

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

**Možnosti řešení dopravních problémů pomocí nástrojů
systémové dynamiky**
Diplomová práce

Autor: Bc. Marek Zanker
Studijní obor: IM2p

Vedoucí práce: doc. Ing. Vladimír Bureš, MBA, Ph.D.

Hradec Králové

duben 2018

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 26.4.2018

Marek Zanker

Poděkování:

Děkuji vedoucímu diplomové práce doc. Ing. Vladimíru Burešovi MBA, Ph.D. za metodické vedení práce, za jeho podporu a mnoho cenných rad.

Anotace

Tato práce se zabývá systémovou dynamikou a je zaměřena na dynamické modely dopravních sítí. Teoretická část je věnována systémovému myšlení, systémové dynamice a nástrojům systémové dynamiky. Základní cíl diplomové práce je aplikace nástrojů systémové dynamiky na řešení dopravních problémů.

Praktická část je zaměřena na model samotný. Nejprve jsou představeny společné proměnné, z nichž vycházející typy křižovatek. Za zmínku stojí možné nastavení modelu, které je demonstrováno na nastavení zelené vlny. Funkčnost modelu a popis reality lze považovat za jeden z hlavních výsledků. Zajímavostí je i nevídaný způsob řešení modelu, kde je struktura postavena unikátním způsobem. Při modelování se vyskytly určité problémy. Hlavní limitací v modelování byla nemožnost kruhové závislosti. Tato limitace měla zásadní dopad na model jako takový.

Klíčová slova: Systémová dynamika, systémové myšlení, Systém, Simulace, STELLA, model, pole, doprava, křižovatka.

Annotation

Title: Possible solutions of traffic problems with the help of system dynamics tools

This diploma thesis deals with system dynamics and focuses on dynamic models of transport networks. The theoretical part is devoted to system thinking, system dynamics and system dynamics tools. The main goal of the diploma thesis is the application of system dynamics tools to solve traffic problems.

The practical part is focused on the model itself. First, common variables are introduced, based on types of intersections. It is worth mentioning the possible setting of the model, which is demonstrated in the green wave setting. Functionality of the model and description of reality can be considered as one of the main results. An interesting thing is a way of solving the model, where the structure is built in a unique way. There were some problems with modeling. The main limitation in modeling was the impossibility of circular dependence. This limitation had a major impact on the model as such.

Keywords: System dynamics, system thinking, system, simulation, STELLA, model, array, traffic, crossroads.

Obsah

| | | |
|-------|--|----|
| 1 | Úvod..... | 1 |
| 2 | Teoretická část | 2 |
| 2.1 | Systémové myšlení | 2 |
| 2.1.1 | Definice systémového myšlení..... | 2 |
| 2.1.2 | Části systémové myšlení..... | 4 |
| 2.1.3 | Přístupy | 5 |
| 2.2 | Systémová dynamika | 7 |
| 2.2.1 | System..... | 7 |
| 2.2.2 | Historie systémové dynamiky | 9 |
| 2.2.3 | Definice systémové dynamiky | 10 |
| 2.2.4 | Nástroje systémové dynamiky | 11 |
| 2.2.5 | Simulace | 21 |
| 3 | Cíle práce a metodika..... | 24 |
| 3.1 | Cíle práce | 24 |
| 3.2 | Metodika | 24 |
| 4 | Praktická část..... | 27 |
| 4.1 | Společné proměnné | 27 |
| 4.1.1 | Základní proměnné | 27 |
| 4.1.2 | Popisné proměnné | 31 |
| 4.2 | Typy křižovatek | 33 |
| 4.2.1 | Křížová křižovatka | 33 |
| 4.2.2 | Dopravní uzel ve tvaru písmena T | 37 |
| 4.2.3 | Kruhový objezd | 39 |
| 4.3 | Propojení městského objezdu | 43 |

| | | |
|-------|---|----|
| 4.4 | Simulace | 44 |
| 4.4.1 | Parametry simulace | 44 |
| 4.4.2 | Výsledky simulace | 44 |
| 4.5 | Zelená vlna | 48 |
| 5 | Shrnutí výsledků | 50 |
| 6 | Problémy | 51 |
| 6.1 | Nemožnost kruhové vazby | 51 |
| 6.2 | Vracení nenulových čísel z funkce „HISTORY“ | 52 |
| 6.3 | Zaokrouhlovací mechanismus..... | 53 |
| 7 | Závěry a doporučení..... | 54 |
| 8 | Seznam použité literatury..... | 55 |

