

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

## DOPRAVNÍ SIMULAČNÍ MODELY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PAVEL ŠVÉDA

BRNO 2008



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV INTELIGENTNÍCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

## **DOPRAVNÍ SIMULAČNÍ MODELY**

TRAFFIC SIMULATION MODELS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**PAVEL ŠVÉDA**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. LUKÁŠ GRULICH**

BRNO 2008

## Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav inteligentních systémů

Akademický rok 2007/2008

### Zadání bakalářské práce

Řešitel: **Švéda Pavel**

Obor: Informační technologie

Téma: **Dopravní simulační modely**

Kategorie: Modelování a simulace

Pokyny:

1. Prostudujte problematiku dopravních modelů (konkrétně modelů pozemních komunikací). Vyberte vhodnou problematiku k demonstraci.
2. Navrhněte jednoduché systémy, které budou modelovat vybrané téma (kopce, křižovatky, kruhové objezdy, atd.). Vyberte vhodné prostředí pro implementaci.
3. Implementujte model, testujte jej. Prezentujte výsledky experimentů.
4. Diskutujte výsledky, možná rozšíření.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- První dva body zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Gulich Lukáš, Ing.**, UITS FIT VUT

Datum zadání: 1. listopadu 2007

Datum odevzdání: 14. května 2008

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav informačních systémů  
602 00 Brno, Božetěchova 2

---

doc. Dr. Ing. Petr Hanáček  
vedoucí ústavu

## **Abstrakt**

Tato práce se zabývá simulačními modely dopravy na pozemních komunikacích a jejich implementací. Dopravní simulace jsou rozděleny podle svých vlastností na jednotlivé typy. Dále je popsán způsob, jak vytvořit model chování vozidla pomocí Car-Following, Line-Changing a Gap-Acceptance modelu. V druhé části je detailní návrh a následná implementace městského dopravního simulátoru s využitím mikro simulace.

## **Klíčová slova**

dopravní simulace, dopravní modely, systémy řízení dopravy, modely chování vozidel

## **Abstract**

This thesis deals with traffic simulation models and their implementation. Traffic simulations are sorted into types by characteristic attributes. It shows how to model car behaviour by using Car-Following, Line-Changing and Gap-Acceptance models. In the second part the analysis and implementation of urban traffic simulator are described.

## **Keywords**

traffic simulation, traffic models, traffic-flow control systems, car behaviour models

## **Citace**

Pavel Švéda: Dopravní simulační modely, bakalářská práce, Brno, FIT VUT v Brně, 2008

# Dopravní simulační modely

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Lukáše Grulichy. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Pavel Švéda  
14.5.2008

## Poděkování

Na tomto místě bych rád poděkoval Ing. Lukáši Grulichovi za hodnotné rady a odborné vedení mé práce. Dále pak bratrovi Martinovi za konzultace situace na našich silnicích a jeho ženě Daniele za pomoc při stylizaci textu.

© Pavel Švéda, 2008.

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

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Simulace</b>	<b>4</b>
2.1	Simulace obecně	4
2.2	Počítačová simulace	4
2.3	Tvorba a ověření modelu	5
2.4	Využití simulace v praxi	5
2.5	Shrnutí	6
<b>3</b>	<b>Dopravní simulační modely</b>	<b>7</b>
3.1	Vymezení pojmů	7
3.2	Typy dopravních simulací	7
3.2.1	Makro simulace	7
3.2.2	Mikro simulace	7
3.2.3	Kombinované simulace	8
3.3	Řídicí modely	8
3.3.1	Car-Following model	8
3.3.2	Line-Changing model	8
3.3.3	Gap-Acceptance model	9
<b>4</b>	<b>Městský dopravní simulátor</b>	<b>11</b>
4.1	Specifikace	11
4.2	Analýza	11
4.2.1	Dopravní síť	11
4.2.2	Segment dopravní sítě	12
4.2.3	Buňka dopravního segmentu	12
4.2.4	Vozidlo	12
4.2.5	Generátor vozidel	13
4.2.6	Pohyb vozidel	13
4.2.7	Režimy	13
4.2.8	Uvážnutí	16
4.3	Implementace	18
4.3.1	Vozidla	18
4.3.2	Dopravní segmenty	19
4.3.3	Načtení dopravní sítě	22
4.3.4	Parametry simulátoru	24
4.3.5	Výstupy simulátoru	24
4.4	Experiment	25

4.5	Budoucí vývoj	27
4.5.1	Semaforey	27
4.5.2	Dopravní segmenty	28
4.5.3	Monitorování kolizí	28
4.5.4	Městská hromadná doprava	28
4.5.5	Grafická nastavba	28
<b>5</b>	<b>Závěr</b>	<b>29</b>
<b>A</b>	<b>Uživatelská příručka</b>	<b>31</b>
<b>B</b>	<b>Instalace</b>	<b>37</b>
B.1	Linux	37
B.2	Windows	37
<b>C</b>	<b>Obsah přiloženého CD</b>	<b>38</b>

# Kapitola 1

## Úvod

Automobilová doprava je oblast, která se v posledních letech dostává stále více do popředí zájmů. Neustále se rozvíjí a ovlivňuje každého z nás. S rostoucí intenzitou silniční dopravy však přibývá také problémů, které je třeba řešit.

Dopravní zácpy už dávno nejsou jen problémem velkoměst a světových metropolí, ale stále častěji se s nimi setkávají i obyvatelé a řidiči malých měst. Mají obrovský ekonomický a sociální dopad. Vozidla v kolonách spotřebují více paliva, spediční firmy nemohou dovézt zboží včas a dlouhé stání v kolonách negativně působí na psychiku řidičů, což může vést k dopravním nehodám. Dlouhé stání vozidel v kolonách ale také dramaticky ovlivňuje kvalitu ovzduší a doplácují tak na to ostatní lidé. Například v Pekingu je situace tak vážná, že v době konání olympijských her v roce 2008 chtějí organizátoři zakázat jezdit několika milionům automobilů a omezit průmyslovou výrobu ve městě. Dle některých studií totiž hrozí riziko, že sportovci budou mít zdravotní problémy [10].

Vlády světových velmocí zpřísňují zákony a předpisy, automobilky vynakládají obrovské sumy peněz na vývoj nových ekologických motorů, které by byly šetrnější. Toto však může snížit pouze dopad automobilové dopravy na životní prostředí.

Řešení je třeba hledat v prevenci, tedy zajistit aby k dopravním zácpám nedocházelo. Násilné omezení dopravy, jako v případě Pekingu, je řešením pouze dočasným a krátkodobým. Naopak možnosti alternativní dopravy a osvěta mezi obyvateli může mít v dlouhodobém měřítku úspěch.

Jedním ze způsobů jak předcházet dopravním zácpám je ale také studium a pochopení chování řidičů na silnicích a vývoj nových technologií v oblasti dopravy. K tomu nám poslouží dopravní simulace. S její pomocí můžeme detailně analyzovat dopravu ve městech, na základě sledování předvídat vývoj dopravní situace a problémům předcházet.



# Kapitola 2

## Simulace

### 2.1 Simulace obecně

Pojem simulace obecně vyjadřuje napodobování objektů a procesů. Samotný proces simulace může být chápán jako snaha o zachycení a napodobení klíčových vlastností nebo chování vybraných reálných nebo imaginárních systémů. [12]

V každodenním životě je tento pojem užíván jako synonymum pro přetvářku a je chápán spíše negativně. Za simulanta, tedy osobu simulující, je označován např. fotbalista, který předstírá faul a domáhá se potrestání protihráče, nebo zaměstnanec, který předstírá chorobu, aby nemusel do práce.

Simulace se však nemusí týkat jen světa lidí. Jako příklad nám můžou sloužit holubi, kteří si na obranu před predátorem lehnou na záda nohama vzhůru a předstírají, že jsou mrtví. Doufají, že orel nebo jiný lovec pohrdne zdechlou kořistí a potravu bude hledat jinde.

### 2.2 Počítačová simulace

*Počítačová simulace* (někdy také *simulace na číslicových počítačích* nebo *vědecká simulace*) je metoda získávání nových znalostí o systému experimentováním s jeho modelem. [6] Ty se pak mohou využít pro odhad chování systému za jiných situací (při zadání jiných parametrů).

V textu je uvedeno několik pojmů, které je třeba si blíže vysvětlit:

- *Systém* můžeme obecně definovat jako soubor elementárních částí (prvků systému), které mají mezi sebou určité vazby (propojení prvků). [6]
- *Model* je speciální případ systému, který má homomorfní vztah (vztah N:1) k originálnímu vzorovému systému (při mapování systému na model se zanedbávají některé nepodstatné detaily systému). Toto zobrazení je obecně nejednoznačné, to znamená, že jeden systém můžeme modelovat více modely podle účelu a úrovně abstrakce a jeden model může odpovídat více vzorům. [6]
- *Modelování* je proces vytváření modelů systémů na základě našich znalostí. [6]
- *Experiment* je jednou provedený proces zadání parametrů simulace, její provedení a vyhodnocení výsledků.

V následujícím textu budeme pod pojmem *simulace* myslet vždy *počítačovou simulaci*.

## 2.3 Tvorba a ověření modelu

Pojmy:

- *Abstraktní model* zjednodušený popis zkoumaného systému. Při jeho tvorbě se zaměřuje jen na charakteristické vlastnosti, které nás zajímají a které jsou podstatné pro simulaci. [6]
- *Simulační model* zápis abstraktního modelu do počítačového programu. [6]

Při zkoumání daného systému nejprve vytvoříme popisem jeho charakteristických vlastností *abstraktní model*, ze kterého pak vytvoříme *simulační model*. Vztah mezi *abstraktním a simulačním modelem* je tzv. izomorfní, neboli že simulační model odpovídá 1:1 abstraktnímu modelu z hlediska struktury a chování. [6]

Ověření simulačního modelu pak má dva stupně:

- *Verifikace* předchází procesu simulace a ověřuje korespondenci simulačního a abstraktního modelu (ověřuje izomorfní vztah mezi abstraktním modelem a simulačním modelem). [6]
- *Validace* je proces ověření platnosti výsledků simulace, kdy se snažíme dokázat, že simulační model v pro nás podstatných vlastnostech odpovídá zkoumanému systému. Problémem je, že nelze absolutně dokázat přesnost modelu. Validitu tedy chápeme jako míru použitelnosti a správnosti získaných výsledků. [6]

## 2.4 Využití simulace v praxi

Důvody, proč použít metody simulace [6]:

- Simulace probíhá na počítačích, takže při ní nemůže dojít ke hmotným škodám.
  - crash testy při vývoji automobilů
- Ze stejného důvodu jsou simulace bezpečné a odpadají zde i některé etické problémy.
  - medicínské výzkumy
- Je možno měnit rychlost simulace a během krátkého časového úseku tak odsimulovat vývoj několika tisíců let nebo naopak detailně prostudovat jevy trvající třeba jen jednotky milisekund.
  - vliv povětrnostních podmínek na kvalitu materiálů
- Simulace umožňuje provádět experimenty s modely velmi složitých systémů a také ty, které by byly jinak neproveditelné.
  - srážky kosmických těles

Díky těmto vlastnostem je simulace nasazována v širokém okruhu vědních oborů a průmyslu.

