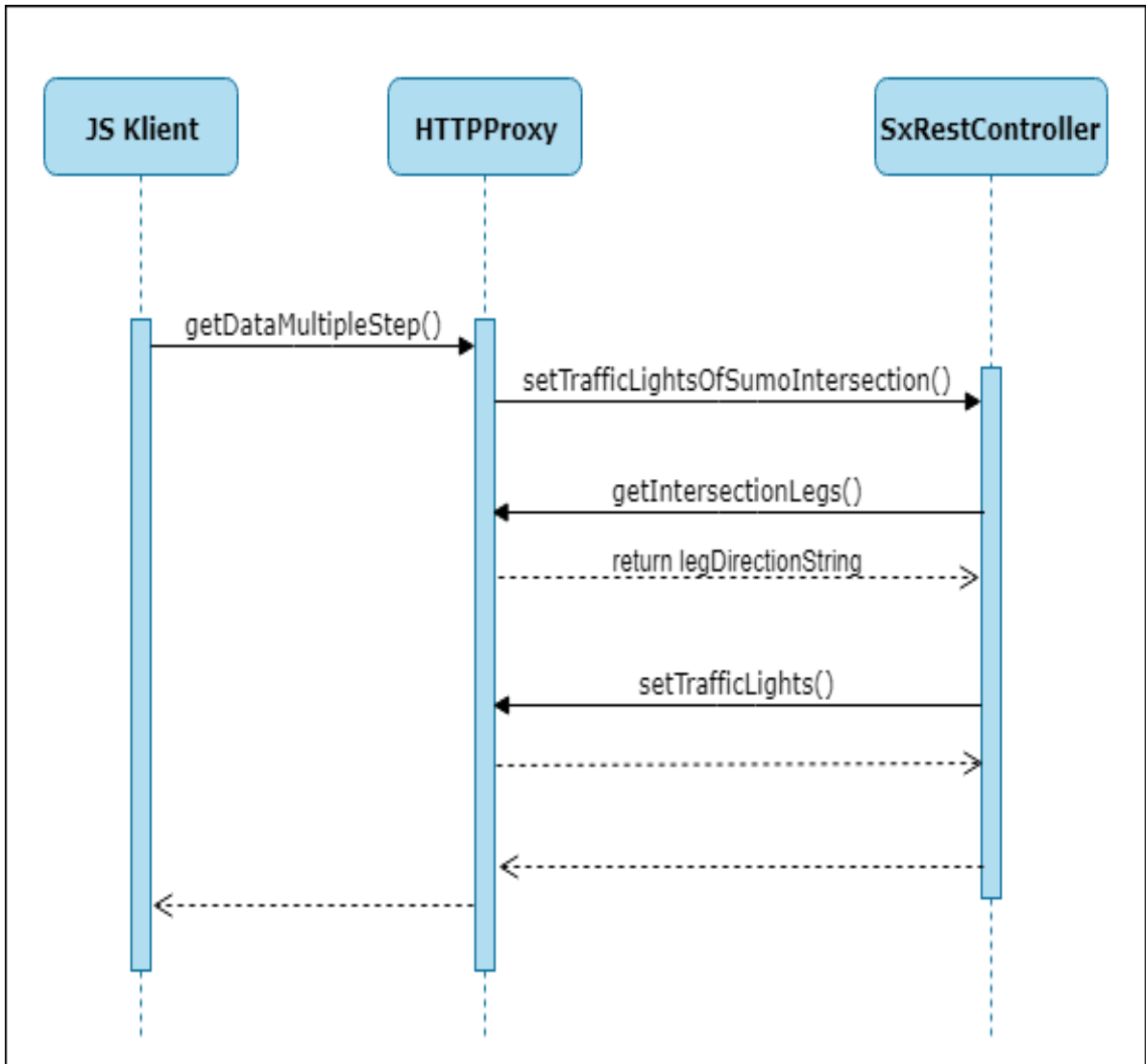
5.3 Komunikace mezi simulátorem a dopravním řadičem `sX`

V této části bude popsána komunikace mezi simulačním systémem a službou řízení dopravního řadiče `sX` pomocí popsaných metod jejich komunikačních rozhraní.

Komunikace při nastavení světelných signálů křižovatek v simulaci

Komunikaci při nastavování světelných signálů iniciuje simulační systém prostřednictvím modulu „`TrafficSimulatorService`“, při přidávání křižovatky do simulace, odesláním požadavku „`setTrafficLightsOfSumoIntersection`“ příslušné řídicí službě dopravního řadiče `sX`. Ta při obdržení tohoto požadavku nejprve odešle simulátoru požadavek „`getIntersectionLegs`“ sloužící pro získání řetězce s povolenými směry jízdy z jednotlivých ramen křižovatky. Důvodem je, že lze v simulaci omezit povolené směry jízdy z jednotlivých dopravních pruhů, směry pak nekorespondují s nahranou konfigurací dopravního řadiče `sX`, je proto nutné získat aktuální povolené směry, aby byl řetězec sestaven korektně. Tento řetězec je následně odeslán jako odpověď na požadavek „`getIntersectionLegs`“ řídicí službě dopravního řadiče

sX. Ta řetězec zpracuje na výsledný řetězec určující, jaké světelné signály mají být na křižovatce zobrazeny. Ten je následně odeslán simulátoru v rámci požadavku „setTrafficLights“, který světelné signály křižovatky nastaví. Komunikace po odeslání požadavku „setTrafficLightsOfSumoIntersection“ probíhá při každé změně světelných signálů ke zobrazení. Komunikaci znázorňuje také sekvenční diagram 5.2.



Obrázek 5.2: Sekvenční diagram nastavování světelných signálů křižovatky (vlastní)

Komunikace při předávání informace o průjezdu vozidel

Komunikaci při simulaci detektoru, za účelem předávání informace o průjezdu vozidel, iniciuje simulační systém prostřednictvím modulu „Simulator“. Ten při obsluze detekovaného vozidla na křižovatce řízené dopravním řadičem sX nejprve převede identifikátor detektoru v simulaci na identifikátor detektoru v dopravním řadiči sX. Poté je příslušné službě řízení dopravního řadiče sX odeslán požadavek „setDetectorState“ s parametrem identifikátoru

detektoru v dopravním řadiči sX. Služba řízení dopravního řadiče sX poté spustí samotnou simulaci příslušného detektoru. Ta je prováděna pomocí technologie JMX¹ a příslušných metod, které po připojení na dopravní řadič sX nabízí vytvořená MBean² pro manipulaci s detektory.

Komunikace při nahrávání konfigurace dopravního řadiče sX

Komunikaci při nahrávání konfigurace dopravního řadiče sX iniciuje simulační systém prostřednictvím modulu „TrafficSimulatorService“. Při přidávání křižovatky do simulace se odesílá požadavek „uploadConfiguration“ příslušné řídicí službě dopravního řadiče sX. Parametrem tohoto požadavku je cesta k souboru v adresářové struktuře systému. Soubor s konfigurací ve formátu „.c10“ je uložen do adresáře „SXconfigFiles“ umístěného v modulu SXcontrol. Uložení tohoto souboru probíhá při přidávání křižovatky řízené dopravním řadičem sX do simulace. Pro samotné spuštění transakce pro nahrávání konfigurace na dopravní řadič sX se spustí nástroj configUploader. Ten jako parametr přijímá cestu k souboru s konfigurací, ip adresu příslušného dopravního řadiče a přihlašovací jméno s heslem. Konfigurace je na dopravní řadič sX nahraná po potvrzení transakce v grafickém uživatelském rozhraní dopravního řadiče sX.