Seznam obrázků

| | |
|--|----|
| Obrázek 1 - Prostředí software STELLA (obrázek autora)..... | 12 |
| Obrázek 2 - Vlastnosti proměnné (obrázek autora) | 14 |
| Obrázek 3 - Nástroje interface (obrázek autora) | 15 |
| Obrázek 4 - Ladič (obrázek autora)..... | 15 |
| Obrázek 5 - Editor pole (obrázek autora) | 16 |
| Obrázek 6 - Vlastnosti proměnné, záložka pole (obrázek autora) | 17 |
| Obrázek 7 - Hlavní struktura (převzato (STERMAN, 2002))..... | 18 |
| Obrázek 8 - Hladina (obrázek autora)..... | 18 |
| Obrázek 9 - Tok (obrázek autora)..... | 18 |
| Obrázek 10 - Regulátor (obrázek autora) | 18 |
| Obrázek 11 - Zdroj nebo Odtok (obrázek autora)..... | 19 |
| Obrázek 12 - Hlavní struktura 2 (obrázek autora)..... | 19 |
| Obrázek 13 - Import dat (obrázek autora) | 20 |
| Obrázek 14 - Export dat (obrázek autora)..... | 21 |
| Obrázek 15 Možnosti hodnot proměnné (převzato (BARLAS, 2007))...... | 23 |
| Obrázek 16 - Celkový vstup (obrázek autora)..... | 32 |
| Obrázek 17 - Celkový výstup a celkový vstup (obrázek autora)..... | 32 |
| Obrázek 18 - Dopravní zácpa (obrázek autora)..... | 33 |
| Obrázek 19 - Dopravní uzel, prvotní návrh (obrázek autora)..... | 34 |
| Obrázek 20 - Dopravní uzel ve tvaru kříže (obrázek autora)..... | 35 |
| Obrázek 21 - Vstup do dopravního uzlu ve stavu kříže (obrázek autora)..... | 35 |
| Obrázek 22 - Výstup z dopravního uzle ve tvaru kříže (obrázek autora)..... | 37 |
| Obrázek 23 - Vstup z externího systému (obrázek autora) | 37 |
| Obrázek 24 - Dopravní uzel ve tvaru písmena T (obrázek autora)..... | 38 |
| Obrázek 25 - Kruhový dopravní uzel (obrázek autora)..... | 39 |
| Obrázek 26 - Vstup do dopravního kruhového uzlu (obrázek autora)..... | 40 |
| Obrázek 27 - Vnější okruh (obrázek autora) | 43 |
| Obrázek 28 Graf - Celkový přehled modulu (obrázek autora)..... | 45 |
| Obrázek 29 Graf - přehled dopravních zácp (obrázek autora) | 46 |
| Obrázek 30 Graf - Nedostatek propustnosti 1 (obrázek autora) | 47 |

Obrázek 31 Graf - Nedostatek propustnosti 2 (obrázek autora)47

1 Úvod

V současnosti téma systémového myšlení a systémové dynamiky není v České republice příliš zmiňováno.

Systémové myšlení tak lze považovat za velmi vzácný způsob k porozumění systému jako celku, ale i jeho částem. Jedná se o způsob, který klade důraz na systém jako takový.

Systémová dynamika se řadí mezi relativně mladé vědní disciplíny. Využití nachází při predikování vývoje dynamických systémů, a také při zkoumání vazeb v systému. Uplatnění systémové dynamiky lze nalézt napříč systémy, od systémů přírodních přes systémy technické až po systémy sociální.

Modelování systémů umožňuje zkoumat vazby mezi elementy systému a zpětnovazebními smyčkami. Díky sestavování struktury systému je možné zkoumat a pozorovat systémy bez samotného zásahu do nich. Simulace umožňuje porozumět chování samotných elementů systému a také jejich působení na okolí a sebe sama. Simulaci lze rozdělit na simulaci spojitou a diskrétní.

Mezi zásadní software pro tvorbu systémové dynamiky patří STELLA, Vensim, Powersim a NetLogo. Každý z těchto programů má své vhodné využití pro různé druhy systémů. Nelze jednoznačně zhodnotit, který z těchto programů lze považovat za nejvhodnější.

Diplomová práce je zaměřena na modelování a simulaci dopravních systémů ve zmíněném programu STELLA, jelikož tato část systémové dynamiky ještě nebyla příliš zkoumána. Zásadním problémem dopravních systémů jsou dopravní kolapsy, které by mělo být možno simulovat právě za pomoci zmíněného programu a systémové dynamiky.

2 Teoretická část

Teoretická část je rozdělena na dvě podkapitoly: systémové myšlení a systémová dynamika. Podkapitola Systémové myšlení je věnována definici systémového myšlení a také přístupům autorů, kteří se danou problematikou zabývali. V následující podkapitole Systémová dynamika je popsán systém jako takový, také historie systémové dynamiky, definice systémové dynamiky a nástroje systémové dynamiky a simulace.