Počítačová simulace však není spásný nástroj, který vede k řešení jakéhokoliv problému, a má také své hranice. Může se stát, že vytvoření modelu v dané situaci může být příliš nákladné a nebo jednoduše existuje jiná metoda, která je schopna dát výsledky rychleji a přesněji. Při volbě metody se tak musí postupovat individuálně podle možností a požadavků té dané situace.

## 2.5 Shrnutí

Problematika simulací je sama o sobě dosti široká vědní disciplína, které se věnují mnohé zdroje [6, 12, 11], a její další rozebírání je již nad rámec této práce.

## Kapitola 3

# Dopravní simulační modely

### 3.1 Vymezení pojmů

- *Dopravním simulačním modelem* budeme rozumět dopravní síť pozemních komunikací a pohyb vozidel v ní. Dopravním modelem tak může být soustava dálnic obklopující velkoměsto, ale i jednoduché křížení kolejí linky městské hromadné dopravy se silnicí uprostřed města.
- *Dopravní simulací* pak rozumíme práci s dopravním simulačním modelem za účelem zlepšení vlastností dopravní sítě. Nejčastěji to je zvýšení propustnosti a bezpečnosti provozu na pozemních komunikacích.

### 3.2 Typy dopravních simulací

Při tvorbě dopravního simulačního modelu je důležité jaký úsek dopravní sítě budeme modelovat, ale také které vlastnosti budeme sledovat. Model dálniční sítě Evropy bude výrazně odlišný od modelu dopravy na náměstí okresního města. Stejně tak sledujeme-li chování řidičů při předjíždění, bude model odlišný od toho co sleduje vliv hustoty dopravy na životní prostředí. V praxi se dá říci, že žádné dva dopravní modely nejsou stejné.

Obecně se ale dopravní simulace dají rozdělit na tři skupiny [2, 8]:

#### 3.2.1 Makro simulace

*Makro simulace* jsou založeny na matematických modelech, které popisují toky vozidel dopravních sítí. Drtivá většina takto založených simulátorů pracuje se všemi vozidly jako s vozidly stejného typu. Jen ty nejdokonalejší jsou schopny rozeznat typy vozidel (např. osobní, nákladní, autobus), ale již nejsou schopny rozeznat jednotlivá vozidla daného typu. V praxi se používají při simulaci rozsáhlých oblastí jako jsou sítě dálnic a státních silnic.

#### 3.2.2 Mikro simulace

*Mikro simulace* nám umožňuje popisovat jednotlivá vozidla v síti a vztahy mezi nimi. U každého vozidla můžeme specifikovat jeho jízdní vlastnosti a stav, ale můžeme také určit, jak zdatný řidič vozidlo řídí a jak agresivně se na silnici chová. Všechny tyto parametry mohou závažným způsobem ovlivnit výsledek simulace. Jejich implementace je ale náročnější a vyšší nároky si klade také na výkon počítače. Využívají se proto pro simulaci spíše menších

částí dopravní sítě, mnohdy pouze jedné křižovatky. Dávají však ale výrazně přesnější výsledky odpovídající realitě.

### 3.2.3 Kombinované simulace

Třetí skupinou jsou *kombinované simulace*, které využívají vlastností a přístupů obou předchozích typů a vzájemně je kombinují. Jejich implementace není jednoduchá, avšak výsledný simulátor pak dokáže rychle reagovat díky *makro simulaci* a zároveň poskytovat přesné výsledky *mikro simulace*.

## 3.3 Řídicí modely

Důležitou součástí dopravních simulačních modelů jsou tzv. *řídicí modely*. Tyto modely jsou kostrou (šablonou) chování vozidel v síti. Definují, jaké možnosti má vozidlo v té dané situaci, a tvoří tak základní kámen celého modelu. Jednotlivé modely spolu komunikují a spolupracují, když v určitých situacích si řešení situace mezi sebou předávají [1, 5, 3].

**Poznámka:** V odborné literatuře jsou názvy těchto řídicích modelů označovány zásadně anglickými spojeními a jejich jednoduchý překlad do češtiny mně nepřijde přesný a vhodný. Proto i já budu v této práci používat tato ustálená označení.

### 3.3.1 Car-Following model

Car-Following model se uplatňuje v případech, kdy jedno vozidlo jede ve stejném jízdním pruhu za druhým. Obecně se snaží, aby nedošlo ke srážce s vozidlem jedoucím před ním. Toho lze dosáhnout dvěma způsoby:

- vozidlo upraví svoji rychlost tak, aby si udrželo dostatečně bezpečný odstup od vozidla jedoucím před ním.
- pokud je to možné, pomalejší vozidlo jedoucí před ním předjede.

Součástí Car-Following modelu je také chování v situaci, kdy vozidlo jede v jízdním pruhu úplně samo a žádné vozidlo jedoucí před ním není. V takovém případě je nejčastějším řešením zvyšování rychlosti až na hranici maximální povolené rychlosti v daném úseku dopravní sítě. Je samozřejmě nesmysl, aby vozidlo upravovalo svoji rychlost vůči vozidlu jedoucím kilometr před ním, a tak i v této situaci se bude vozidlo chovat, jako kdyby před ním žádné jiné nebylo. Konečné rozhodnutí, jak se vozidlo bude chovat, se děje na základě výpočtu tzv. *prahů* (angl. thresholds). Výpočty jejich hodnot a určení výsledného chování se ale liší u každé jednotlivé implementace dopravního modelu a nelze je nijak generalizovat.

Car-Following model se uplatňuje pouze v situacích, kdy se vozidlo nachází na úseku silnice bez křižovatek nebo při přednostním průjezdu křižovatkami (např. pokud vozidlo jede po hlavní silnici). V jiných případech totiž může nastat situace, kdy je vozidlo ve špatném jízdním pruhu z hlediska průjezdu křižovatkou. Tuto situaci je třeba řešit, stejně jako u předjíždění, přejetím do vedlejšího pruhu.

### 3.3.2 Line-Changing model

Line-Changing model přichází ke slovu ve chvíli, kdy chce vozidlo přejet do vedlejšího pruhu. Důvody jsou různé (předjetí pomalejšího vozidla, přejetí do správného pruhu, objetí překážky) a různé je také výsledné chování vozidla v reakci na ně.

První věc, ve které se liší, je pruh, do kterého bude vozidlo přejíždět. Musíme si uvědomit, že dopravní simulační modely nám poskytují výsledky nejen vzhledem k parametrům daného dopravního úseku, ale také vzhledem k zákonům o silničním provozu v dané zemi. Pokud budeme modelovat dopravu v České republice, kde se jezdí vpravo, mimo obec je možno předjet pomalejší vozidlo pouze levým jízdním pruhem. Když tedy na osmiproudé dálnici vozidlo v nejlevějším pruhu dojede pomalejší vozidlo, předjet jej nemůže, bez ohledu na to, jak jsou obsazeny ostatní pravé jízdní pruhy. Naopak při odbočení doprava nebo objetí překážky je možné, že vozidlo bude muset změnit jízdní pruh vpravo.

Pokud je v cílovém pruhu dostatek místa pro přejetí, je toto provedeno, ať už byl důvod jakýkoliv. Různá chování ale nastanou v případě, kdy změna jízdního pruhu není možná. Obecně mohou nastat tři situace:

- *Změna pruhu je vhodná.* Toto je přesně situace, kdy chce vozidlo předjet to pomalejší před ním. Pokud v takové situaci není možné přejetí provést, vozidlo zkrátka upraví svoji rychlost podle definice v Car-Follow modelu.
- *Změna pruhu je potřebná.* Vozidlo je ještě poměrně daleko od křižovatky, ale ve špatném pruhu. V případě, že přejet nemůže, má několik možností, ale obecně upraví svou rychlost tak, aby po určitém čase mohlo změnit jízdní pruh. Ani v tomto případě ale nesmíme zapomenout na to, že před ním může být jiné pomalejší vozidlo, kterému je třeba se přizpůsobit.
- *Změna pruhu je nutná.* Vozidlo v této situaci už nemá na výběr, ať už je to z jakéhokoli důvodu (překážka nebo křižovatka je již moc blízko), a musí změnit jízdní pruh. Pokud v cílovém pruhu není místo, vozidlo zkrátka zastaví a čeká, až se pruh uvolní.

Line-Changing model tedy na základě určitých informací rozhoduje, zda dojde ke změně jízdního pruhu, nebo zda vozidlo setrvá v tom aktuálním s určitou změnou rychlosti. Tyto informace, na základě kterých se rozhoduje, musí ale nejprve získat a k tomu poslouží třetí řídicí model.

### 3.3.3 Gap-Acceptance model

Gap-Acceptance model je od zbylých dvou lehce odlišný, neboť pracuje ve dvou rovinách.

Na jedné nerozhoduje o výsledném chování vozidla, ale funguje pouze jako autorita právě pro Line-Changing model. Jeho úkolem je odhadnout potřebný prostor v cílovém pruhu před a za vozidlem s ohledem na rychlost ostatních vozidel a tyto informace předat Line-Changing modelu, který si o přejetí či nepřejetí rozhodne již sám.

Druhá část Gap-Acceptance modelu je již plnohodnotným řídicím modelem, kdy rozhoduje o chování vozidla v případech, kdy dává jinému přednost. Typickým příkladem je vozidlo, které z hlavní silnice vedoucí rovně odbočuje doleva a musí dát přednost těm protijedoucím. Gap-Acceptance model tedy na základě aktuální rychlosti všech zainteresovaných vozidel rozhodne, zda je zde dostatek prostoru pro odbočení, nebo zda musí přibrzdit a počkat, až protijedoucí vozidla projedou. Za předpokladu, že výpočty *prahů* jsou korektně nastaveny, bude model pracovat bez chyb, avšak ne efektivně. Nesmíme zapomenout na pravidla silničního provozu, která umožňují současné projetí vozidel křižovatkou například v situaci, kdy obě vozidla odbočují vlevo. Pokud bychom tato pravidla neimplementovali, vozidla by si dávala přednost i v situacích, kdy nemusí, a docházelo by tak ke zbytečným průtahům doby průjezdu křižovatkou.