5.4 Rozdíly při komunikaci s reálným dopravním řadičem sX

Komunikace s reálným dopravním řadičem sX je možná stejným způsobem, jako s řadičem virtuálním. Pro jeho využití je ale potřeba správně nastavit síťový přístup a být v příslušné síti společnosti Siemens, s.r.o. Dále také povolit přístup pomocí JMX technologie, ta se při práci s reálným dopravním řadičem externě v reálném provozu nevyužívá. Poté fungují všechny potřebné části týkající se změny zobrazovaných světelných signálů křižovatky, zaslání informace o průjezdu vozidel a nahrávání konfigurace.

Při využití reálného dopravního řadiče pro řízení křižovatky v simulaci je ale implementovaný jiný způsob přidávání křižovatky. Při přidávání křižovatky ve webovém rozhraní je při volbě „Use real sX“ očekáváno zadání IP adresy, na které je reálný dopravní řadič dostupný. Dále jsou také zadávány přístupové údaje. Při potvrzení tlačítkem „Upload configuration“ je v rámci sestavování požadavku pro simulátor předávána informace o využití reálného dopravního řadiče. Dále také informace o IP adrese a přístupových údajích.

Při provádění metody „addIntersection“ na straně simulátoru se pro přidání křižovatky poté neprovádí akce související s virtuálními počítači. Příslušné vytvořené křižovatce je ale nastaven příznak využití reálného řadiče. Při spouštění řídicí služby dopravního řadiče sX jsou poté údaje o IP adrese a přístupových údajích předány. Příslušná řídicí služba poté tyto údaje uchovává a využívá je pro komunikaci s reálným dopravním řadičem sX. Komunikace simulátoru s řídicí službou poté probíhá pomocí stejných metod, nezávisle na tom, jestli je dopravní řadič sX reálný nebo virtuální.

¹<https://docs.oracle.com/javase/tutorial/jmx/overview/index.html> [Online; navštíveno 19.5.2019]

²<https://docs.oracle.com/javase/tutorial/jmx/mbeans/index.html> [Online; navštíveno 19.5.2019]

5.5 Problémy při práci se simulačním systémem

Snahou při implementaci rozšíření stávajícího simulačního systému bylo také optimalizovat hledisko použitelnosti výsledné úpravy simulačního nástroje. Výsledkem je automatizování činností spjatých se správou virtuálních počítačů a nahrávání konfigurace dopravního řadiče sX. Určitým omezením se však vyhnout nepodařilo.

Při práci s virtuálními počítači, na kterých běží dopravní řadiče sX, existuje problém s dobou jejich uvedení do spuštěného stavu. Ta se pohybuje v rámci několika minut. Tato skutečnost bohužel ovlivnit nelze. Řešením je spustit virtuální počítače a nechat je plně naběhnout ještě před využíváním simulačního nástroje. Ten je poté při svém spuštění detekuje jako dostupné a manipuluje s nimi poté tak, jak již bylo popsáno v části 5.1. Je to také důvod, proč po ukončení simulačního systému jsou virtuální počítače uvedeny do pozastaveného stavu a nejsou vypínány. Uživatel poté může sám rozhodnout, jestli je bude chtít vypnout.

Doba uvedení virtuálního počítače do spuštěného stavu byla otestována na různých zařízeních. Experimentovalo se také z možnostmi zefektivnění přiřazením větší kapacity zdrojů virtuálnímu počítači. Výsledkem však nebylo dosaženo výraznějšího rozdílu. Rozdíl se pohyboval pouze v řádech několika sekund.

Dalším problémem je také specifikované síťové rozhraní virtuálních počítačů. Nezávisle na simulačním systému musí být zajištěno, že virtuální počítače jsou dostupné na správných ip adresách, jak bylo popsáno v části 4.3. Také je potřeba zajistit jejich správné pojmenování.

Při nahrávání konfigurace je problémem nutnost potvrzení transakce v grafickém uživatelském rozhraní dopravního řadiče sX. V některých případech je poté také vyžadován restart virtuálního počítače. Ideálním stavem je, pokud na dopravním řadiči sX je již nahraná konfigurace, která má být využita při řízení křižovatky v simulaci.

Všechny zmíněné problémy výsledného simulačního nástroje se týkají přípravné fáze simulace. Na samotný běh simulace tedy nemají vliv a tím pádem nedochází ke zkreslování výsledků simulace.

Kapitola 6

Testování simulačního systému

Testováním bude ověřeno správné fungování simulačního systému a také budou zdokumentovány jeho výkonnostní limity. Zaměřím se hlavně na nové prvky, které byly přidány implementovaným rozšířením původního simulačního systému. Aby bylo možné využívat nové funkcionality, které implementované rozšíření nabízí, bylo potřeba zásahu do původních částí simulačního systému. Bude tedy potřeba otestovat také prvky původní aplikace. Celkový průběh testování bude popsán v této kapitole. Samotné testování bylo prováděno v simulovaném prostředí s využitím vytvořených konfigurací dopravních řadičů sX. Testování simulačního systému dělím na tři hlavní kategorie:

- Testování původních funkcí simulačního systému - rozšířený simulační systém zachovává možnost využití původního způsobu řízení křižovatek v simulaci. Je tedy potřeba otestovat korektní chování po provedených úpravách. Tímto ověřením se zabývám v této části.
- Testování rozšíření simulačního systému - tuto část věnuji testování funkcí samotného rozšíření simulačního systému. Budou zde ověřeny veškeré přidané prvky v kombinaci s původními za účelem ověření jejich správného fungování vůči návrhu simulačního systému.
- Výkonnostní testování - v této části se zabývám testováním výkonu simulačního systému po rozšíření jeho funkcionalit. Proběhne také porovnání výkonu původního systému oproti systému využívajícímu rozšíření.

Testovací sada konfigurací dopravních řadičů sX

Po konzultaci a dohodě s vedoucí práce bylo rozhodnuto, že se pro testování využije stejná testovací sada konfigurací dopravních řadičů sX, jaká byla využita při testování původního simulačního systému. Důvodem je možnost objektivnějšího následného porovnání výsledků testování s výsledky testování původního simulačního systému. Tyto konfigurace také již pokrývají dostupné možnosti nastavení řízení křižovatek. Parametry využitých konfigurací jsou shrnuty v tabulce 6.1. Rozdíly mezi typy signálních plánů byly podrobně rozebrány v kapitole 1.

ID	Typ signálního plánu	Využití detektorů	Počet ramen	Úhly ramen
1	Dynamický	Ano	2	0, 45
2	Pevný	-	2	0, 90
3	Dynamický	Ne	2	0, 135
4	Pevný	-	2	0, 180
5	Dynamický	Ano	3	0, 45, 90
6	Pevný	-	3	0, 45, 135
7	Dynamický	Ne	3	0, 45, 180
8	Pevný	-	3	0, 90, 180
9	Dynamický	Ano	3	0, 90, 225
10	Pevný	-	4	0, 45, 90, 135
11	Dynamický	Ne	4	0, 45, 90, 180
12	Pevný	-	4	0, 45, 90, 270
13	Dynamický	Ano	4	0, 90, 135, 180
14	Pevný	-	4	0, 90, 180, 270
15	Dynamický	Ne	4	0, 45, 135, 315

Tabulka 6.1: Parametry testovací sady konfigurací

6.1 Testování původních funkcí simulačního systému

Testování původních funkcí simulačního systému bylo provedeno ve dvou fázích. První fáze se zabývá tzv. „Black-box“ testováním. Funkčnost simulačního systému je v této fázi ověřována nezávisle na jeho vnitřní implementaci. Druhá fáze se týká tzv. „White-box“ testování. V této fázi se již ověřovalo vnitřní fungování simulačního systému.

Black-box testování původních funkcí

Původní funkce simulačního systému byly otestovány z pohledu uživatele. Účelem bylo otestovat korektní reakce simulačního systému na ovládací prvky webového rozhraní po provedených úpravách. Využito bylo již vytvořených testovacích případů v rámci diplomové práce „Simulátor dopravních infrastruktur a situací. [11]“ dostupných v její příloze C. Modifikací těchto testovacích případů je ponechání volby „Use SUMO“ při přidávání křižovatek.