2.1 Systémové myšlení

Na systémové myšlení začal být kladen důraz již během dvacátého století. Bureš ve své knize uvádí, že ve dvacátých letech dvacátého století se začalo zmiňovat a využívat systémové myšlení v několika vědních oborech zároveň a to nezávisle na sobě. Systémy se začaly zkoumat do takové hloubky, že už nebylo možné využívat jen přístupy mechanistické, mechanické, redukcionistické a atomistické. Na místo těchto přístupů nebo řekněme ruku v ruce s nimi, se začaly využívat přístupy holistické, organistické, systémové a ekologické (BUREŠ, 2006).

Senge uvedl důvod pro používání systémového myšlení následovně: *„For systems theorists, the source of poor performance and organizational failure is often to be found in the limited cognitive skills and capabilities of individuals compared to the complexity of the systems they are called upon to manage.“* (SENGE, 1992).

Tímto výrokem naznačuje nekompatibilitu mezi komplexními systémy a zastaralým způsobem myšlení. V zásadě Senge naznačuje, že není možné pozorovat a pochopit složité systémy za využití klasických způsobů. Senge také uvádí důsledky, které tato inkonzistence může způsobit.

2.1.1 Definice systémového myšlení

Pojem „systémové myšlení“ nemá přesnou definici. Prozatím totiž nebyla nalezena shoda pro jasné vymezení systémového myšlení. Jediné, v čem se autoři shodují, je propojení systémové dynamiky se systémovým myšlením. Na druhou stranu už dle autorů není možné určit průnik systémového myšlení se systémovou dynamikou.

Sám zakladatel systémové dynamiky Jay W. Forrester ve volném překladu napsal: pojem systémové myšlení nemá žádnou čistou definici ani využití. (FORRESTER, 1994).

Barry Richmond naopak vidí systémové myšlení jako pojem nadřazený systémové dynamice. Richmond uvádí, že systémové myšlení je systémová dynamika s auru (RICHMOND, 1994).

Obě citace vyšly ve stejném roce, což ukazuje zásadní rozdíl mezi vnímáním systémového myšlení a systémové dynamiky mezi zakladatelem a průkopníkem v aplikaci tohoto vědního oboru.

Dalším výrazným autorem na poli systémového myšlení je Sterman. Ten shrnuje stav systémového myšlení takto: Existuje mnoho pohledů na systémové myšlení, některé zdůrazňují kvalitativní metody, jiné zase zdůrazňují formální modelování (STERMAN, 2000). Sterman dále upozorňuje na diverzitu oborů, kde má systémové myšlení využití. Například uvádí antropologii, biologii, inženýrství, lingvistiku, psychologii, fyziku a mnoho dalších oborů. Od roku 2000, kdy Sterman uvedl, že systémové myšlení je ve světě vzácné, se mnoho poznatků změnilo. Systémové myšlení se postupem času velmi rozšířilo. Jeho vzácnost však zůstává důležitým pozitivem.

Poměrně zajímavým způsobem popsal systémové myšlení Fan: Systémové myšlení je hlavní částí všech simulací systémové dynamiky, je jádrem popisu sil a vzájemných vztahů činností, které představují hlavní chování vyšetřovaného systému (FAN, 2018).

Fan považuje systémové myšlení za podstatnou schopnost a způsob, jak popsat hlavní vzájemné vztahy mezi elementy v systému. Tímto přispívá k různorodosti názorů na systémové myšlení. Lze konstatovat, že se Fan přiklání spíše na Richmondovu stranu, než na stranu Forrestera. Ovšem pořád lze říci, že definovat systémové myšlení není a asi nebude možné.

2.1.2 Části systémové myšlení

Vzhledem k tomu, že nebylo dosaženo shody ohledně systémového myšlení, nejsou sjednoceny ani přístupy používané v této problematice. Opět je možné říci, že různí autoři uvádějí různé myšlenky. Už se ale nerozcházejí tolik jako u definice.