Stejně jako v Line-Changing modelu i zde je třeba upravovat chování vozidla nejen vzhledem k těm, kterým dává přednost, ale také k těm, které jedou ve stejném pruhu před ním. Lze ale použít poměrně elegantní řešení, ve kterém se Gap-Acceptance model dostává ke slovu až ve chvíli, kdy je vozidlo v daném pruhu první a před ním již žádné jiné není.

## Kapitola 4

# Městský dopravní simulátor

Pro demonstraci problematiky dopravních simulačních modelů vytvořím městský dopravní simulátor.

### 4.1 Specifikace

Jak již bylo zmíněno, budu se zabývat problematikou dopravy ve městech. Model se bude skládat z jednotlivých obecných segmentů dopravní sítě (křižovatka, rovný úsek, zatáčka), ze kterých pak následně bude možno sestavit model reálného dopravního systému, který je předmětem zkoumání.

Bude postaven na tzv. *mikro simulacích*. Díky tomu bude při sestavování výsledného modelu možné definovat strukturu dopravní sítě a množství vozidel, které jí projíždí, ale také detailně parametrizovat jednotlivá vozidla, a to jak po stránce konstrukce, tak po stránce chování a schopností řidiče.

Pro pohyb vozidel bude užito Car-Following, Lane-Changing a Gap-Acceptance modelů v kontextu se zadanou problematikou.

V systému bude možno sledovat zejména výskyt a charakter dopravních kolon, dobu průjezdu jednotlivými segmenty sítě a průměrnou rychlost, kterou zde vozidla dosahují. Tato data se dají uplatnit při optimalizaci průjezdu vozidel danou dopravní sítí.

Tento model bude na závěr implementován. Vytvořený městský dopravní simulátor bude podroben sérii testů na případech již existujících dopravních systémů a výsledky budou diskutovány.

### 4.2 Analýza

#### 4.2.1 Dopravní síť

Dopravní síť je chápána jako soustava jednotlivých dopravních segmentů, které jsou vzájemně propojeny ve spojovacích bodech. Jednotlivé typy segmentů jsou v modelu již přednastaveny a není možno další uživatelsky přidávat. Pokud spojovací bod některého ze segmentů není napojen k žádnému jinému, znamená to hranici zkoumané dopravní sítě. Vozidla, která do tohoto místa dorazí, ze systému mizí a v těchto místech také mohou být *generátory vozidel*.



## 4.2.2 Segment dopravní sítě

Jak již bylo řečeno, jednotlivé typy segmentů dopravní sítě (dopravních segmentů) jsou v modelu pevně přednastaveny. Každý segment se skládá z *buněk* a má pevně daný jejich počet v ose X a v ose Y. Dopravní segmenty jsou navrhnuté tak, aby oddělovaly nezávislé úseky dopravní sítě, kdy vozidlo nacházející se v jednom segmentu nijak neovlivňuje vozidlo v tom druhém (s výjimkou, kdy dvě vozidla jedoucí těsně za sebou se nacházejí na hranici mezi dvěma segmenty).

Každý segment má 2-4 očíslované spojovací body, kterými se napojuje na další segmenty. Pokud se jedná o rovný úsek nebo zatáčku, má celý segment stejný sklon vozovky. V případě křižovatek ale může mít každá větev jiný sklon.

Dopravní segment je také oblastí, ke které se vztahuje sledování většiny výstupních hodnot simulace.

## 4.2.3 Buňka dopravního segmentu

Je základním stavebním kamenem dopravního segmentu, která už dále není nijak dělitelná. Zároveň je nositelem všech informací o daném místě. Díky této vlastnosti jsou možnosti tvorby dopravních segmentů rozsáhlé.

Buňka je charakterizována zejména typem – ten udává, zda se jedná o buňku vozovky nebo buňku okolí. Buňky vozovky dále definují směr pruhu, ve kterém se nachází, maximální povolenou rychlost a případnou dopravní značku, která v daném místě platí, a sklon vozovky. Ty, které se nacházejí na samém okraji dopravního segmentu, navíc nesou číslo výstupu pro pruh vedoucí ven ze segmentu, nebo vstupu pro pruh vedoucí opačným směrem. Tato čísla se používají nejen pro propojení se sítí, ale některé dopravní segmenty (křižovatky) sledují pro jednotlivé větve specifické hodnoty.

Buňky vozovky si také udržují informace o případném vozidle, které se na jejich místě nachází.

## 4.2.4 Vozidlo

Každé vozidlo systému je modelováno jako samostatná entita, která o svém chování rozhoduje sama. Jeho konstrukční vlastnosti jsou charakterizovány jeho délkou, maximální možnou akcelerací (zrychlování) a maximální možnou decelerací (brzdění). Schopnosti řidiče jsou určeny dvěma parametry a to schopností řídit vozidlo a mírou agresivity. Díky těmto pěti hodnotám je možné definovat širokou škálu nejrůznějších vozidel, která se pohybují v dopravním systému.

Volba trasy mezi segmenty dopravní sítě se neděje na základě stochastických parametrů pro jednotlivé směry křižovatky, ale trasa je předem učena při generování vozidla. Toto řešení poskytuje určitou výhodu. Představme si zvolenou trasu jako trasu, kterou zadal řidič vozidla do svého navigačního přístroje. Mapy, které navigační přístroje obsahují, jsou v dnešní době poměrně přesné i aktuální, ale jsou to stále jen statická data. Navigační přístroj řidiči zobrazí pro něj nejvhodnější trasu k cílovému místu, ale pokud se na trase stane např. dopravní nehoda, která vozovku zneprůjezdí, dostane se vozidlo do dopravní zácpy.

Zvolené řešení ale umožňuje simulovat aktivní systém navigace, který v případě, že je nahlášena ona dopravní nehoda, upozorní na tuto skutečnost řidiče a změni naplánovanou trasu k cíli. Vozidlo se tak dopravní zácpě vyhne a zlepši se tak průjezdnost daného úseku.

#### 4.2.5 Generátor vozidel

Generátor vozidel je zdrojem vozidel pro dopravní síť. Je navázán na vstupní spojovací bod dopravního segmentu, který ale nesmí být připojen. Na jeden vstupní bod může být navázáno i více generátorů, jeden generátor ale může být navázán pouze na jeden vstupní bod.

Jeho parametry jsou procentuální zastoupení výskytu jednotlivých typů vozidel, řidičů jednotlivých stupňů zručnosti a agresivity a naplánovaných tras. Je tedy možné, aby jeden generátor produkoval vozidla různých typů jedoucí různými směry, ale je také možno generovat vozidla jednoho typu jedoucí jedním směrem jedním generátorem a vozidla jiného typu jedoucí jiným směrem druhým.

Vozidla jsou generována stochasticky dle zadaného intervalu. Generování probíhá standardně po celou dobu simulace, avšak tuto dobu lze omezit.

#### 4.2.6 Pohyb vozidel

O pohyb vozidel v síti se stará tzv. *řídící dozorce* (inspektor). Vždy po uplynutí stanoveného časového úseku provede jeden krok simulace. Ten má tři fáze:

- *Vyhodnocení situace*

Skládá se ze zjištění aktuálního stavu, porovnání s úkony, které musí vozidlo provést, a nalezení optimálního řešení situace. Jedná se o velice složitý proces, který tvoří inteligenci vozidla. Kombinují se zde všechny tři řídicí modely, které jsou nastaveny tak, že vozidla svým chováním dodržují pravidla silničního provozu (jako předlohou sloužila pravidla provozu na pozemních komunikacích platná v České republice [4]) a není možné, aby došlo ke kolizi. Řešením situace se rozumí úprava aktuální akcelerace vozidla (může být i záporná – decelerace) a případné rozhodnutí o změně jízdního pruhu (viz. 4.2.7).

- *Posun vozidla*

Na základě aktuální hodnoty akcelerace se dopočítá aktuální hodnota rychlosti vozidla a provede se posun o vzdálenost, kterou vozidlo touto rychlostí urazí za stanovený časový úsek – doba mezi dvěma aktivacemi *řídícího dozorce*.

- *Vyhodnocení sledovaných hodnot a příprava dalšího kroku*

V této fázi jsou zapsány hodnoty do všech statistik, které se pravidelně aktualizují každým krokem simulace, a provedeny další úkony potřebné pro přípravu dalšího kroku.

Onen jeden krok simulace vypadá tak, že *řídící dozorce* nejprve provede vyhodnocení situace u všech vozidel, pak všechna vozidla provedou posun a na konec dojde u všech k vyhodnocení hodnot a přípravě dalšího kroku. Toto řešení zabraňuje, aby docházelo k nekonzistentnosti dat během jednoho kroku, kdy by některá vozidla měnila svoji polohu dříve, než jiná provedou vyhodnocení situace.

#### 4.2.7 Režimy

Jak již bylo řečeno, řešením aktuální dopravní situace se rozumí úprava aktuální akcelerace vozidla a případná změna jízdního pruhu. To se děje na základě volby vhodného tzv.

*režimu*. To je detailní specifikace úpravy chování vozidla pro danou situaci. Výčet režimů seskupených podle příslušnosti k jednotlivým modelům:

- *Car-Following model*.
  - Free Flow
  - Car Follow
  - Emergency
  - Tailback
- *Lane-Changing model*.
  - Lane Change
- *Gap-Acceptance model*.
  - Intersection Through
  - Intersection Yield
  - Yielding

### Režimy Car-Following modelu

Vozidlo nacházející se na rovném úseku v minimální bezpečné vzdálenosti od vozidla před ním je v režimu *Free Flow* a může akcelarovat až na maximální povolenou rychlost v daném úseku. Ona bezpečná vzdálenost je dána prahem, označíme si jej  $thresholdA$ . Udává vzdálenost, za kterou je vozidlo schopno bezpečně zastavit, aniž by muselo nouzově brzdit (využití maximální možné decelrace vozidla), i poté, co bude ještě po dobu jednoho kroku zrychlovat. Tato vzdálenost však musí být menší než 50m – omezeno jako dohled řidiče z vozidla. Hodnota prahu je vypočtena jako

$$thresholdA = \min(50, \frac{(v * a_{am} * t_{step})^2}{a_{dm}} + d_{safe})$$

kde  $v$  je aktuální rychlost vozidla,  $a_{am}$  maximální možné zrychlení,  $t_{step}$  doba jednoho kroku simulace,  $a_{dm}$  maximální možné zpomalení a  $d_{safe}$  je charakteristika řidiče udávající bezpečný odstup od vozidla vpředu.