Testováním pomocí těchto testovacích případů byly nalezeny následující dvě chyby v původních funkcích simulačního systému:

1. Chyba úhlu křižovatky při přidávání křižovatky do simulační situace - chyba byla nalezena při využití testovacího případu #14. Úhel křižovatky musí být v rozpětí 0° - 359° . Testovací konfigurace 5 obsahuje zadaný úhel 360° . Tato hraniční hodnota byla povolena a při jejím zadání je upravena na 0° .
2. Chyba propojení některých křižovatek - chyba byla nalezena při využití testovacího případu #15. Jedna z křižovatek zůstala nepropojena na středovou křižovatku, i když ji bylo opticky možné napojit. Tato chyba opravena nebyla z důvodu nutnosti velkého zásahu do logiky propojování křižovatek. Chyba nastává pouze při určitých úhlech ramen křižovatek.

White-box testování původních funkcí

V této části byly testovány části systému z hlediska jejich vnitřního fungování. Korektnost vizualizace simulace byla testována přímým porovnáním s vizualizací, kterou nabízí program SUMO. Využito bylo již vytvořených testovacích případů v rámci diplomové práce „Simulátor dopravních infrastruktur a situací. [11]“ dostupných v její příloze D.

Testování pomocí těchto testovacích případů, při využití libovolně rozsáhlých dopravních situací, neodhalilo žádné nové chyby. Ověření se týkalo zejména korektního chování vozidel a zobrazovaných světelných signálů křižovatek. Chybou již zdokumentovanou v rámci diplomové práce „Simulátor dopravních infrastruktur a situací. [11]“ zůstává problém s centry některých křižovatek. Ty se svou velikostí rozcházejí s centry křižovatek ve vestavěné vizualizaci programu SUMO. V důsledku se v některých případech vozidla při průjezdu křižovatkou nacházejí mimo její prostor.

Shrnutí výsledků testování původních funkcí

Testováním původních funkcí byla ověřena funkčnost simulačního systému při využití řízení křižovatek nástrojem SUMO. Simulační systém lze tedy korektně využívat původním způsobem a není nutné využívat možnosti poskytnuté implementovaným rozšířením.

6.2 Testování rozšíření simulačního systému

Testování rozšíření simulačního systému bylo provedeno také ve dvou fázích. Pro fázi „Black-box“ testování byly vytvořeny nové testovací případy pro přidávání křižovatek. Ve fázi „White-box“ testování byla ověřena korektnost řízení křižovatky dopravním řadičem sX. Testování probíhalo pouze s konfiguracemi obsahujícími detektory. Pouze v těchto případech nabízí využití dopravních řadičů sX přidanou hodnotu. Bez detektorů by řízení křižovatky probíhalo vždy shodně jako při řízení křižovatky nástrojem SUMO.

Black-box testování rozšíření simulačního systému

Funkce spadající pod rozšíření simulačního systému byly také otestovány z pohledu uživatele. Účelem bylo otestovat korektní fungování simulačního systému při využívání prvků rozšíření. Simulační systém také nabízí možnost kombinovat způsoby řízení jednotlivých křižovatek v jedné simulaci.

Vytvořil jsem tedy nové testovací případy pro „Black-box“ testování. Ty pokrývají nové možnosti přidávání křižovatek do simulace. Konkrétně přidání křižovatky řízené dopravním řadičem sX a křižovatky řízené reálným dopravním řadičem sX. Tyto vytvořené testovací případy jsou dostupné v příloze C této práce. Příkladem je testovací případ uvedený v tabulce 6.2. Dále byly také znovu využity již vytvořené testovací případy v rámci diplomové práce „Simulátor dopravních infrastruktur a situací. [11]“ dostupné v její příloze C, krom případů #1 a #3 týkajících se přidávání křižovatek. Tentokrát byly ale využity na křižovatky řízené dopravními řadiči sX.

Z pohledu uživatelského rozhraní se s přidanými křižovatkami pracuje stejným způsobem, nezávisle na způsobu jejich řízení. Bylo ale potřeba ověřit, že se veškeré akce provádějí korektně pro všechny typy řízení křižovatek. Testování proběhlo také na dopravních situacích, kde byly přidané křižovatky s různými typy jejich řízení zároveň. Při tomto testování se objevila chyba při zobrazování světelných signálů křižovatky řízené dopravním řadičem sX. V některých případech se v křižovatce nezměnily zobrazované světelné signály. Tento

problém se objevuje při vyšší náročnosti spuštěné simulace a souvisí s vyššími požadavky na výkon po provedených úpravách. Tomuto problému se věnuji v části „Výkonnostní testování“ v kapitole 6.3.

ID: 001	Přidání první křižovatky řízené dopravním řadičem sX
Popis:	Simulační situace je prázdná a lze přidat první křižovatku.
Vstupní podmínky:	1. Simulační systém je spuštěný a webové rozhraní je na něj korektně připojeno. 2. Simulační situace je prázdná. 3. Existuje virtuální počítač ve stavu „Dostupný“ nebo ve stavu „Vypnutý“.
Prováděné akce:	1. Ve webovém rozhraní klikneme na tlačítko „Add first intersection“. 2. Ve zobrazeném formuláři přepneme řízení křižovatky na volbu „Use virtual sX“. 3. Ve zobrazeném formuláři necháme pole „Angle“ prázdné, zaškrtneme políčko „Cars“ a nahrajeme testovací konfiguraci číslo 4 ve formátu souboru „.c10“.
Očekávaný výstup:	1. V simulační situaci je přidána první křižovatka s ID 0. 2. Tlačítko „Add first intersection“ je skryté.

Tabulka 6.2: Testovací případ 001

White-box testování rozšíření simulačního systému

Tato část je věnována ověření vnitřní logiky týkající se řízení křižovatek dopravními řadiči sX. Potřeba je otestovat korektní funkčnost následujících dvou částí logiky implementovaného rozšíření simulačního systému:

- Správa virtuálních počítačů - ta se týká pouze případů, kdy není využitý reálný dopravní řadič sX.
- Řízení křižovatky dopravním řadičem sX.

Správa virtuálních počítačů byla otestována z hlediska manipulace s virtuálními počítači. Jedná se o jejich detekování při startu simulačního systému, jejich spouštění a pozastavování dle využití příslušných akcí a také správné uchovávání jejich stavů. Ověřování uchovávání stavů probíhalo pomocí kontrolních výpisů stavů v terminálu se spuštěným simulátorem. Spouštění, pozastavení a detekování virtuálních počítačů bylo ověřováno v programu VirtualBox. Pro toto testování jsem vytvořil nové testovací případy s ID 104-107 dostupné v příloze D této práce. Příkladem je testovací případ uvedený v tabulce 6.3.

Pomocí těchto testovacích případů byly odhaleny následující tři chyby, které byly následně opraveny:

1. Smazání křižovatky řízené dopravním řadičem sX při běžící simulaci nezastaví zaslání požadavku na změnu světelných signálů křižovatky a nezmění se stav virtuálního počítače na stav „Dostupný“.
2. Kliknutím na tlačítko „Get simulation output“ se pozastavuje simulace, neproběhne ale pozastavení virtuálních počítačů ve stavu „Využitý“.
3. Pozastavení běžící simulace nepozastaví virtuální počítače ve stavu „Využitý“.

ID: 104	Ověření korektního načtení stavů virtuálních počítačů při připojení na klientskou aplikaci.
Popis:	Při připojení na klientskou aplikaci do prázdné situace proběhne zjišťování dostupnosti virtuálních počítačů.
Vstupní podmínky:	1. Simulační systém je spuštěný. Jsou povoleny kontrolní výpisy do terminálu.
Prováděné akce:	1. Připojíme webové rozhraní.
Očekávaný výstup:	1. Ve výpisu stavů virtuálních počítačů je u všech spuštěných virtuálních počítačů uveden stav „Dostupný“. U všech vypnutých virtuálních počítačů je uveden stav „Vypnutý“.
	2. Pokud nebyl žádný virtuální počítač ve stavu „Dostupný“. Proběhne spuštění virtuálního počítače. Ve výpisu stavů virtuálních počítačů je u prvního uveden stav „Dostupný“.
	3. Ověříme vůči spuštěným a vypnutým virtuálním počítačům v programu VirtualBox.