Vojtko vymezil 7 základních dovedností systémového myšlení (převzato (VOJTKO, 2005).):

- **Dynamické myšlení** – zaměřuje se na rozpoznání na první pohled skrytých vzorů chování (spojování jednotlivých událostí v čase).
- **Pohled z 10 kilometrů** – situace se vždy jinak jeví z nadhledu a jinak, jste-li v ní zakomponováni (pouze z nadhledu ovšem můžete spatřit celistvost a některé důležité souvislosti).
- **Systém jako příčina** – na rozdíl od hledání viníků vně systému (Změníte-li správně způsob fungování systému, pak problém vyřešíte. Pokud vyměníte lidi, ale systém zůstane, problém se obvykle trvale nevyřeší).
- **Operační myšlení** – jde o uchopení systému z hlediska jeho infrastruktury, fungování prvků a vzájemných vazeb mezi nimi i v čase (snaha o poznání důležitých toků, akumulací a vztahu mezi nimi – vytváření odpovídajícího modelu či mapy problému).
- **Myšlení v uzavřených smyčkách** – zdůraznění vlivu a chování zpětnovazebních struktur (řízení, regulace, růst) a jejich specifikace (přizpůsobení systému změn vs. změna fungování systému).
- **Nelineární myšlení** – akce a reakce nemusejí být přímo úměrné, důvodem jsou především zpětné vazby (nap. kompenzační působení), zpoždění a defenzivní reakce systému.
- **Vědecké myšlení** – spočívá v kvantifikaci, důsledném definování a testování hypotéz, které vedou k vylepšování stávajících znalostí.

Ossimitz na rozdíl od Vojtko definoval čtyři základní prvky systémového myšlení. Ve třech prvcích se Ossimitz s Vojtkem shoduje a to v myšlení v modelech, myšlení v uzavřených smyčkách a v dynamickém myšlení. Odlišným prvkem je

řízení systému. Ossimitz tento prvek popisuje jako schopnost prakticky řídit systémy. (OSSIMITZ, 2000).

Zajímavým způsobem popsal systémové myšlení Doležal: „Systémové myšlení je způsob nazírání, které dává přednost celkovému pohledu na důležité aspekty projektu, a to s respektováním všech významných souvislostí.“ (DOLEŽAL, 2009)

Doležal tímto vyzdvihuje nutnost vidět systém jako celek, nikoliv jen jako jeho části. Tímto tvrzením poukazuje na nutnost využívání systémového přístupu.

2.1.3 Přístupy

Pohled na systém lze vymežit dvěma různými přístupy. První přístup se nazývá mechanistický a druhý systémový. Nedá se říci, že by jeden přístup byl vhodnější než druhý. Jediné, co lze konstatovat je skutečnost, že jeden přístup je více používaný než druhý.

Mechanistický přístup

Mechanistický přístup je charakterizován dvěma základními principy, redukcionismem a mechanismem.

„Podstata redukcionismu spočívá v tom, že každý jev, předmět nebo sebesložitější předmět může být redukován, rozložen nebo rozebrán na základní dále nedělitelné prvky.“ (BUREŠ, 2006)

Pomocí redukcionismu se dá rozložit cokoli na dále nedělitelné prvky. S využitím tohoto rozdělení lze poté popsat celek.

Bureš uvádí, že fundamentálním problémem této metody je úzká množina využitelnosti. Pro správné využití redukcionismu je řešení problémů s nízkou závislostí mezi sebou, nejlépe vůbec žádnou nebo alespoň nepodstatnou (BUREŠ, 2011).

Mechanismus je založen na názoru, že veškeré chování a poznávání může být zcela vysvětleno pomocí kauzálně interagujících mechanických procesů (ZELENKA, 2005).

Mechanismus využívá především kauzalitu. Problém s mechanismem nastává při špatné interpretaci výsledků. Většinou tato chyba nastává při silné korelaci na sobě nezávislých jevů.

Systémový přístup

Podle Bureše je systémový přístup výsledkem nové vědeckotechnické cesty, kde je redukcionismus a mechanismus nahrazen novými přístupy. V polovině dvacátého století bylo zapotřebí zkoumat a používat nové přístupy. Dlouhodobě bylo už téměř neúnosné zkoumat nové technologie a trendy původními přístupy. V té době se začaly rozvíjet počítačové technologie, vodíkové bomby a také začalo dobývání vesmíru. Pro každou z těchto vědeckých otázek již nebylo možné využívat starý způsob přístupu k jejich zkoumání (BUREŠ, 2009).

Za předchůdce systémového přístupu je považován množinový a celostní přístup.

Množinový přístup popisuje Janíček následovně: *„U tohoto přístupu jsou entity chápány pouze jako množiny prvků, což znamená, že vazby mezi prvky se neuvažují. Množinový přístup byl velkým pokrokem oproti předcházejícímu stavu, protože entity již nebyly chápány jako nerozdělitelné celky, ale byly členěny (dekomponovány) na jednotlivé prvky.“* (JANÍČEK, 2013).