Pokud v této vzdálenosti je nějaké jiné vozidlo, vstupuje do hry druhý práh –  $thresholdB$ . Ten udává vzdálenost, za kterou je vozidlo schopno bezpečně zpomalit na rychlost vozidla jedoucího před ním, aniž by muselo nouzově brzdit, i přes to, že bude přední vozidlo zpomalovat. Toto samozřejmě platí jen v případě, kdy je přední vozidlo pomalejší, jinak je práh nulový. Hodnota prahu je vypočtena jako

$$thresholdB = \begin{cases} \frac{(v - v_p)^2}{a_{dm}} + d_{safe} & \text{pro } v \geq v_p \\ 0 & \text{pro } v < v_p \end{cases}$$

kde  $v$  je aktuální rychlost vozidla,  $v_p$  je aktuální rychlost předního vozidla,  $a_{dm}$  maximální možné zpomalení a  $d_{safe}$  je charakteristika řidiče udávající bezpečný odstup od vozidla vpředu.

Když je vzdálenost mezi vozidly větší než  $thresholdB$ , jedná se o *Car Follow* režim. V tomto režimu se vozidlo snaží nijak dramaticky neměnit vzdálenost mezi oběma vozidly

a rychlost jízdy sladit s vozidlem jedoucím vepředu. Dochází tedy k mírnému brzdění, udržování rychlosti nebo akceleraci v závislosti na tom, které z vozidel je rychlejší a na tom, zda přední vozidlo brzdí. Míra akcelerace se pak odvíjí od vzdálenosti samotné.

V situaci, kdy je vzdálenost mezi vozidly menší nebo rovna prahu  $thresholdB$ , nastává riziko srážky, tedy režim *Emergency*. Vozidlo se primárně pokusí o předjetí. To však není možné, když v cílovém pruhu není místo, nebo když se nachází blíže jak 50m od křižovatky. Předjíždění při průjezdu křižovatkou je totiž z bezpečnostních důvodů zakázáno. Pokud je platná alespoň jedna z těchto podmínek, vozidlo musí začít brzdit tak, aby bezpečně zpomalilo na rychlost předního vozidla.

Režim *Tailback* je speciální režim, který se stará o úplné zastavení vozidla při zařazení se do kolony např. před křižovatkou.

### Režimy Lane-Changing modelu

Lane-Changing modelu přísluší jediný režim a to *Lane Change*. Jak už název napovídá, stará se o přejezd z jednoho jízdního pruhu do druhého. Důležité ale je, že ve chvíli, kdy vozidlo jednou začne měnit jízdní pruh, již nemůže tento proces zastavit. S tímto je potřeba počítat zejména při odhadování volného místa v době, kdy se o přejetí teprve rozhoduje.

V průběhu přejezdu se vozidlo chová tak, jako by jelo v obou pruzích zároveň a musí dbát na dodržování bezpečného odstupu od vozidel v obou pruzích. Po určitém čase vozidlo přejeđe do nového jízdního pruhu a funkce *Lane Change* režimu je tak u konce.

### Režimy Gap-Acceptance modelu

Tyto režimy se používají pro průjezd křižovatkami. Pokud je vozidlo blíže jak 50m od křižovatky, na základě dopravního značení a směru, kterým chce křižovátku opustit, se určí, zda musí při průjezdu dávat některému směru přednost či nikoliv.

Vozidlo jedoucí křižovatkou rovně po hlavní silnici nikomu přednost dávat nemusí a je tedy v *Intersection Through* režimu. Jedná se sice o režim průjezdu křižovatkou, ale chová se přesně jako režimy Car-Following modelu, tedy udržuje bezpečnou vzdálenost od vozidel před ním s tím, že pomalejší vozidla nebude nikdy předjíždět (nachází se v blízkosti křižovatky). Pouze v případě, kdy ona hlavní silnice vede např. doprava, vozidlo mírně zpomalí na rychlost asi 25km/h (pokud jede rychleji), protože odbočení v plné rychlosti není možné.

Pokud však při průjezdu křižovatkou vozidlo musí dávat přednost některému ze směrů, je aktivován *Intersection Yield* režim. Jeho úkolem je docílit, aby vozidlo ve vzdálenosti 10m od křižovatky jelo rychlostí 18km/h (s jistou tolerancí). Tato situace poskytuje dobrou pozici pro řidiče, který dává přednost příjezdějícím vozidlům. Vozidlo v *Intersection Yield* režimu je totiž ještě daleko a do křižovatky „nevidí“.

Vozidlo, které je méně jak 10m od křižovatky, je v *Yielding* režimu. Pokud se ke křižovatce blíží jiné vozidlo, kterému je třeba dát přednost, začne řidič brzdit tak, aby dobrzdil na hranici křižovatky. V opačném případě může volně projet křižovatkou a použije k tomu jeden z režimů Car-Following modelu. Onen prostor, který vozidlo potřebuje k projetí křižovatky, aniž by musel dávat přednost se vypočte následovně:

- minimální vzdálenost, kterou musí vozidlo urazit, aby projelo křižovatkou je

$$d_{min} = d_d + d_k + d_{safe} + d_p$$

kde  $d_d$  je délka vozidla,  $d_k$  vzdálenost od hranice křižovatky,  $d_{safe}$  bezpečný odstup a  $d_p$  je vzdálenost, kterou musí urazit k projetí středu křižovatky (závisí na počtu v jednotlivých větvích křižovatky a na směru, kterým bude vozidlo křižovatku opouštět).

- čas potřebný pro tento manévr je

$$t = \frac{\sqrt{v^2 + 2 * a_a * d_{min}} - v}{a_a}$$

kde  $v$  je rychlost vozidla,  $a_a$  je maximální možná akcelerace a  $d_{min}$  je minimální vypočtená vzdálenost.

- hodnota minimálního prostoru, ve kterém se nesmí nacházet žádné příjezdějící vozidlo ze směru, kterému dává přednost, nesmí být větší jak 40m – omezení výhledu řidiče z vozidla. Je tedy určena jako

$$d = \min(40, v_p * t + d_{safe})$$

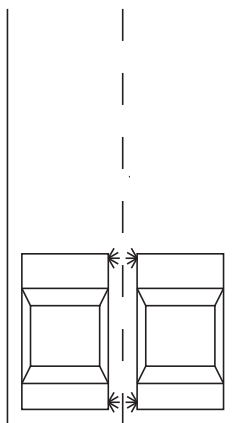
kde  $v_p$  je rychlost příjezdějícího vozidla,  $t$  je potřebný čas našeho vozidla pro projetí křižovatky a  $d_{safe}$  bezpečný odstup.

#### 4.2.8 Uvíznutí

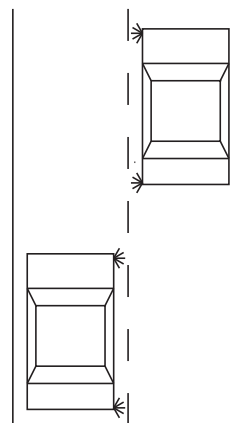
V modelu jsou řešena tzv. *uvíznutí* (angl. deadlocks). To jsou situace, kdy z nějakého důvodu dojde k trvalému zastavení vozidel v jednom nebo více ramenech dopravního segmentu.

##### Výměna jízdních pruhů

Toto nastane, když na silnici, která má alespoň dva pruhy v každém směru, vedle sebe jedou dvě vozidla, která si potřebují vyměnit pozice – neboli to v pravém pruhu potřebuje přejet do levého a naopak (obr. 4.1). Základním chováním pro případ, kdy vozidlo potřebuje přejet do jiného pruhu, kde ale není místo, je zpomalování. V tomto případě tak ale činí obě vozidla až dojde k jejich úplnému zastavení.



Obrázek 4.1: Uvíznutí při výměně jízdních pruhů



Obrázek 4.2: Problém uvíznutí je vyřešen

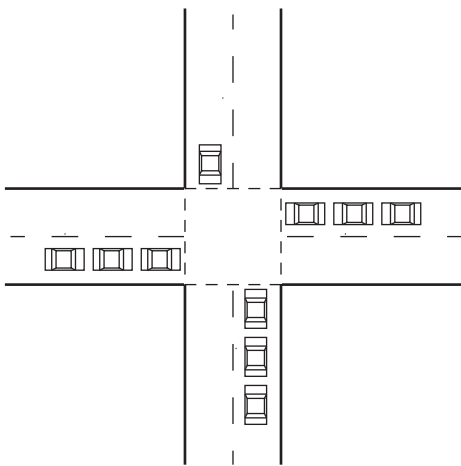
Řešení je ale poměrně jednoduché a vyplývá už z pravidel silničního provozu [4]. Vozidlo jedoucí v levém pruhu musí dát v takovém případě přednost zprava vozidlu v pravém.

Znamená to, že zpomalovat bude pouze vozidlo v levém pruhu, kdežto vozidlo v pravém bude mírně zrychlovat (pokud mu to daná situace dovolí). Po určité době se vozidla dostanou dále od sebe a obě mohou změnit jízdní pruh (obr. 4.2).

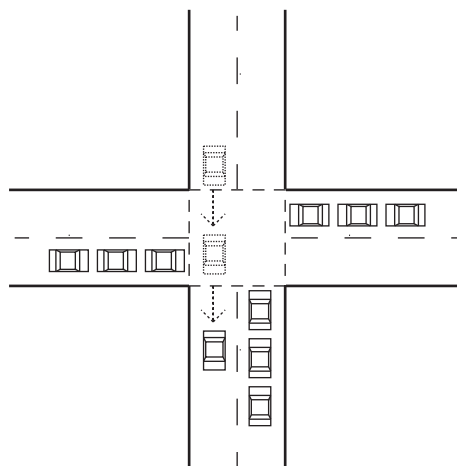
### Zaplnění křižovatky

Tato situace se týká pouze křižovatek, které nejsou řízené a které nemají určené ani hlavní a vedlejší silnice dopravními značkami. Platí zde tzv. přednost zprava [4], kdy vozidlo dává přednost vždy vozidlům, které jsou napravo. Představme si ale klasickou křižovatku typu kříž (obr. 4.3), na kterou zároveň ze všech stran přijela vozidla. Každé z vozidel má napravo jiné, kterému musí dát přednost, a přednosti tak tvoří kruh – dochází k uvíznutí.

Řešení však není ani v pravidlech silničního provozu [4] a řidiči se musí domluvit. V modelu to vypadá tak, že vozidlo, které má za sebou nejmenší kolonu, je pověřeno přednostním průjezdem křižovatkou (obr. 4.4). Výběr větve s nejmenší kolonou navíc umožní návrat k normálnímu chodu křižovatky v co nejkratším čase.



Obrázek 4.3: Uvíznutí při zaplnění křižovatky



Obrázek 4.4: Problém uvíznutí je vyřešen

## 4.3 Implementace

Městský dopravní simulátor je implementován v jazyce C++ s podporou simulační knihovny SIMLIB/C++ ve verzi 2.18 [7]. Pro načtení dopravní sítě ze XML souboru je použita knihovna TinyXML verze 2.5.3 [9]. Je přeložitelný na operačních systémech MS Windows a Linux (testováno na MS Windows XP Professional a Linux CentOS).