Tabulka 6.3: Testovací případ 104

Řízení křižovatky dopravním řadičem sX bylo testováno vůči zobrazovaným informacím uživatelského rozhraní dopravního řadiče sX. Ověřováno bylo korektní zobrazování světelných signálů křižovatky v simulaci, korektní zaslání informace o průjezdu vozidel dopravnímu řadiči sX a také nahrávání konfigurace na dopravní řadič sX. Pro účel tohoto ověření jsem vytvořil nové testovací případy s ID 101-103 dostupné v příloze D této práce. Příkladem je testovací případ uvedený v tabulce 6.4.

Tyto testovací případy odhalily chybu v simulaci při zobrazování světelných signálů křižovatky řízené dopravním řadičem sX, kdy se v některých případech v křižovatce nezměnily zobrazované světelné signály. Stejná chyba byla odhalena při „Black-box testování“ rozšíření simulačního systému. Tentokrát ale bylo možné ověřit, že se světelné signály křižovatky korektně mění na straně dopravního řadiče sX. Při povolení kontrolního výpisu v terminálu s běžícím simulátorem bylo také potvrzeno, že požadavky na změnu světelných signálů křižovatky byly korektně přijaty. Příčina chyby souvisí s požadavky na výkon systému a je rozebrána v části „Výkonnostní testování“, kapitola 6.3.

ID: 101	Verifikace světelných signálů křižovatek
Popis:	Kontrola zobrazovaných světelných signálů křižovatky vůči GUI příslušného dopravního řadiče sX.
Vstupní podmínky:	1. Simulační systém je spuštěný a webové rozhraní je na něj korektně připojeno. 2. Simulační situace obsahuje křižovatku řízenou dopravním řadičem sX na pozici [0,0] při využití konfigurace 13. 3. Příslušná služba dopravního řadiče sX je korektně spuštěná.
Prováděné akce:	1. V GUI dopravního řadiče sX otevřeme záložku „Topology“. 2. Ve webovém rozhraní klikneme na tlačítko „Start simulation“. 3. Manuálně zkontrolujeme, jestli jsou zobrazené světelné signály křižovatek totožné.
Očekávaný výstup:	1. Všechny světelné signalizační zařízení křižovatky mění své zobrazované světelné signály zároveň s dopravním řadičem sX. 2. Simulace je stále ve spuštěném stavu.

Tabulka 6.4: Testovací případ 101

6.3 Výkonnostní testování

Výkonnostní testování obecně slouží k ověření, jestli systém vyhovuje vůči specifikovaným požadavkům na výkon. Dále se také systém testuje zatížením nad rámec specifikovaných výkonnostních požadavků až do selhání systému. Poté se analyzuje, jak a proč systém selhal [9].

Při testování výkonu simulačního systému budu sledovat zejména vliv náročnosti simulace na akce prováděné simulátorem a také jejich vliv na vizualizaci simulace na straně webového klienta. Oproti původnímu simulačnímu systému přibyla komunikace s dopravními řadiči sX. Akce spjaté s touto komunikací mají vliv na celkový výkon simulačního systému. Faktorem s největším vlivem na výkon ale zůstává počet vozidel nacházejících se v simulaci. Tento faktor ovlivňuje dobu výpočtu simulačního kroku a má vliv i na samotnou vizualizaci. Navyšuje také potřebné množství komunikace s dopravními řadiči sX, kterým se zasílá informace o průjezdu vozidel příslušnou křižovatkou.

Samotné testování probíhalo na notebooku Acer Aspire 5750G s nainstalovaným operačním systémem Windows 8.1 Pro 64bit. Hardware komponenty využitého notebooku jsou:

- Procesor¹ - Intel(R) Core(TM) i5-2430M CPU @ 2.40GHz (2 jádra, 4 logické procesory) Intel Virtualization Technology (VT-x), Intel VT-x with Extended Page Tables.
- Paměť - 2x Kingston 4GB 1333MHz RAM DDR3.
- Disk² - 750GB WDC WD7500BPVT 5400RPM 8Mb cache SATA 3Gb/s.

¹http://ark.intel.com/products/53450/Intel-Core-i5-2430M-Processor-3M-Cache-up-to-3_00-GHz
[Online; navštíveno 19.5.2019]

²<http://www.wdc.com/wdproducts/library/SpecSheet/ENG/2879-701278.pdf>
[Online; navštíveno 19.5.2019]

Porovnání výkonu při různém způsobu řízení křižovatek

Cílem tohoto porovnání je zjistit, jaký vliv na výkon má využití dopravního řadiče sX pro řízení křižovatky v simulaci. Měřenou veličinou byla doba zpracování jednoho simulačního kroku. Pro testování bylo využito konfigurace číslo 13. Nejprve jsem do simulační situace přidal křižovatku řízenou nástrojem SUMO. Poté byla spuštěna simulace. Při dosažení maximálního počtu vozidel, které je nastavené na 50, byla zaznamenána délka zpracování každého desátého simulačního kroku. Pro vypočítání průměrné délky zpracování bylo využito deset zaznamenaných údajů. Poté jsem stejným způsobem otestoval křižovatku řízenou dopravním řadičem sX. Na závěr tohoto testu ještě byla otestována simulační situace, kde byly přidány dvě křižovatky. Jedna řízená nástrojem SUMO a druhá řízená dopravním řadičem sX. Měření délky zpracování jednoho simulačního kroku probíhalo od začátku příslušné metody do jejího ukončení. Testování bylo poté opakováno, aby se zajistila větší přesnost výsledků. Výsledek měření shrnuje následující tabulka 6.5:

Typ řízení křižovatky	Průměrná délka zpracování simulačního kroku	% vůči SUMO řízení
Nástroj SUMO	114,4ms	100%
Řadič sX	223,5ms	195%
SUMO zároveň s sX	340,3ms	297%

Tabulka 6.5: Shrnutí výstupu testování délky zpracování simulačního kroku

Dle údajů z tabulky lze vypožorovat, že doba zpracování jednoho simulačního kroku při řízení křižovatky dopravním řadičem sX je skoro dvojnásobně náročnější, než při řízení křižovatky nástrojem SUMO. Pokud se v dopravní situaci nachází zároveň dvě křižovatky, jedna řízená nástrojem SUMO a druhá dopravním řadičem sX, náročnost zpracování jednoho simulačního kroku odpovídá přibližně součtu jejich jednotlivých délek zpracování.

Při řízení křižovatky dopravním řadičem sX je ale část simulačního kroku zpracována příslušnou řídicí službou. Doba získávání světelných signálů křižovatky dosahuje průměrné hodnoty 19ms. Doba zasílání informace o průjezdu vozidla dopravnímu řadiči sX je v průměru 11ms pro každé vozidlo. Tyto hodnoty jsou již započítány v naměřené celkové hodnotě délky zpracování.