Celostní přístup Janíček zaznamenal jako: *„Šokem pro vědu v první polovině minulého století bylo zjištění, že na chování složitých entit (tedy těch, které se skládají z většího množství prvků) nelze usuzovat na základě izolovaných analýz jejich prvků. Tedy řečeno opačně: na vlastnosti a chování složitých entit je možné usuzovat v kontextu většího celku, tvořeného prvky a vazbami mezi nimi (proto celostní přístup). To znamenalo, že pro chování složitých entit jsou stejně podstatné jak prvky entity, tak i vazby mezi nimi. Je to problematika „holismu“, který soustavám připisuje takové vlastnosti celku, jež nejsou odvoditelné pouze z jeho izolovaných prvků.“* (JANÍČEK, 2013).

Bureš vymazuje tři hlavní úrovně systémového přístupu:

- Zřetel k celku – zajišťuje, že všechny sledované předměty, jevy a procesy jsou součástí větších celků – vše je nutno hodnotit v širší souvislosti.
- Systematické hledání cílů.
- Nový syntetický způsob myšlení – syntetický přístup podává vysvětlení vlastností prvků z hlediska jejich funkce a vazeb v systému a vysvětlení jejich chování z hlediska chování celého systému.

Jinými slovy lze popsat podstatu systémového přístupu jako nutnost zkoumat celý systém, nikoliv jen jeho části. Důležité je předpokládat, že části systému na sebe mohou působit a vzájemně se mohou ovlivňovat.

2.2 Systémová dynamika

Před zaměřením se na systémovou dynamiku je vhodné nejdříve nastínit, co to systém je. Pro pochopení celé problematiky je následně zapotřebí popsat poznatky historie systémové dynamiky, definice systémové dynamiky, nástroje systémové dynamiky a simulace.

2.2.1 Systém

Obecnou definici popsal Bureš následovně: „*Systém je komplex prvků spolu se vztahy mezi nimi a mezi jejich atributy*“ (BUREŠ, 2011).

Je důležité zdůraznit, že obecnou definici systému lze rozšířit o konkrétnější definice systému.

Konstrukcí definice systému se zabývá obecná teorie systémů. Výsledkem této teorie jsou tři teoretické formy postupu. Vlček tyto tři formy popsal následovně (VLČEK, 1999):

- Axiomaticky na základě apriorních nepochybných primitivních (atomických) předpokladů. Axiomy představují jev (fenomén) či jeho části. Definice vzniká kombinací axiomů podle předem daných pravidel. Nabývá formálního matematického tvaru. Z atomických předpokladů jsou formou vět odvozovány složky definovaného jevu. Postup je označován jako deduktivní.
- Experimentálně na základě pozorování či měření. Vytvořená definice představuje ověřené pozorování či měření. Může se týkat i části jevu (parciální ověření) a celková definice (např. druhu) vzniká dodatečně. Postup je označován jako induktivní.
- Gnoseologicky na základě souvislého postupu či prohloubení poznání jevu (fenoménu). Metodikou gnoseologického přístupu ke konstrukci definice je kladení otázek. Postup je často označován jako postup podle společenské zkušenosti.

Jak Vlček uvádí, tyto tři základní postupy se mohou vzájemně doplňovat a podporovat.

Další možnosti definice systémů uvádí Bureš. Vymezil pět základních definic, které jsou lépe využitelné v praxi. Do této pětice patří definice pomocí vnějších veličin, definice pomocí dané aktivity, definice pomocí chování, definice pomocí prvků a vazeb a definice pomocí stavově přechodné struktury (BUREŠ, 2011).

Bureš také poukazuje na rozdíl mezi systémem a hromadou. Díky tomuto rozdělení lze lépe porozumět tomu, co systém opravdu je a co systémem už není. Bureš vymezuje pět základních parametrů pro rozpoznání hromady od systému. Jedná se o propojenost částí a jejich celkovou funkčnost, dělitelnost částí, důležitost uspořádání, důležitost každé části a důležitost struktury (BUREŠ, 2011).

Systémy jsou rozdělovány do několika skupin podle různých parametrů.

Jedno z možných rozdělení rozlišuje systémy podle organizace. Pelánek dělí systémy na jednoduché, neorganizované a komplexní (PELÁNEK, 2011).

Další rozdělení systému uvedl Bureš. Vybral tři základní atributy, podle kterých lze dělit systémy: vazba k okolí, vztah k realitě a chování v čase nebo původu systému. Následně popsal systémy podle těchto atributů (BUREŠ, 2011):

- otevřené a uzavřené,
- abstraktní a konkrétní,
- statické a dynamické,
- adaptivní a neadaptivní,
- deterministické a stochastické,
- přirozené a umělé
- hmotné, nehmotné a informační,
- měkké a tvrdé.

Fontes popsal systémovou dynamiku následujícím způsobem (FONTES, 2018):. Systémová dynamika je považována silnou metodu pro popis a analýzu vztahů mezi faktory v komplexních systémech, pro simulaci chování systému v průběhu času.