### 4.3.1 Vozidla

V simulátoru jsou přednastaveny tři typy vozidel, tedy tři sady parametrů:

1. *osobní automobil*

- délka: 4 m
- maximální akcelerace:  $2.65 \text{ m/s}^2$
- maximální decelerace:  $6.67 \text{ m/s}^2$
- počáteční rychlost: 4 m/s (14.4 km/h)

2. *nákladní automobil*

- délka: 8 m
- maximální akcelerace:  $1.73 \text{ m/s}^2$
- maximální decelerace:  $5.14 \text{ m/s}^2$
- počáteční rychlost: 4 m/s (14.4 km/h)

3. *autobus*

- délka: 12 m
- maximální akcelerace:  $1.28 \text{ m/s}^2$
- maximální decelerace:  $4.84 \text{ m/s}^2$
- počáteční rychlost: 4 m/s (14.4 km/h)

**Poznámka:** Parametry byly určeny na základě reálných hodnot, kterých dosahují běžná vozidla daného typu. Jsou však jen orientační a slouží pro základní rozdělení vozidel dle jednotlivých typů.

Jak již bylo řečeno, při generování vozidla je možno parametrizovat i chování řidiče a to určením jeho schopností ( $S$ ) a agresivity ( $A$ ) – nabývají hodnot celých čísel 1-5. Tyto hodnoty však neovlivňují simulaci přímo, ale pomocí nich se určí jiné, pro chování vozidla podstatné, parametry:

- *Odhad vzdálenosti.* vzorec:  $1.5 - 0.1 * S$

Schopnost odhadu vzdálenosti, nebo také odhad prostoru, se uplatňuje např. při přejezdu vozidla do jiného pruhu. Zkušený řidič ví, kolik prostoru na manévr potřebuje, kdežto méně schopný řidič potřebuje prostoru více, než je nutné, aby se k přejetí odhodlal.

- *Bezpečná vzdálenost.* vzorec:  $5.5 - 0.7 * A$

Podle tohoto parametru si řidiči nechávají minimální odstup od vpředu jedoucího vozidla, nebo je také použit jako rezerva při odhadu vzdálenosti příjíždějícího vozidla, kterému dávají přednost. Platí, že agresivnější řidiči mají tuto bezpečnostní rezervu menší než ti klidnější.

- *Chování při akceleraci.* vzorec:  $0.4 + 0.08 * S + 0.04 * A$

Určuje, jak řidič dokáže využít akcelerační potenciál vozidla. Závisí nejen na schopnostech řidiče, ale i na agresivitě a to v kladném slova smyslu (i profesionální řidič v běžném provozu nebude na každé křižovatce akcelerovat na maximum, když je klidný a nikam nespěchá).

- *Chování při deceleraci.* vzorec:  $0.75 + 0.05 * S$

Určuje jak řidič dokáže využít potenciál vozidla při brzdění, závisí však pouze na schopnostech řidiče.

Vzorce a konstanty pro výpočet parametrů byly zvoleny experimentálně tak, aby se chování řidičů v simulátoru co nejvíce podobalo realitě.

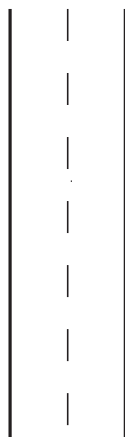
### 4.3.2 Dopravní segmenty

Při sestavování dopravní sítě se spojují jednotlivé dopravní segmenty různých typů. V simulátoru je implementováno několik typů dopravních segmentů, které mají pevně očíslovány spojovací body. Dají se rozdělit do tří skupin:

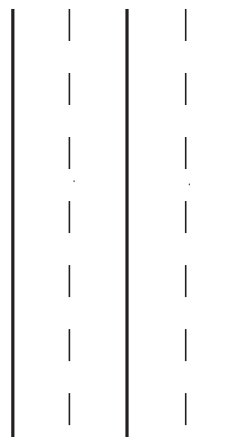
#### Rovné úseky a zatáčky

- ROAD2 - rovný úsek vozovky, dva pruhy (obr. 4.5)
- ROAD4 - rovný úsek vozovky, čtyři pruhy (obr. 4.6)
- TURN2 - úsek zatáčky, dva pruhy
- TURN4 - úsek zatáčky, čtyři pruhy

Délka všech těchto úseků je 50m.



Obrázek 4.5: ROAD2 - rovný úsek vozovky, dva pruhy



Obrázek 4.6: ROAD4 - rovný úsek vozovky, čtyři pruhy

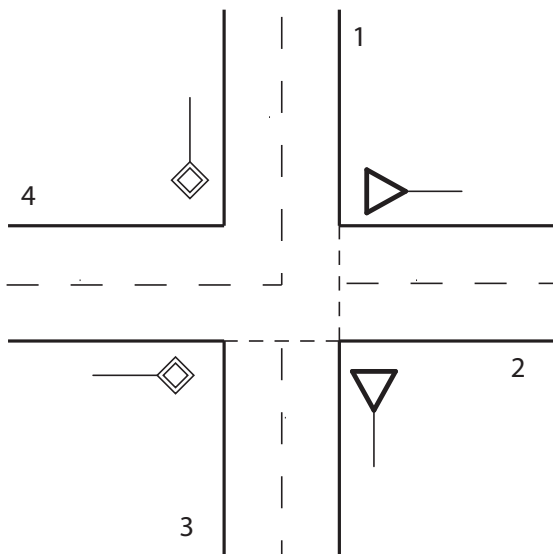


## Křižovatky typu kříž

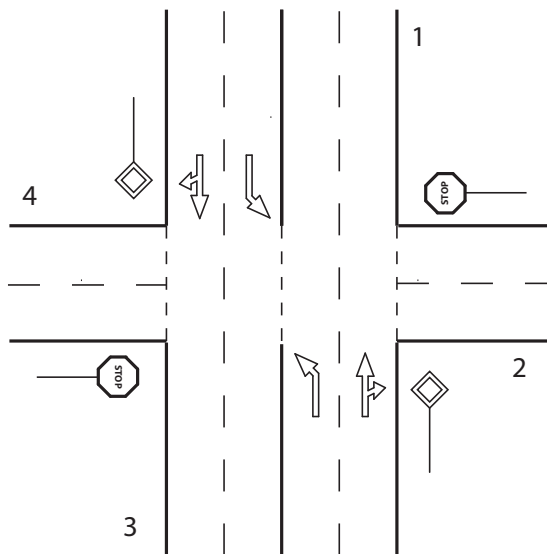
Obecně XROAD\_A\_B\_C - A určuje počet příjezdových pruhů ze spojovací bodů 1 a 3 (vertikálně), B určuje počet příjezdových pruhů ze spojovací bodů 2 a 4 (horizontálně) a C určuje typ dopravních značek – směr hlavní silnice ze směru od spojovacího bodu 1 (shora).

- XROAD\_1\_1\_NS - bez dopravních značek
- XROAD\_1\_1\_MAIN - hlavní silnice rovně, z vedlejší "DEJ PŘEDNOST V JÍZDĚ"
- XROAD\_1\_1\_MAIN\_LEFT - hlavní silnice doleva, z vedlejší "DEJ PŘEDNOST V JÍZDĚ"
- XROAD\_1\_1\_MAIN\_RIGHT - hlavní silnice doprava, z vedlejší "DEJ PŘEDNOST V JÍZDĚ" (obr. 4.7)
- XROAD\_1\_1\_MAIN\_STOP - hlavní silnice rovně, z vedlejší "STOP"
- XROAD\_1\_1\_MAIN\_STOP\_LEFT - hlavní silnice doleva, z vedlejší "STOP"
- XROAD\_1\_1\_MAIN\_STOP\_RIGHT - hlavní silnice doprava, z vedlejší "STOP"
- XROAD\_2\_1\_MAIN - hlavní silnice rovně, z vedlejší "DEJ PŘEDNOST V JÍZDĚ"
- XROAD\_2\_1\_MAIN\_STOP - hlavní silnice rovně, z vedlejší "STOP" (obr. 4.8)

Rozměr všech křižovatek je shodný a to 50x50m.



Obrázek 4.7: křižovatka typu kříž:  
XROAD\_1\_1\_MAIN\_RIGHT



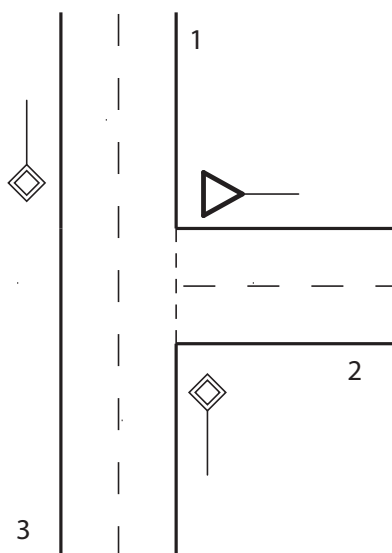
Obrázek 4.8: křižovatka typu kříž:  
XROAD\_2\_1\_MAIN\_STOP

## Křižovatky typu T

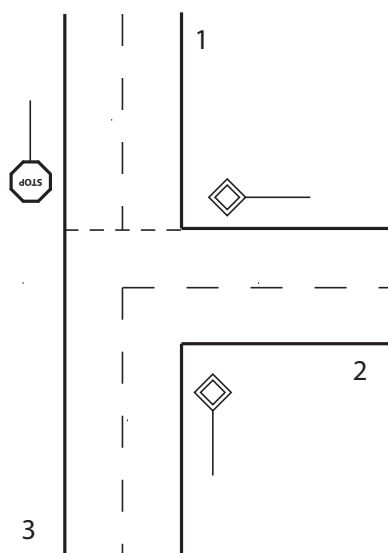
Obecně TROAD\_A - A určuje dopravní značku popř. směr hlavní silnice ze směru od spojovacího bodu 1 (shora).

- TROAD\_NS - bez dopravních značek
- TROAD\_MAIN - hlavní silnice rovně, z vedlejší "DEJ PŘEDNOST V JÍZDĚ" (obr. 4.9)
- TROAD\_MAIN\_LEFT - hlavní silnice doleva, z vedlejší "DEJ PŘEDNOST V JÍZDĚ"
- TROAD\_MAIN\_STOP - hlavní silnice rovně, z vedlejší "STOP"
- TROAD\_MAIN\_LEFT\_STOP - hlavní silnice doleva, z vedlejší "STOP"
- TROAD\_YIELD - od spoj. bodu 1 je vedlejší se značkou "DEJ PŘEDNOST V JÍZDĚ"
- TROAD\_STOP - od spoj. bodu 1 je vedlejší se značkou "STOP" (obr. 4.10)

Rozměr všech křižovatek je shodný a to 50x50m.