Rozbor chyby zobrazování světelných signálů křižovatky

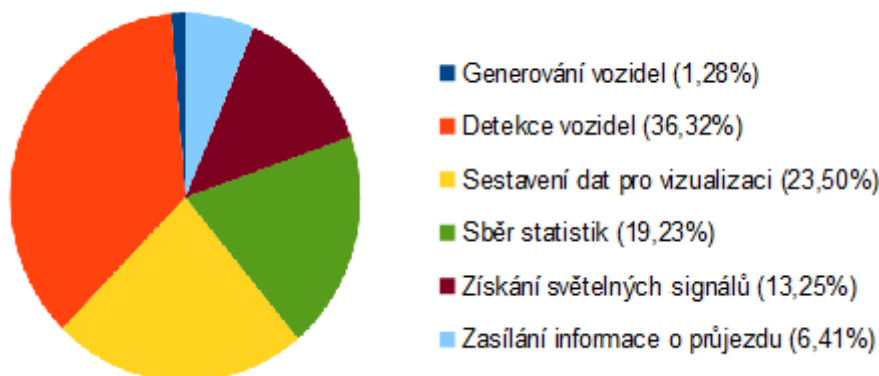
Tato část se týká chyby, která byla zmíněna při testování funkčnosti rozšířeného simulačního systému. Konkrétně při zobrazování světelných signálů křižovatky se v některých případech zobrazované signály nezměnily. Pomocí výkonnostního testování byla odhalena příčina této chyby. Získávání světelných signálů průměrně trvá 19ms, v některých případech se ale tato hodnota navýšila průměrně na 578ms. V tu chvíli dochází k uvedené chybě. Tento problém souvisí přímo s komunikací mezi řídicí službou a dopravním řadičem sX, kdy se získávají změny zobrazovaných světelných signálů. Příčina tohoto občasného navýšení doby trvání nalezena nebyla. Nicméně dalším testováním bylo zjištěno, že při navýšení délky simulačního kroku z 0,5s na 0,7s se chyba neobjevovala tak často. Zvyšování délky simulačního kroku nad hranici 0,7s již zlepšení nepřineslo a mělo pouze negativní vliv na plynulost vykreslované simulace.

Rozbor náročnosti jednotlivých úkonů simulačního kroku

Cílem této části je analýza doby zpracování jednoho simulačního kroku. To může v budoucnu sloužit pro optimalizaci simulačního systému. Zpracování jednoho simulačního kroku při využití řízení dopravním řadičem sX se skládá z více částí. Zjištěna byla časová náročnost zpracování každé z nich. Jedná se o následující úkony:

- Generování vozidel v simulaci.
- Detekce vozidel v simulaci.
- Sestavení dat k odeslání webovému klientovi pro vizualizaci simulace.
- Sběr statistických dat pro výstup simulace.
- Získávání světelných signálů křižovatky z dopravního řadiče sX.
- Zasílání informace o průjezdu vozidla dopravnímu řadiči sX.

Při testování byla opět využita konfigurace číslo 13. Testováno bylo stejným způsobem, jako v části „Porovnání výkonu při různém způsobu řízení křižovatek“. Měření délky zpracování ale probíhalo po jednotlivých úkonech. Testování bylo opakováno, aby se zajistila větší přesnost výsledků. Ty shrnuje následující graf 6.1:



Obrázek 6.1: Procentuální složení jednotlivých úkonů simulačního kroku (Vlastní)

Generování vozidel v simulaci tvoří nejmenší část z hlediska časové náročnosti. Vozidla nejsou generována neustále a také je nastavený limit jejich maximálního počtu v simulaci. Detekce vozidel v simulaci je ale nejnáročnější částí. Vozidlo se detekuje pomocí TraCI rozhraní a počtem vozidel narůstá náročnost zpracování. Sběr statistických dat a sestavení dat pro vizualizaci také ovlivňuje počet vozidel a v důsledku tedy patří k náročnějším částem. Získávání světelných signálů a zasílání informace o průjezdu vozidla křižovatkou jsou méně náročné než předchozí části.

Analýza výkonu vizualizace

Cílem této části bylo zjistit, zdali způsob řízení křižovatky v simulaci ovlivňuje dobu vykreslování vizualizace simulace. Testování proběhlo na stejných dopravních situacích, jako v části „Porovnání výkonu při různém způsobu řízení křižovatek“ za využití konfigurace číslo 13. Průměrné délky zpracování simulačního kroku vizualizace se při různém způsobu řízení křižovatek lišily pouze v rámci 1ms. Taková hodnota lze považovat za pouhou odchylku při testování. Tento výsledek není překvapující, protože se zásahy do struktury této části webového rozhraní neprováděly.

6.4 Shrnutí výstupů testování

V této kapitole je popsáno testování rozšířeného simulačního systému. Otestován byl nejprve z hlediska funkčnosti. Funkční testování odhalilo celkem šest chyb. Čtyři chyby byly méně závažného charakteru a byla také provedena jejich oprava. Chyba propojení křižovatek při některých úhlech jejich ramen opravena nebyla z důvodu potřeby velkého zásahu do logiky propojování křižovatek. Poslední chyba souvisí se získáváním světelných signálů křižovatky z dopravního řadiče sX. Tato chyba byla odhalena již při funkčním testování, její příčina ale nesouvisí s funkční chybou. Tato chyba byla rozebrána v další části testování týkající se testování výkonu simulačního systému. Základní veličinou při testování výkonu simulačního systému byla doba zpracování jednoho simulačního kroku. Porovnáním této doby byla zjištěna vyšší náročnost na zpracování simulačního kroku při využití řízení křižovatky dopravním řadičem sX. I přes zvýšenou náročnost výpočtu, křižovatky v simulaci řízené dopravním řadičem sX neovlivnily plynulost vizualizace simulace. Avšak navýšením délky simulačního kroku z 0,5s na 0,7s byla omezena vyskytující se chyba při zobrazování světelných signálů. Ta způsobuje, že v některých případech neproběhne změna zobrazovaných světelných signálů příslušné křižovatky řízené dopravním řadičem sX v simulaci.

Pro testování funkcí simulačního systému bylo využito metody „Black-box“ i „White-box“ testování. Systém byl takto ověřen z pohledu uživatele i z pohledu vnitřního fungování. Využito bylo již vytvořených testovacích případů v rámci diplomové práce „Simulátor dopravních infrastruktur a situací. [11]“ dostupných v její příloze C a D. Vytvořeny byly také nové testovací případy pro testování prvků rozšíření včetně kombinace s původními prvky simulačního systému. Dostupné jsou v příloze C a D této práce.

Kapitola 7

Závěr

Tato diplomová práce navazuje na semestrální projekt. Cílem této práce bylo nejprve nastudovat obecnou problematiku řízení dopravních křižovatek a konfiguraci dopravních řadičů, které dané dopravní křižovatky řídí. Zejména poté konfiguraci dopravních řadičů sX společnosti Siemens, s.r.o. Potřeba bylo také nastudovat principy dynamického řízení křižovatek. Tyto principy umožňují optimalizaci řízení křižovatky za účelem zefektivnění dopravy. Dále také možnosti doplňujících konfigurací programu pro návrh křižovatek, které umožňují návrh sítě křižovatek. Poté bylo cílem seznámit se se stávajícím implementovaným simulačním systémem a navrhnout jeho rozšíření tak, aby bylo možné využívat pro řízení křižovatky v simulaci logiku dopravního řadiče sX. V rámci dokončení diplomové práce bylo poté potřeba vytvořený návrh rozšířené simulačního systému implementovat. Následně také provést testování implementovaného simulačního systému po rozšíření jeho funkcí a ověřit tím jeho použitelnost. Na závěr bylo také potřeba zhodnotit dosažené výsledky a diskutovat další možnosti budoucího rozšíření simulačního systému.

Nejprve jsem se tedy seznámil s problematikou řízení dopravních křižovatek a s principy dynamického řízení křižovatky. Také s vytvářením konfigurací řadičů a s dopňkovými konfiguracemi. Tyto poznatky mně pomohly pochopit problematiku řízení dopravních křižovatek a získat obecný přehled potřebný pro vypracování této práce.

Po seznámení se s problematikou bylo potřeba, abych plně pochopil funkčnost stávajícího simulačního systému. Potřeboval jsem pochopit, jak byl navrhnout a implementován, abych poté mohl navrhnout jeho rozšíření.