Fontes touto teorií vymezuje hlavní přínosy pro využití systémové dynamiky. Z popisu je patrné, že použití systémové dynamiky má hlavní opodstatnění pro

složitější systémy. Dalším poznatkem z výše uvedené citace je využívání simulace pro zkoumání systému v průběhu času.

Drmola vysvětluje systémovou dynamiku jako (DRMOLA, 2018): Systémová dynamika je metodologie a technika matematického modelování k popisu, porozumění a analyzování komplexních problémů a otázek. Zároveň systémová dynamika umožňuje navrhnout možná řešení a vést o těchto řešeních diskuzi.

Drmola popisuje systémovou dynamiku pod odlišným úhlem pohledu než Fontes. Zajímavé je, že Drmola vyzdvihuje problémy a otázky, zatímco Fontes považuje za rozhodující analýzu vztahů a vazeb. Oba autoři se shodují ve využití systémové dynamiky pro komplexní systémy.

2.2.2 Historie systémové dynamiky

Jak už bylo uvedeno výše, zakladatel systémové dynamiky je Jay W. Forrester, který působil na MIT Sloan School of Management. Systémová dynamika se začala vyvíjet v padesátých letech minulého století. Prvotně byl pro systémovou dynamiku používán název „Industrial dynamic“ v překladu průmyslová dynamika (JANIŠOVÁ, 2013).

Základ průmyslové dynamiky vznikl na půdě společnosti General Electric. Organizace byla v minulosti spojována s problémem velkých výkyvů pracovních míst. Jednalo se o prudké změny poptávky práce. Výkyvy byly natolik velké, že v jednom roce měla General Electric čtyřsměnný provoz a během následujícího roku musela propustit polovinu zaměstnanců pro nadbytečnost. General Electric hledala pro tento problém řešení, které následně našel Jay W. Forrester. Byl vytvořen program SIMPLE („Simulation of Industrial Management Problems with Lots of Equations“), který je považován za první program pro zkoumání průmyslové dynamiky.

V šedesátých letech se principy průmyslové dynamiky začaly rozšiřovat do měst. Tento proces dal vzniknout urbánní dynamice (v originále Urban Dynamics). Postupem času se začaly principy průmyslové a urbánní dynamiky rozšiřovat do ostatních systémů. Následkem tohoto vývoje vznikla systémová dynamika (v originále System Dynamics).

Velkým přínosem pro systémovou dynamiku byl rozvoj informačních technologií. V tento okamžik bylo možné zkoumat systémy pomocí modelů s výpočtem rovnic a využití informačních technologií. Tento krok přinesl pozitivní posun díky snížení časových a finančních nákladů na zkoumání systémů. Postupem času se začal objevovat software přímo zaměřený na tvorbu modelů systémové dynamiky například: Dynamo, Vensim, STELLA, Powersim.

Od sedmdesátých let se modely systémové dynamiky běžně využívaly a to především ve Spojených státech amerických. Rozptyl využití systémové dynamiky byl enormní. Systémová dynamika pak našla své využití od jednoduchých systémů až po velmi složité a rozsáhlé. (BUREŠ, 2008).

2.2.3 Definice systémové dynamiky

Sterman popisuje spojitost mezi systémovou dynamikou a leteckým simulátorem.

Systémová dynamika je metoda za pomoci, které lze dosáhnout pochopení komplexních systémů. Stejně jako letecké společnosti používají letecké simulátory pro učení pilotů létat, tak je systémová dynamika částečně metodou pro vývoj managementu leteckých simulátorů. Často počítačové simulace umožňují nahlédnout do komplexní dynamiky. Pochopit zdroj neefektivnosti pravidel a navrhnout nová a více efektivní pravidla. (STERMAN, 2000).

Sterman zde poukazuje na využitelnost systémové dynamiky, kterou připodobňuje k leteckému simulátoru. Vyzdvihuje zde určité vlastnosti systémové dynamiky. Jako nejdůležitější část vidí úsporu času a financí a případně zkoumání systému bez nutného zásahu do něj.

Castellacci vymezuje vlastnosti modelů systémové dynamiky následovně (CASTELLACCI, 2017): Modely systémové dynamiky jsou skvěle postaveny pro zkoumání co-evolučních procesů. Tuto vlastnost získávají díky jejich zaměření na komplexní soustavu zpětnovazebních mechanismů anebo kauzálních smyček. Toto popisuje interakce mezi relevantními proměnnými v systému.

Forrester považuje za hlavní vlastnost dynamických systémů průchod časem a prostorem. Za největší problém dynamických systémů pak Forrester vidí jejich

samostatnou realizaci, kde je nutné určit správnou míru abstrakce tak, aby odpovídala požadavkům modelu. (FORRESTER, 1994).