Obrázek 4.9: křižovatka typu T:  
TROAD\_MAIN



Obrázek 4.10: křižovatka typu T:  
TROAD\_STOP

### 4.3.3 Načtení dopravní sítě

Jak již bylo zmíněno, struktura dopravní sítě včetně charakteristiky vozidel a řidičů je uložena mimo aplikaci v XML souboru. Příklad souboru definující dva rovné propojené úseky silnice a generátor různých typů vozidel na jednom konci:

```
<!-- verze jazyka XML -->
<?xml version="1.0" ?>
<!-- začátek definice dopravní sítě -->
<Network>
  <!-- definice plánů tras -->
  <RoutePlans>
    <!-- plán trasy s číselným označením -->
    <RoutePlan number="1">
      <!-- položka trasy -->
      <!-- "segname" - dopravní segment; -->
      <!-- "direction" - směr průjezdu segmentem -->
      <Item segname="rovna1" direction="straight" />
      <Item segname="rovna2" direction="straight" />
    </RoutePlan>
  </RoutePlans>
  <!-- začátek definice dopravních segmentů -->
  <Segments>
    <!-- dopravní segment s generátorem -->
    <!-- "name" - jedinečné jméno segmentu (max. 16 znaků)-->
    <!-- "type" - typ segmentu -->
    <!-- "gradient1" - udává sklon vozovky v procentech -->
    <Segment name="rovna1" type="ROAD2" gradient1="10">
      <!-- generátor vozidel náležící segmentu "rovna1" -->
      <!-- "socket" - číslo spojovacího bodu pro generování vozidel -->
      <!-- "interval" - časový interval generování vozidel(v sekundách) -->
      <!-- "start" "end" - čas generování od-do (nepovinné), pokud není -->
      <!-- uvedeno nebo je hodnota 0 -> generuje po celou dobu simulace -->
      <Generator socket="1" interval="5" start="10" end="60">
        <!-- procentuální zastoupení jednotlivých typů vozidel -->
        <VehicleParams>
          <!-- osobní automobil -->
          <Item value="80" />
          <!-- nákladní automobil -->
          <Item value="10" />
          <!-- autobus -->
          <Item value="10" />
        </VehicleParams>
      </Generator>
    </Segment>
  </Segments>
</Network>
```

```

<!-- procentuální zastoupení řidičů dle schopností (1-5) -->
<SkillParams>
  <Item value="20" />
  <Item value="20" />
  <Item value="20" />
  <Item value="20" />
  <Item value="20" />
</SkillParams>
<!-- procentuální zastoupení řidičů dle agresivity (1-5) -->
<AggressiveParams>
  <Item value="20" />
  <Item value="20" />
  <Item value="20" />
  <Item value="20" />
  <Item value="20" />
</AggressiveParams>
<!-- procentuální zastoupení užití jednotlivých plánů tras -->
<RoutePlanParams>
  <Item number="1" value="100" />
</RoutePlanParams>
</Generator>
</Segment>
<!-- dopravní segment bez generátoru -->
<Segment name="rovna2" type="ROAD2" gradient1="10">
</Segment>
</Segments>
<!-- začátek definice spojení dopravních segmentů -->
<Connections>
  <!-- jedno spojení dvou dopravních segmentů -->
  <!-- není povinné např. při modelování jen jednoho segmentu -->
  <Connection>
    <!-- první a druhý segment spojení -->
    <!-- "name" - jméno segmentu -->
    <!-- "socket" - číslo bodu, kterým se bude napojovat -->
    <First segname="rovna1" socket="2" />
    <Second segname="rovna2" socket="1" />
  </Connection>
</Connections>
</Network>

```

Podrobný popis jednotlivých tagů a atributů lze nalézt v uživatelské příručce (příloha [A](#)).

#### 4.3.4 Parametry simulátoru

Simulátor se ovládá z příkazové řádky pomocí parametrů programu.

Povinné parametry:

- `<xml_file>` - jméno XML souboru se strukturou dopravní sítě a charakteristikou vozidel
- `<simulation_time>` - doba, po kterou bude simulace probíhat. Čas je ve formátu hh:mm a jeho maximální hodnota je omezena na 10000 hodin.

Nepovinné parametry:

- `-a <segment_name>` - jméno detailně sledovaného dopravního segmentu, který bude v každém kroku simulace vykreslen na výstup v ASCII módu.
- `-v <mode>` - nastavuje mód, ve kterém se budou vypisovat informace o vstupu/výstupu vozidel do/ze systému.
  - `-v start` - vypisuje informace o vygenerování nových vozidla do sítě.
  - `-v end` - vypisuje informace o tom, že vozidlo dorazilo na hranici dopravní sítě.
  - `-v all` - spojuje funkce obou předešlých módů.
  - pokud není mód definován, nevypisují se žádné informace o vozidlech.

Nápověda k ovládání simulátoru je vytištěna po spuštění s jediným parametrem `-h`. Podrobné ovládání programu včetně příkladů lze nalézt v uživatelské příručce (příloha A).

#### 4.3.5 Výstupy simulátoru

Nejprve si ujasněme některé pojmy:

- *Doba průjezdu segmentem* je doba, která uplyne mezi vjezdem vozidla do segmentu a jeho opuštěním (moment, kdy vozidlo vjede do dalšího segmentu, popř. dorazí na hranici dopravní sítě). Sleduje se u rovných úseků a zatáček.
- *Doba přiblížení* je čas, který vozidlo potřebuje, aby projelo křižovatkou, neboli čas od vjezdu do segmentu do najetí do křižovatky (ve chvíli, kdy vozidlo najede do křižovatky, už nezastavuje a křižovátku projede).
- *Délka kolony*. Pod pojmem kolona rozumíme souvislou řadu vozidel v jednom jízdním pruhu, pro které platí, že maximální odstup mezi vozidly je 8 metrů a maximální rychlost každého vozidla je 9 km/h (2.5 m/s). Délka kolony je pak vzdálenost posledního vozidla, která splňuje předešlé podmínky, od hranice křižovatky včetně vozidla samotného. Zmíněné parametry byly určeny experimentálně tak, aby se simulované kolony co nejvíce podobaly těm reálným.

Výstupy simulátoru se liší podle zadaných volitelných parametrů. Mohou být obsáhlejšího charakteru, proto je vhodné si výstup přesměrovat do souboru a data prohlížet až tam.

- *Statistika průjezdu rovným úsekem nebo zatáčkou*

Vypíše se vždy souhrnně na konci simulace pro každý segment. Zobrazí:

- dobu průjezdu daným segmentem
- průměrnou rychlost, které vozidlo při průjezdu dosáhlo

- *Statistika průjezdu křižovatkou*

Vypíše se vždy souhrnně na konci simulace pro každý segment. Pro každý z příjezdů (3 nebo 4 v závislosti na typu křižovatky) zobrazí:

- dobu přiblížení ke křižovatce
- délku kolon, pokud se v daném směru nějaké vyskytly

- *Detailní sledování dopravního segmentu*

Zapíná se parametrem `-a`, kterým se určí jméno sledovaného segmentu. Při detailním sledování je po provedení každého kroku simulace na výstup zobrazen stav daného segmentu v ASCII módu, včetně všech vozidel, které se v něm nacházejí. Umožňuje tak bližší pohled na chování jednotlivých vozidel.

- *Informace o vozidlech*

Zapíná se parametrem `-v`. Podle zvoleného módu jsou vypisovány informace o nově vygenerovaných vozidlech a o vozidlech, která dorazila na hranici dopravní sítě. Výpis oznamuje, kdy a kde se daná událost stala a o jaké vozidlo se jedná – rozlišeny jsou podle času vjezdu do sítě.

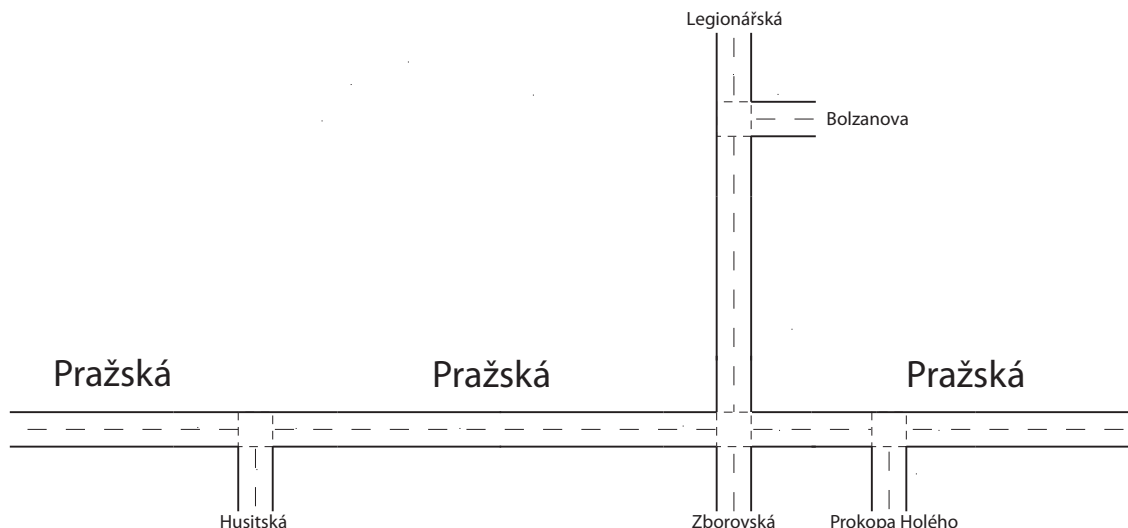
Podrobný popis výstupů včetně příkladů lze nalézt v uživatelské příručce (příloha [A](#)).

## 4.4 Experiment

V rámci testování simulátoru jsem namodeloval reálnou dopravní situaci. Jedná se o úsek znojemské frekventované ulice Pražská (obr. [4.11](#)).

Ulice Pražská je hlavní tah ze Znojma směrem na Moravské Budějovice, který je využíván nejen pro osobní, ale i nákladní dopravu. Jezdí zde autobusy městské hromadné dopravy a to jak od křižovatky s ulicí Husitskou, tak po ulici Legionářské. Detailně budeme sledovat zejména křižovatku Pražská-Legionářská. Za normální situace je doprava na ní klidná, menší kolony se někdy tvoří jen ze směru od ulice Legionářské (`direction4`).

```
Simulation time: 1:00:00.0
+-----+
| Direction 4:                               |
+-----+
| Tailback length: (m)                       |
+-----+
| Min = 2                                     Max = 10                               |
| Number of records = 279                    |
| Average value = 3.8                         |
+-----+
```



Obrázek 4.11: Schéma modelovaného dopravního úseku

Hlavním tahem ze Znojma směrem na Přímětice je ulice Přímětická (nebyla modelována), jež spojuje ulice Pražskou a Legionářskou. V případě, kdy ulice Přímětická bude uzavřena, např. z důvodu rekonstrukce, stanou se právě ulice Pražská a Legionářská hlavním tahem ze Znojma na Přímětice. Hustota dopravy se v tomto úseku zvýší, což povede ke tvorbě kolon u křižovatky Pražská-Legionářská, zejména ze směru od ulice Legionářské (`direction4`).