Abych mohl navrhnout požadované rozšíření, musel jsem se nejprve seznámit s možnostmi využívání dopravního řadiče sX. Za tímto účelem byly domluveny každý týden pravidelné konzultace s mým konzultantem ve společnosti Siemens, s.r.o. Postupně bylo potřeba odstraňovat problémy, na které jsem narazil. Nakonec se mně podařilo vyřešit všechny. Jednalo se například o nahrávání konfigurace dopravního řadiče sX jinak, než přes jeho webové rozhraní a zajištění korektního běhu více virtualizovaných dopravních řadičů sX zároveň. Samozřejmě bylo také potřeba zajistit komunikaci s dopravním řadičem sX. Zejména za účelem získávání informací o zobrazovaných světelných signálech křižovatky a také zasílání informace o průjezdu vozidla křižovatkou dopravnímu řadiči sX. Následoval poté samotný návrh rozšíření stávajícího simulačního systému, který zohledňoval možný způsob práce s dopravními řadiči sX.

Po vytvoření návrhu následovala samotná implementace navrženého rozšíření. Ta byla také postupně konzultována. Vytvořena byla řídicí služba dopravního řadiče sX, která slouží pro získávání údajů o zobrazovaných světelných signálech křižovatky, zasílání informace o průjezdu vozidla v simulaci a nahrávání konfigurace. Poté bylo potřeba upravit také části

původního simulačního systému, aby byla umožněna komunikace s touto řídicí službou. Důsledkem těchto změn je poskytnutá možnost přímého řízení křižovatky v simulaci dopravním řadičem sX. Díky tomu je možné simulovat řízení křižovatky pomocí reálné komponenty, která v praxi řídí provoz skutečných křižovatek.

Důležitou částí bylo také testování simulačního systému. Nejprve probíhalo testování funkčnosti, poté výkonnostní testování. Funkční testování odhalilo několik chyb, většina byla mírnějšího charakteru a byla opravena. Výkonnostní testování odhalilo zvýšenou náročnost výpočtu při řízení křižovatky dopravním řadičem sX. Veškeré výsledky byly zdokumentovány a můžou posloužit při námětu na budoucí úpravy implementovaného simulačního systému.

Výsledkem této práce je tedy možnost přímého řízení křižovatky v simulaci dopravním řadičem sX. Díky tomu výsledky simulace více odpovídají reálným případům. Při dalším vývoji v budoucnu by tento systém mohl pomáhat při návrhu reálných křižovatek a návrhu jejich konfigurace řízení.

7.1 Námět na budoucí úpravy simulačního systému

Možné budoucí rozšíření se může týkat webového rozhraní i samotného simulátoru. Webové rozhraní v rámci této práce bylo měněno pouze mírně, zejména s ohledem na potřebu zabudování možnosti přidávání křižovatek s různým způsobem jejich řízení. Z dopravního řadiče je možné získávat i další „online“ informace průběhu řízení, ty by mohly být také zobrazovány v rámci webového rozhraní při zobrazení detailu křižovatky.

Simulace by také mohla podporovat chodce a cyklisty, ty je potřeba v reálných dopravních situacích brát v úvahu. Aktuálně generování vozidel nezohledňuje hlavní dopravní tahy, tato úprava by také simulaci přiblížila více reálným případům. Úpravy by se také mohly týkat zejména optimalizace výkonu. Při snížených nárocích na výkon by bylo možné povolit vyšší limit počtu vozidel zároveň přítomných v simulaci. Dopravní řadiče sX jsou nadále vyvíjeny. Je tedy možné, že přibudou další možnosti, které by poté mohly být využity při simulaci.

Literatura

- [1] *SUMO User Documentation*. [Online; navštíveno 10.12.2018].
URL http://sumo.dlr.de/wiki/SUMO_User_Documentation
- [2] *TraaS - TraCI as a Service*. [Online; navštíveno 12.12.2018].
URL <http://traas.sourceforge.net/cms/>
- [3] *Chapter 9. VBoxManage*. [Online; navštíveno 30.11.2018].
URL <https://www.virtualbox.org/manual/ch09.html>
- [4] *TP 81. Navrhování světelných signalizačních zařízení pro řízení provozu na pozemních komunikacích*. Praha: Ministerstvo dopravy č. j. 122/2015-120-TN/2, 2015, [Online; navštíveno 1.10.2018].
URL <http://www.pjpk.cz/viewFile.asp?file=1542>
- [5] Siemens, s.r.o.: Siemens traffic detectors overview. Siemens AG 2014, dostupné na příloženém CD.
- [6] Siemens, s.r.o.:
ITS_INFO_2018-010_Annex_A_Sittraffic_sX_3.0.1_Featureset_sx_-_C9N,
Siemens AG 2018, dostupné na příloženém CD.
- [7] Siemens, s.r.o.: ITS_INFO_2018-010_sX_advanced_V301_EN, Siemens AG 2018,
dostupné na příloženém CD.
- [8] Barceló, J.: *Fundamentals of Traffic Simulation*. International Series in Operations Research & Management Science, Springer New York, 2011, ISBN 9781441961426.
- [9] Farrell-Vinay, P.: *Manage Software Testing*. Auerbach Publications, 2008, ISBN 0-8493-9383-3, 978-0-8493-9383-9.
- [10] Fielding, R.; Gettys, J.; Mogul, J.; aj.: RFC 2616, Hypertext Transfer Protocol – HTTP/1.1. June 1999.
- [11] Švaňa, P.: *Simulátor dopravních infrastruktur a situací*. Diplomová práce, Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. RNDr. Jitka Kreslíková, CSc., Brno 2018.

Příloha A

Obsah přiloženého CD

Přiložené CD obsahuje tyto adresáře:

- bibliography/ - soubory ve formátu PDF odkazované z využití literatury.
- tex-src/ - zdrojové soubory textové části diplomové práce.
- thesis/ - diplomová práce ve formátu PDF.
- client - zdrojové soubory webového rozhraní simulačního systému.
- simulator - zdrojové a binární soubory simulačního systému.
- README.txt - návod ke spuštění simulačního systému.