2.2.4 Nástroje systémové dynamiky

„Hladiny a toky jsou stavební prvky modelů systémové dynamiky.“ (FORD, 1991). Tímto výrokem Ford charakterizoval základní strukturu modelů systémové dynamiky. Mezi programy na tvorbu modelů systémové dynamiky patří například STELLA, Vensim, Powersim a NetLogo.

Podrobněji je v práci popsán software od společnosti isee systems. Jedná o verzi STELLA 10.0.

Historie společnosti isee systems

Roku 1985 Barry Richmond založil společnost isee systems za účelem vyvíjení software pro tvorbu modelů systémové dynamiky. O čtyři roky později isee systems dostala cenu Jay Wright Forrester. Tato cena byla udělena za vytvoření prvního interaktivního simulačního prostředí systémové dynamiky. (ISEE SYSTEMS).

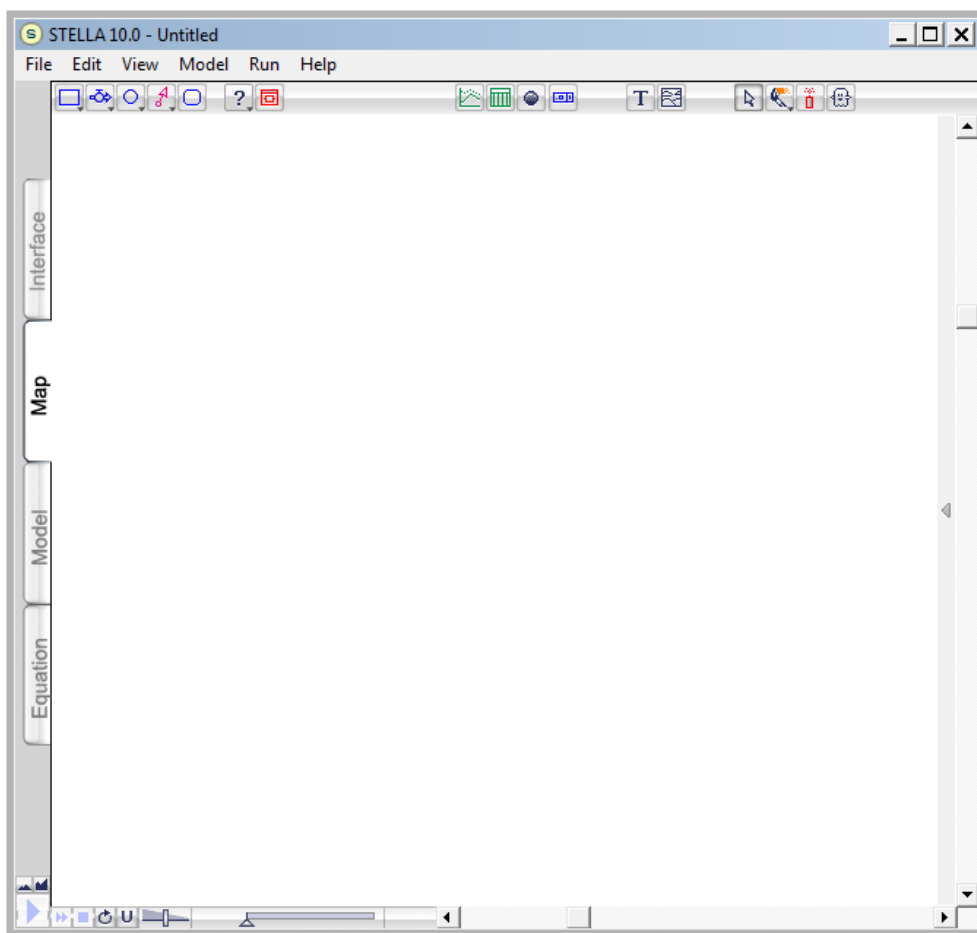
Produkty

Mezi produkty společnosti isee systems patří především software. Příkladem je STELLA 10.0, nebo Stella Professional. Další produkty by se daly označit jako podpora pro software a systémové myšlení. Do této kategorie patří výuková videa, odborné články, knihy a tutoriály.

STELLA 10.0

V programu STELLA 10.0 není implementována podpora pro český jazyk. Využití diakritiky vede k různým chybám od ztráty dat až po nefunkčnost modelu. Z tohoto důvodu je vhodné nepoužívat diakritiku, anebo používat čistě anglickou terminologii.

Prostředí programu STELLA 10.0 můžeme rozdělit na několik základních částí, prostředí je vidět na obrázku 1. První z nich je levá postranní lišta, kde se nacházejí čtyři záložky rozhraní (Interface), mapa (Map), model (Model), rovnice (Equation).