Simulation time: 1:00:00.0

```

+-----+
| Direction 4:                                     |
+-----+
| Tailback length: (m)                           |
+-----+
| Min = 2                                         Max = 28   |
| Number of records = 2655                       |
| Average value = 6.3                             |
| Standard deviation = 4.9                       |
+-----+

```

Řešením této situace může být změna dopravního značení křižovatky Pražská-Legionářská. Hlavní silnice nebude podél ulice Pražské, ale bude jí spojnice ulic Pražská a Legionářská. Tím se sice začnou tvořit menší kolony z obou směrů ulice Pražské (`direction1` a `direction3`), ale ty z ulice Legionářské (`direction4`) budou naprosto eliminovány.

Simulation time: 1:00:00.0

```
+-----+
| Direction 1:                               |
+-----+
| Tailback length: (m)                       |
+-----+
| Min = 2                                     Max = 20 |
| Number of records = 343                    |
| Average value = 5.8                        |
| Standard deviation = 4.9                   |
+-----+
+-----+
| Direction 3:                               |
+-----+
| Tailback length: (m)                       |
+-----+
| Min = 2                                     Max = 12 |
| Number of records = 1270                   |
| Average value = 4.7                        |
| Standard deviation = 2.4                   |
+-----+
+-----+
| Direction 4:                               |
+-----+
| Tailback length: (m)                       |
+-----+
| no record                                  |
+-----+
```

Ukázkové experimenty včetně jejich výpisů je možno najít na příloženém CD v adresáři /linux/ popř. /win32/ dle operačního systému.

## 4.5 Budoucí vývoj

Implementace simulátoru je plně funkční, ale je to pouze základ pro další práci. V následující části uvedu hrubý nástin možností pro další vývoj.

### 4.5.1 Semafory

Ke stavbě dopravní sítě je možno použít jen křižovatek bez řízeného provozu, tedy křižovatek, kde platí přednost zprava, nebo přednosti určují dopravní značky. Ve městech je však mnoho křižovatek řízených semaforey. Proto jsou tyto jedním z rozšíření, která budou implementována nejdříve.

Tím vzniká prostor pro další zajímavé téma a to synchronizaci světelných křižovatek. Každý řidič zná tu situaci, kdy stojí na křižovatce na červenou, po přepnutí na zelenou se rozjíždí, ale ujede jen pár desítek metrů k další křižovatce, kde je opět červená. Jsou místa a situace, kde vzhledem k okolnostem lepší řešení neexistuje, ale pokud se to stává pravidelně na hlavním tahu městem (městském okruhu), je něco špatné. Lepší synchronizace světelných křižovatek pak může zvýšit plynulost provozu a ušetřit nemalé peníze.



## 4.5.2 Dopravní segmenty

Naprogramování světelných křižovatek, ale i dalších dopravní segmentů, povede k rozšíření využitelnosti simulátoru. Dopravní segmenty typu kruhového objezdu, zúžení nebo naopak rozšíření jízdních pruhů nebo nájezdy a sjezdy včetně průběžných pruhů jsou také možnými rozšířeními.

## 4.5.3 Monitorování kolizí

Dopravní kolize jsou zajímavé tím, že mohou být chápány ve dvou odlišných rovinách:

- *Kolize jako sledovaná hodnota.* Nastavením parametrů počtu vozidel v síti nebo agresivity řidičů můžeme docílit toho, že bude v simulaci docházet ke kolizím (pokud to simulátor umožňuje) a sledovat tyto situace.
- *Kolize jako parametr.* Kolize jako parametr simulace umožňuje určit přesně kdy a kde k ní dojde, připravit scénář situace a sledovat jeho další vývoj.

Osobně se domnívám, že dopravní nehoda je jev dosti náhodný, přestože hustý provoz a nervózní řidiči k němu mohou přispět, a jeho parametrizace je téma na samostatnou práci. Proto pokud bude simulátor doplněn o kolize, rozhodně to budou kolize jako parametr simulace.

S dopravními nehodami souvisí také aktivní navigace (viz. 4.2.4), kdy např. při dopravní nehodě jsou ostatní řidiči o této informováni a mohou se danému úseku vyhnout. Nemusí se však jednat jen o nehody, ale také o obyčejné odpolední dopravní zácpy, které jsou ve větších městech na denním pořádku. Tento systém tak může být velkým pomocníkem při řešení krizových situací v dopravě.

## 4.5.4 Městská hromadná doprava

Při simulaci dopravy ve městě je potřeba počítat také s městskou hromadnou dopravou. Autobusy jsou v simulátoru již implementovány a trolejbusy se od nich svým chováním výrazně neliší.

Tramvaje však dopravu ovlivňují zásadním způsobem. Nejen že jezdí po kolejích (přibývají další typy dopravních segmentů jako přejezdy apod.), ale tramvaje mají v určitých případech přednostní právo průjezdu křižovatkou před ostatními vozidly.

## 4.5.5 Grafická nastavba

Vytvořený simulátor je jen konzolová aplikace. Parametry, struktura dopravní sítě i výstupy jsou pouze textové. Na využitelnosti simulátoru to nic neubírá, ale práce v textovém režimu je náročnější a obtížnější, což může vést i k chybám.

Řešením je tedy grafická nastavba aplikace. Grafický nástroj pro sestavování dopravních sítí, který by generoval XML soubor pro simulaci, by rozhodně práci zpříjemnil a vizualizace výstupu by omezila chybu lidského faktoru při vyhodnocování dat. Zobrazován by pak mohl být samotný průběh simulace a to třeba i v 3D pohledu.

## Kapitola 5

# Závěr

Tak, jak roste provoz na pozemních komunikacích, roste i potřeba tento provoz řídit a to co nejbezpečněji a nejefektivněji. K dosažení tohoto cíle přispívají dopravní simulace. Před tím, než se začne stavět nová křižovatka, měl by být její provoz nasimulován. Plány se dají změnit, ale jakmile je stavba postavena, náprava je dosti obtížná a nákladná.

V teoretické části práce jsem popsal obecné metody a přístupy, které se užívají v dopravních simulacích. Dále jsem nastínil, jak popsat chování vozidla v systému pomocí řídicích modelů, tedy Car-Following, Line-Changing a Gap-Acceptance modelu. Na základě těchto znalostí jsem navrhl městský dopravní simulátor pro sledování hustoty a plynulosti dopravy se zaměřením na výskyt kolon. Simulátor je založen na metodě mikro simulace, což dovoluje nejen rozlišovat typy vozidel, ale také rozlišovat jednotlivá vozidla. Jejich chování v systému ovlivňují nejen parametry konstrukce vozidla, ale parametrizovat lze také chování řidiče. Simulátor je zasazen do prostředí dopravní sítě České republiky a akceptuje u nás platná pravidla silničního provozu.

V současné době je implementováno jádro simulátoru, základní segmenty dopravní sítě (malé křižovatky, rovné úseky, zatáčky) a tři přednastavené typy vozidel (osobní automobil, nákladní automobil, autobus). Provoz na křižovatkách není řízen a vozidla jimi projíždí na základě pravidla přednosti zprava, nebo jsou přednosti řízeny dopravními značkami.

Problém ale je s validací modelu, neboť se mi nepodařilo získat potřebná data o reálném dopravním systému, na kterém by mohla být provedena. Pro alespoň částečné ověření jsem namodeloval dopravní úsek města Znojma, ulici Pražskou, který velmi dobře znám. Podle provedených experimentů je model funkční a dává uspokojivé výsledky. Jeho přesnost však je možné nadále zvyšovat, například podrobnějším zkoumáním chování řidičů v provozu.

Vývoj simulátoru nadále pokračuje a v nejbližší době bych se chtěl zaměřit zejména na rozšíření jeho použitelnosti. V první řadě to bude implementace semaforů, dále pak nadefinování dalších dopravních segmentů a typů vozidel. V budoucnu by definice typů vozidel mohla být ze simulátoru vyčleněna a přesunuta do konfigurace struktury dopravní sítě (XML záznamy). Ke zpříjemnění a zefektivnění práce se simulátorem by určitě přispěla také grafická nástavba.

# Literatura

- [1] Ahmed, K. I.: *Modeling Drivers' Acceleration and Lane Changing Behavior*. Dizertační práce, Massachusetts Institute of Technology, Massachusetts, únor 1999.
- [2] Ehlert, P.: *Intelligent Driving Agents*. Diplomová práce, Delft University of Technology, Delft, Leden 2001.
- [3] Jiménez, T.; Mussi, P.; Siegel, G.: A Road Traffic Simulator: Car-Following and Lane-Changing. In *Proceedings of the 14th European Simulation Multiconference on Simulation and Modelling: Enablers for a Better Quality of Life*, INRIA, Sophia Antipolis: SCS Europe, květen 2000, ISBN 1-56555-204-0, s. 241–245.
- [4] Kotál, R.: Pravidla provozu na pozemních komunikacích: Aktuální znění zákona a prováděcích předpisů. [online], [duben 2008].  
URL <http://www.autoskoly.cz/psp/>
- [5] Nagel, K.; Stretz, P.; Pieck, M.; aj.: TRANSIMS traffic flow characteristics. Technická zpráva, Los Alamos National Laboratory, Los Alamos, červen 1998.
- [6] Peringer, P.: *Modelování a simulace IMS: Studijní opora*. Fakulta informačních technologií VUT v Brně, Brno, listopad 2006.
- [7] Peringer, P.: SIMLIB: Simulation Library for C++. [online], [duben 2008].  
URL <http://www.fit.vutbr.cz/~peringer/SIMLIB/>
- [8] Schulze, T.; Fliess, T.: Urban Traffic Simulation with Psycho-Physical Vehicle-Following Models. In *Winter Simulation Conference*, Drážďany: Technical University Dresden, Prosinec 1997, ISBN 0-7803-4278-X, s. 1222–1229.
- [9] Thomason, L.: TinyXML Project. [online], [duben 2008].  
URL <http://www.grinninglizard.com/tinyxml/>
- [10] ČTK: MOV: Ovzduší v Pekingu je pro některé sporty riskantní. [online], březen 2008.  
URL <http://www.sport.cz/clanek/126798-mov-ovzdusi-v-pekingu-je-pro-nektere-sporty-riskantni.html>
- [11] Wikipedia: Computer Simulation. [online], [cit. 28.4.2008].  
URL [http://en.wikipedia.org/wiki/Computer\\_simulation](http://en.wikipedia.org/wiki/Computer_simulation)
- [12] Wikipedia: Simulation. [online], [cit. 28.4.2008].  
URL <http://en.wikipedia.org/wiki/Simulation>

# Příloha A

## Uživatelská příručka

Simulátor se spouští z příkazové řádky a jeho chování lze nastavit pomocí parametrů programu. Mezi povinné parametry patří jméno souboru s uloženou strukturou dopravní sítě a doba, po kterou bude simulace probíhat. Nepovinnými parametry lze definovat výstupy. Pokud je simulátor spuštěn s jediným parametrem `-h`, je zobrazena nápověda ke spuštění. Podrobněji v následujícím přehledu:

Povinné parametry:

- `<xml_file>` - jméno XML souboru se strukturou dopravní sítě a charakteristikou vozidel
- `<simulation_time>` - doba, po kterou bude simulace probíhat. Čas je ve formátu hh:mm a jeho maximální hodnota je omezena na 10000 hodin.