Příloha B

Ukázky úpravy webového rozhraní

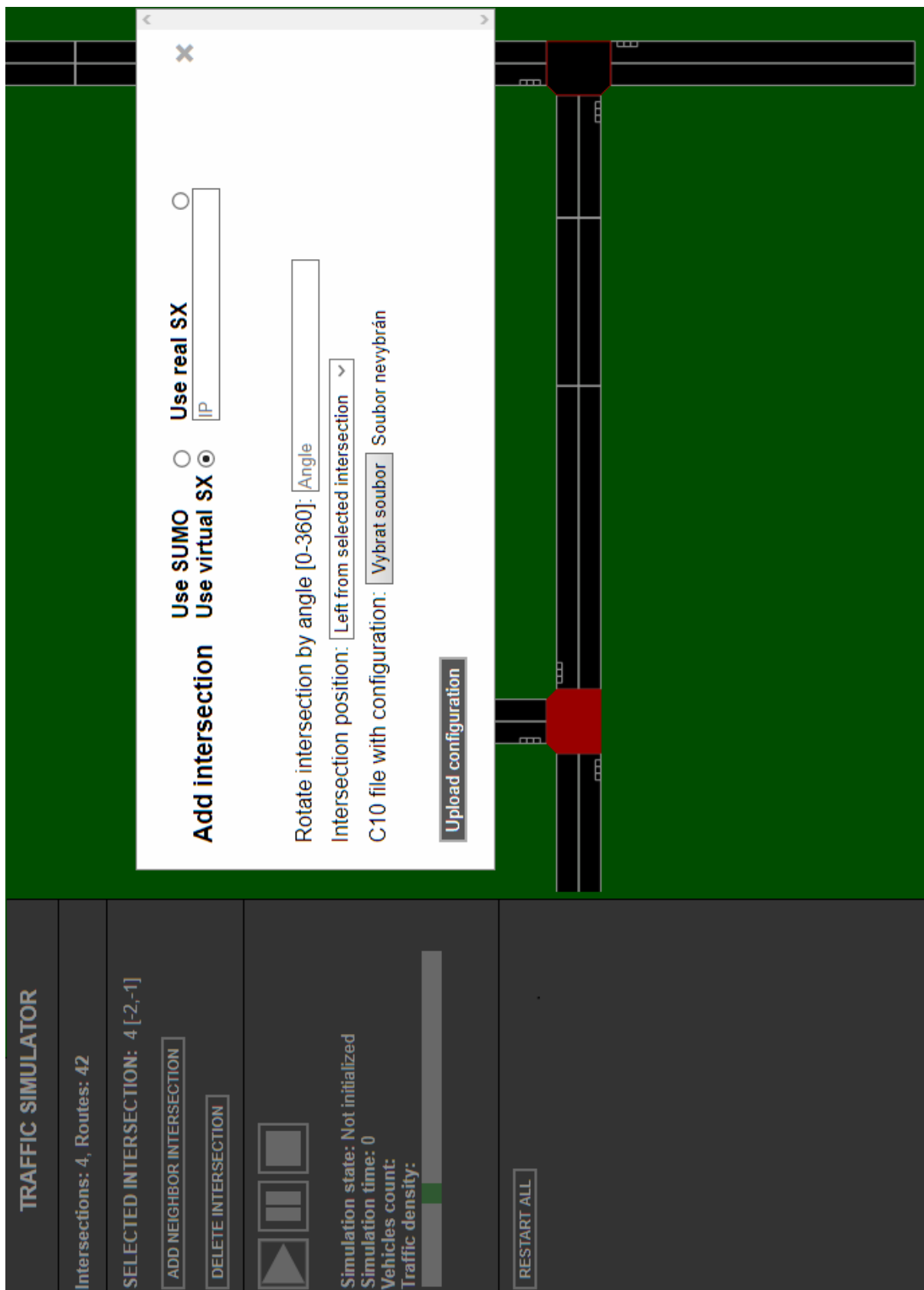
The image displays three sequential screenshots of a web application interface for adding a new intersection. Each screenshot is contained within a separate window with a title bar and a close button.

Top Screenshot: The window title is "Add intersection". It features two radio buttons: "Use SUMO" (selected) and "Use real SX" (unselected). Below them are "Use virtual SX" (unselected) and an "IP" input field. The main text reads "Rotate intersection by angle [0-360]:" followed by an "Angle" input field. Below that, "Vehicle types (for the whole layout):" includes checkboxes for "Cars" (checked), "Vans", "Public buses", "Coaches", and "Trucks". The configuration file is listed as "XML file with configuration:" with a "Vybrat soubor" button and the filename "hlc_base.xml". An "Upload configuration" button is at the bottom.

Middle Screenshot: The window title is "Add intersection". The radio buttons are "Use SUMO" (unselected) and "Use real SX" (selected). "Use virtual SX" is selected. The "IP" input field is present. The main text is "Rotate intersection by angle [0-360]:" with an "Angle" input field. "Vehicle types" are the same as in the top screenshot. The configuration file is "C10 file with configuration:" with a "Vybrat soubor" button and the filename "4-0-90-135...O-DET.c10". An "Upload configuration" button is at the bottom.

Bottom Screenshot: The window title is "Add intersection". Both "Use SUMO" and "Use real SX" are unselected. "Use virtual SX" is unselected. The "IP" input field is present. Below this, there are "User:" and "Password:" labels with corresponding input fields. The main text is "Rotate intersection by angle [0-360]:" with an "Angle" input field. "Vehicle types" are the same as in the top screenshot. The configuration file is "C10 file with configuration:" with a "Vybrat soubor" button and the filename "4-0-90-135...O-DET.c10". An "Upload configuration" button is at the bottom.

Obrázek B.1: Ukázka všech možností přidání křižovatky (Snímek obrazovky)



Obrázek B.2: Ukázka přidání další křižovatky do existující situace (Snímek obrazovky)

Příloha C

Testovací případy pro Black-box testování

ID: 001	Přidání první křižovatky řízené dopravním řadičem sX
Popis:	Simulační situace je prázdná a lze přidat první křižovatku.
Vstupní podmínky:	1. Simulační systém je spuštěný a webové rozhraní je na něj korektně připojeno. 2. Simulační situace je prázdná. 3. Existuje virtuální počítač ve stavu „Dostupný“ nebo ve stavu „Vypnutý“.
Prováděné akce:	1. Ve webovém rozhraní klikneme na tlačítko „Add first intersection“. 2. Ve zobrazeném formuláři přepneme řízení křižovatky na volbu „Use virtual sX“. 3. Ve zobrazeném formuláři necháme pole „Angle“ prázdné, zaškrtneme políčko „Cars“ a nahrajeme testovací konfiguraci číslo 4 ve formátu souboru „.c10“.
Očekávaný výstup:	1. V simulační situaci je přidána první křižovatka s ID 0. 2. Tlačítko „Add first intersection“ je skryté.

Tabulka C.1: Testovací případ 001

ID: 002	Přidání další křižovatky řízené dopravním řadičem sX
Popis:	Simulační situace není prázdná, obsahuje libovolný počet křižovatek. Lze přidat další křižovatku řízenou dopravním řadičem sX.
Vstupní podmínky:	1. Simulační systém je spuštěný a webové rozhraní je na něj korektně připojeno.
	2. Existuje virtuální počítač ve stavu „Dostupný“ nebo ve stavu „Vypnutý“.
Prováděné akce:	1. Ve webovém rozhraní vybereme libovolnou křižovatku.
	2. Ve webovém rozhraní klikneme na tlačítko „Add neighbour intersection“.
	3. Ve zobrazeném formuláři přepneme řízení křižovatky na volbu „Use virtual sX“.
	4. Ve zobrazeném formuláři necháme pole „Angle“ prázdné, a nahrajeme testovací konfiguraci číslo 10 ve formátu souboru „.c10“.
Očekávaný výstup:	1. V simulační situaci je přidána další křižovatka.

Tabulka C.2: Testovací případ 002

ID: 003	Přidání první křižovatky řízené reálným dopravním řadičem sX
Popis:	Simulační situace je prázdná a lze přidat první křižovatku.
Vstupní podmínky:	1. Simulační systém je spuštěný a webové rozhraní je na něj korektně připojeno.
	2. Simulační situace je prázdná.
Prováděné akce:	1. Ve webovém rozhraní klikneme na tlačítko „Add first intersection“.
	2. Ve zobrazeném formuláři přepneme řízení křižovatky na volbu „Use real sX“.
	3. Zadáme IP adresu, na které je reálný dopravní řadič sX dostupný. Případně také port ve formátu IP:PORT.
	4. Zadáme přihlašovací jméno a heslo, pokud je to potřeba.
	5. Ve zobrazeném formuláři necháme pole „Angle“ prázdné, zaškrtneme políčko „Cars“ a nahrajeme testovací konfiguraci číslo 6 ve formátu souboru „.c10“.
Očekávaný výstup:	1. V simulační situaci je přidána první křižovatka s ID 0.
	2. Tlačítko „Add first intersection“ je skryté.

Tabulka C.3: Testovací případ 003

ID: 004	Přidání další křižovatky řízené reálným dopravním řadičem sX
Popis:	Simulační situace není prázdná, obsahuje libovolný počet křižovatek. Lze přidat další křižovatku řízenou reálným dopravním řadičem sX.
Vstupní podmínky:	1. Simulační systém je spuštěný a webové rozhraní je na něj korektně připojeno.
Prováděné akce:	1. Ve webovém rozhraní vybereme libovolnou křižovatku.
	2. Ve webovém rozhraní klikneme na tlačítko „Add neighbour intersection“.
	3. Ve zobrazeném formuláři přepneme řízení křižovatky na volbu „Use real sX“.
	4. Zadáme IP adresu, na které je reálný dopravní řadič sX dostupný. Případně také port ve formátu IP:PORT.
	5. Zadáme přihlašovací jméno a heslo, pokud je to potřeba.
	6. Ve zobrazeném formuláři necháme pole „Angle“ prázdné, a nahrajeme testovací konfiguraci číslo 8 ve formátu souboru „.c10“.
Očekávaný výstup:	1. V simulační situaci je přidána další křižovatka.