Nepovinné parametry:

- `-a <segment_name>` - jméno detailně sledovaného dopravního segmentu, který bude v každém kroku simulace vykreslen na výstup v ASCII módu.
- `-v <mode>` - nastavuje mód, ve kterém se budou vypisovat informace o vstupu/výstupu vozidel do/ze systému.
  - `-v start` - vypisuje informace o vygenerování nových vozidla do sítě.
  - `-v end` - vypisuje informace o tom, že vozidlo dorazilo na hranici dopravní sítě.
  - `-v all` - spojuje funkce obou předešlých módů.
  - pokud není mód definován, nevypisují se žádné informace o vozidlech.

**Příklad:** `trafim sit.xml 1:30 -a hlavni -v end`

Na konci simulace jsou vždy zobrazena data o průjezdu vozidel jednotlivými segmenty. Ta se liší v závislosti na typu segmentu:

- *Statistika průjezdu rovným úsekem nebo zatáčkou.* Zobrazí:
  - statistiku doby průjezdu daným segmentem
  - statistiku průměrné rychlosti, kterou vozidlo při průjezdu dosáhlo

```

+-----+
| Segment name: PRAZSKA 1           Segment type: ROAD2 |
+-----+
| Segment time: (s)                  |
+-----+
| Min = 3.3                          Max = 7.4          |
| Number of records = 396            |
| Average value = 4.7                |
| Standard deviation = 1.2           |
+-----+
| Average speed: (km/h)              |
+-----+
| Min = 22.4                         Max = 50.2         |
| Number of records = 396            |
| Average value = 38.5               |
+-----+

```

- *Statistika průjezdu křižovatkou.* Pro každý z příjezdů (3 nebo 4 v závislosti na typu křižovatkou) zobrazí:

- statistiku doby přiblížení, tedy doby kterou vozidlo potřebuje, aby projelo křižovatkou
- statistiku délky kolon, pokud se v daném směru nějaké vyskytly

```

+-----+
| Segment name: PRAZSKA-HUSITSKA    Segment type: T-ROAD |
+-----+
| Direction 1:                       |
+-----+
| Approaching time: (s)              |
+-----+
| Min = 3.4                          Max = 6.4          |
| Number of records = 193            |
| Average value = 3.6                |
| Standard deviation = 0.3           |
+-----+
| Tailback length: (m)               |
+-----+
| Min = 2                            Max = 4            |
| Number of records = 28             |
| Average value = 3.1                |
+-----+

```

```

+-----+
| Direction 2:                                     |
+-----+
| Approaching time: (s)                           |
+-----+
| Min = 5.3                                     Max = 12.5   |
| Number of records = 24                         |
| Average value = 6.9                             |
+-----+
| Tailback length: (m)                           |
+-----+
| Min = 2                                       Max = 6       |
| Number of records = 79                         |
| Average value = 3.7                             |
+-----+
+-----+
| Direction 3:                                     |
+-----+
| Approaching time: (s)                           |
+-----+
| Min = 3.5                                     Max = 4.3   |
| Number of records = 222                         |
| Average value = 3.5                             |
| Standard deviation = 0.1                         |
+-----+
| Tailback length: (m)                           |
+-----+
| no record                                       |
+-----+

```

Další výpisy jsou volitelné a aktivují se zadáním příslušných parametrů programu.

- *Detailní sledování dopravního segmentu* – parametr `-a <segment_name>`

Při detailním sledování je u segmentu daného jména po provedení každého kroku simulace vytisknut aktuální čas simulace a zobrazena mapa segmentu v ASCII módu. Na mapě jsou vyznačeny hlavní silnice (:), vedlejší silnice (.) a jednotlivá vozidla označená podle směru, kterým daný segment projedou – rovně (S), doprava (R) nebo doleva (L). Vozidla, která přejíždí do jiného pruhu jsou označena opět dle směru (rr, ll). Za každým zobrazením mapy segmentu je vytištěno návěští MORE. Pokud je výstup simulace přeměřován do souboru, je poté možno výstupní soubor krokovat právě podle návěští MORE a emulovat tak triviální animaci pohybu vozidel v segmentu.



Když vozidlo dorazí na hranici dopravní sítě, je zobrazen aktuální čas simulace, identifikace vozidla (časem vygenerování vozidla do sítě) a místo, kde vozidlo dosáhlo hranice dopravní sítě.

```
Time: 0:00:21.1 - vehicle(0:00:00.0) reached network border at socket 2 of segment PRAZSKA 4.
```

Výstupy simulace mohou mít, zejména při použití volitelných výstupů, rozsáhlejší charakter, proto je vhodnější je přeměrovat do souboru a vyhodnocovat je až po ukončení simulace.

Struktura dopravní sítě a popis chování vozidel je uložen v XML souboru s využitím následujících tagů:

- **<Network>** - definice dopravní sítě
- **<RoutePlans>** - seznam plánů tras vozidel
- **<RoutePlan>** - samotný plán trasy
  - atribut **number=""** - jedinečné číselné označení trasy
  - **<Item>** - položka určující úsek trasy – jeden dopravní segment
    - \* atribut **segname** - jméno segmentu
    - \* atribut **direction** - směr, kterým vozidlo daný segment projede – odbočí doprava, doleva nebo projede rovně
- **<Segments>** - seznam segmentů dopravní sítě
- **<Segment>** - jeden dopravní segment
  - atribut **name** - jedinečné jméno segmentu, maximální možná délka je 16 znaků
  - atribut **type** - typ segmentu
  - atributy **gradientX** - udává sklon vozovky v procentech (např. 10% znamená, že na 100m délky vozovky je výškový rozdíl 10m). U rovných úseků a zatáček má celý segment stejný sklon (**gradient1**), kdežto v případě křižovatek může mít každá větev jiný (**gradient1**, **gradient2**, **gradient3**, popř. **gradient4**). Kladná hodnota značí stoupání, záporná klesání z pohledu příjezdu ke křižovatce. U rovných úseků se za směrodatný považuje pohled od spojovacího bodu 1.
- **<Generator>** - generátor vozidel připadající dopravnímu segmentu
  - atribut **socket** - určení spojovacího bodu (strany) segmentu, kam se budou vozidla generovat
  - atribut **interval** - interval (v sekundách) ve kterém se budou vozidla generovat
  - atribut **start** (nepovinný) - simulační čas (v sekundách), ve kterém se má generátor poprvé aktivovat. Pokud není uveden nebo je jeho hodnota 0, bude generovat od začátku simulace.



- atribut **end** (nepovinný) - simulační čas (v sekundách), do kterého bude generátor aktivní. Pokud není uveden nebo je jeho hodnota 0, bude generovat do konce simulace.
- **<VehicleParams>** - procentuální zastoupení jednotlivých typů vozidel – osobní automobil, nákladní automobil a autobus
  - \* **<Item value="">** - procentuální hodnota
- **<SkillParams>** - procentuální zastoupení řidičů dle schopností – 5 položek od nejhorších řidičů až po ty nejlepší
  - \* **<Item value="">** - procentuální hodnota
- **<AggressiveParams>** - procentuální zastoupení řidičů dle agresivity – 5 položek od neklidnějších řidičů až po ty nejagresivnější
  - \* **<Item value="">** - procentuální hodnota
- **<RoutePlanParams>** - procentuální zastoupení tras pro vozidla. Uvádí se pouze trasy, které budou použity v tomto generátoru.
  - \* **<Item number="" value="">** - procentuální výskyt trasy s číslem **number**
- **<Connections>** - seznam spojení dopravních segmentů
- **<Connection>** (nepovinný) - propojí segmenty **<First>** a **<Second>**
  - **<First segname="" socket="" />** - jméno prvního segmentu a číslo spojovacího bodu
  - **<Second segname="" socket="" />** - jméno druhého segmentu a číslo spojovacího bodu

# Příloha B

## Instalace

### B.1 Linux

1. Nejprve je potřeba přeložit a na nainstalovat simulační knihovnu SIMLIB/C++. Zdrojové soubory jsou na CD v adresáři `/linux/src/simlib/` a k přeložení bude použit soubor `Makefile.Linux`.
2. Pak je teprve možno přeložit simulátor, jehož zdrojové soubory včetně `Makefile` jsou v adresáři `/linux/src/`.
3. Vytvoří se tak spustitelný soubor `trafim`.

### B.2 Windows

1. Nejprve je potřeba přeložit simulační knihovnu SIMLIB/C++. Zdrojové soubory jsou na CD v adresáři `/win32/src/simlib/` a k přeložení bude použit soubor `Makefile.Win`.
2. Přeloženou knihovnu a hlavičkový soubor (soubory `simlib.so` a `simlib.h`) je potřeba nakopírovat ke zdrojovým souborům simulátoru, které jsou v adresáři `/win32/src/`. Ty se pak pomocí přiloženého `Makefile` souboru přeloží.
3. Vytvoří se tak spustitelný soubor `trafim.exe`.

## Příloha C

# Obsah přiloženého CD

```
+--doc
|   +--html (generovaná dokumentace programem Doyxgen)
|   +--src  (zdrojové kódy dokumentace)
|   +--text (text práce v elektronické podobě)
|
|
+--linux
|   +--bin (přeložený simulátor a příklady dopravních sítí v XML)
|   +--src (zdrojové kódy simulátoru a potřebných knihoven)
|   |
|   +--readme (popis instalace simulátoru operačním systému Linux)
|   +--test_prazska.sh, testy_uviznuti.sh (ukázkové experimenty)
|   +--výstupy ukázkových experiemntů
|
|
+--win32
|   +--bin (přeložený simulátor a příklady dopravních sítí v XML)
|   +--src (zdrojové kódy simulátoru a potřebných knihoven)
|   |
|   +--readme (popis instalace simulátoru operačním systému Windows)
|   +--test_prazska.bat, testy_uviznuti.bat (ukázkové experimenty)
|   +--výstupy ukázkových experiemntů
|
|
+--readme
```