Tabulka C.4: Testovací případ 004

Příloha D

Testovací případy pro White-box testování

ID: 101	Verifikace světelných signálů křižovatek
Popis:	Kontrola zobrazovaných světelných signálů křižovatky vůči GUI příslušného dopravního řadiče sX.
Vstupní podmínky:	1. Simulační systém je spuštěný a webové rozhraní je na něj korektně připojeno. 2. Simulační situace obsahuje křižovatku řízenou dopravním řadičem sX na pozici [0,0] při využití konfigurace 13. 3. Příslušná služba dopravního řadiče sX je korektně spuštěná.
Prováděné akce:	1. V GUI dopravního řadiče sX otevřeme záložku „Topology“. 2. Ve webovém rozhraní klikneme na tlačítko „Start simulation“. 3. Manuálně zkontrolujeme, jestli jsou zobrazené světelné signály křižovatek totožné.
Očekávaný výstup:	1. Všechny světelné signalizační zařízení křižovatky mění své zobrazované světelné signály zároveň s dopravním řadičem sX. 2. Simulace je stále ve spuštěném stavu.

Tabulka D.1: Testovací případ 101

ID: 102	Verifikace zasílání informace o průjezdu vozidel
Popis:	Kontrola korektního předávání informace o průjezdu vozidel příslušnému dopravnímu řadiči sX.
Vstupní podmínky:	1. Simulační systém je spuštěný a webové rozhraní je na něj korektně připojeno. 2. Simulační situace obsahuje křižovatku řízenou dopravním řadičem sX na pozici [0,0] při využití konfigurace 13. 3. Příslušná služba dopravního řadiče sX je korektně spuštěná.
Prováděné akce:	1. V GUI dopravního řadiče sX otevřeme záložku „Visu STP“. 2. Ve webovém rozhraní klikneme na tlačítko „Start simulation“. 3. Manuálně zkontrolujeme, jestli při průjezdu vozidla v simulaci je na příslušném detektoru křižovatky v GUI dopravního řadiče sX signalizován průjezd vozidla pomocí modré čáry.
Očekávaný výstup:	1. Každý průjezd vozidla v simulaci vyvolá simulaci průjezdu vozidla na příslušném detektoru dopravního řadiče sX. 2. Simulace je stále ve spuštěném stavu.

Tabulka D.2: Testovací případ 102

ID: 103	Verifikace nahrávání konfigurace
Popis:	Kontrola korektního vyvolání transakce pro nahrání konfigurace na příslušný dopravní řadič sX.
Vstupní podmínky:	1. Simulační systém je spuštěný a webové rozhraní je na něj korektně připojeno. 2. Příslušná služba dopravního řadiče sX je korektně spuštěná.
Prováděné akce:	1. Provádíme akce dle testovacího případu 001. 2. V GUI příslušného dopravního řadiče sX je spuštěna transakce pro nahrávání konfigurace.
Očekávaný výstup:	1. Po potvrzení transakce je do simulační situace přidána křižovatka s ID 0. 2. Tlačítko „Add first intersection“ je skryté.

Tabulka D.3: Testovací případ 103

ID: 104	Ověření korektního načtení stavů virtuálních počítačů při připojení na klientskou aplikaci.
Popis:	Při připojení na klientskou aplikaci do prázdné situace proběhne zjišťování dostupnosti virtuálních počítačů.
Vstupní podmínky:	1. Simulační systém je spuštěný. Jsou povoleny kontrolní výpisy do terminálu.
Prováděné akce:	1. Připojíme webové rozhraní.
Očekávaný výstup:	1. Ve výpisu stavů virtuálních počítačů je u všech spuštěných virtuálních počítačů uveden stav „Dostupný“. U všech vypnutých virtuálních počítačů je uveden stav „Vypnutý“.
	2. Pokud nebyl žádný virtuální počítač ve stavu „Dostupný“. Proběhne spuštění virtuálního počítače. Ve výpisu stavů virtuálních počítačů je u prvního uveden stav „Dostupný“.
	3. Ověříme vůči spuštěným a vypnutým virtuálním počítačům v programu VirtualBox.

Tabulka D.4: Testovací případ 104

ID: 105	Ověření korektní změny stavu virtuálního počítače při přidání křižovatky řízené dopravním řadičem sX do simulační situace.
Popis:	Při přidání křižovatky řízené dopravním řadičem sX je virtuálnímu počítači změněn stav na „Využitý“.
Vstupní podmínky:	1. Simulační systém je spuštěný a webové rozhraní je na něj korektně připojeno. Jsou povoleny kontrolní výpisy do terminálu.
Prováděné akce:	1. Provedeme akce podle testovacího případu 001.
Očekávaný výstup:	1. Ve výpisu stavů virtuálních počítačů je u příslušného virtuálního počítače uveden stav „Využitý“.
	2. Pokud neexistuje další virtuální počítač ve stavu „Dostupný“ a existuje další virtuální počítač ve stavu „Vypnutý“, ve výpisu bude u dalšího virtuálního počítače uveden stav „Dostupný“. V programu VirtualBox poté ověříme spuštění dalšího virtuálního počítače.

Tabulka D.5: Testovací případ 105

ID: 106	Ověření korektního pozastavení virtuálních počítačů při příslušných akcích.
Popis:	Akce týkající se pozastavení a zastavení simulace, a také změn simulační situace nebo získání výstupu simulace vyvolají pozastavení všech virtuálních počítačů ve stavu „Využitý“.
Vstupní podmínky:	1. Simulační systém je spuštěný a webové rozhraní je na něj korektně připojeno. Jsou povoleny kontrolní výpisy do terminálu.
	2. Simulace, kde existuje křižovatka s využitím dopravního řadiče sX pro její řízení, je spuštěná.
Prováděné akce:	Provedeme postupně tyto akce na běžící simulaci:
	1. Zastavení simulace.
	2. Pozastavení simulace.
	3. Klikneme na tlačítko „Get simulation output“.
	4. Libovolná změna dopravní situace při běžící simulaci.
	5. Klikneme na tlačítko „Restart all“
Očekávaný výstup:	1. Všechny výše uvedené akce pozastaví všechny virtuální počítače ve stavu „Využitý“.
	2. Ověříme vůči pozastaveným virtuálním počítačům v programu VirtualBox.
	3. Volba „Restart all“ navíc uvede počítače ve stavu „Využitý“ do stavu „Dostupný“.
	4. Ověříme ve výpisu stavů virtuálních počítačů.

Tabulka D.6: Testovací případ 106

ID: 107	Ověření korektního opětovného uvedení virtuálních počítačů do spuštěného stavu.
Popis:	Při opětovném spuštění pozastavené nebo zastavené simulace se pozastavené virtuální počítače ve stavu „Využitý“ uvedou do spuštěného stavu.
Vstupní podmínky:	1. Simulační systém je spuštěný a webové rozhraní je na něj korektně připojeno.
	2. Simulace, kde existuje křižovatka s využitím dopravního řadiče sX pro její řízení, je pozastavená nebo zastavená.
Prováděné akce:	1. Opětovně spustíme simulaci.
Očekávaný výstup:	1. Všechny virtuální počítače ve stavu „Využitý“ budou ve spuštěném stavu.
	2. Ověříme vůči spuštěným virtuálním počítačům v programu VirtualBox.

Tabulka D.7: Testovací případ 107