Obrázek 1 - Prostředí software STELLA (obrázek autora)

Mapa (Map) slouží pro vytváření návrhu modelu. Účelem této části je vytvořit základní stavební kameny modelu. Model se v Mapě vytváří pomocí hladin (Stock), toků (Flow), konvektorů (Converter) a propojovačů akcí (Action Connector). Samostatnou částí svislé horní lišty je modul (Modul), tento prvek umožňuje vytvoření „modelu v modelu“. V mapě nelze vytvářet matematické rovnice, toto vychází z premisy návrhu modelu.

Následující pole je pro tvorbu grafu. V grafu je možno zobrazit až 5 proměnných z modelu. Další možnou funkcí je tabulka, která může zachytit více proměnných než graf. Tabulka znázorňuje podrobnější tvrdá data z modelu, ale na druhou stranu nedokáže informace graficky vizualizovat jako graf.

Další neméně důležitou funkcí je textové pole. Tato funkcionality má za úkol graficky podpořit přehlednost a znovupoužitelnost modelu. Textové pole by mělo být využíváno pro popis částí modelu.

Štětec (Paintbrush) umožňuje změnit barvu elementů modelu. Tato možnost byla implementována z důvodu přehlednosti modelu. Dynamit (Dynamite) slouží pro odstraňování elementů v modelu.

Poslední část je duch (Ghost). Duch umožňuje zrcadlení proměnných a slouží tak pro umístování jedné proměnné na více míst v modelu. Využitelnost tohoto prvku je v zpřehlednění modelu a propojování modulů mezi sebou.

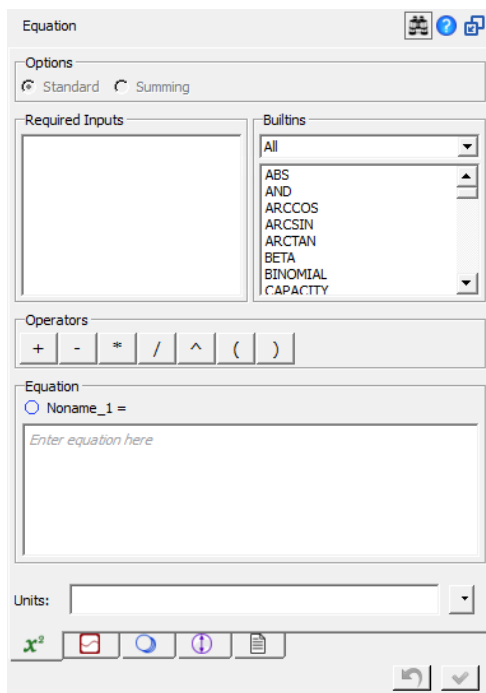
Model

Model slouží pro vytváření matematických vztahů mezi proměnnými. V záložce modelu je vodorovná lišta totožná s vodorovnou lištou v záložce mapy.

Zásadní rozdíl mezi záložkou modelu a záložkou mapy je v možnostech práce s proměnnými. Po otevření vlastností proměnné se zobrazí okno s možnostmi proměnné. Na rozdíl od map je zde vložena možnost dokumentace.

První rozšíření v části modelu oproti části mapy je položka „Equation“ (dále už jen položka rovnice). Okno položky rovnice se dá rozdělit na pět základních částí a to: „Required Inputs“, „Builtins“, „Operators“, „Equation“ a „Units“. Okno rovnic je vidět na obrázku 2.

Část „Required Inputs“ sdružuje proměnné, které jsou propojeny pomocí „Action Connector“. Jedná se o proměnné, které jsou logicky spojeny s danou proměnnou. Část „Builtins“ je výčet všech vestavěných funkcí v software. V části operátorů (v originále „Operators“) se nachází základní matematické znaky, které slouží pro tvorbu rovnic (v originále „Equation“). Za hlavní část obrázku se dá považovat „Equation“. V tomto úseku se vytváří matematická logika proměnné. Poslední věc, která je vidět na obrázku 2 je „Units“. Jednotky slouží pro určení druhu částic procházejících proměnnými.



Obrázek 2 - Vlastnosti proměnné (obrázek autora)

Mimo položky rovnice se ve vlastnostech proměnné nachází položka s grafem (v originále „Graphical Function“). Položka s grafy umožňuje nastavit hodnoty proměnné pro běh simulace za pomoci grafu. Díky této utilitě je možno nadefinovat různé hodnoty pro různé časové úseky simulace.

Další položkou je část pro práci s poli. Zde je možno proměnnou nastavit jako proměnnou s polem, vybrat předdefinované pole, nebo jej odebrat.

Předposlední položka se nazývá „Scales and Ranges“. Zde je možné upravovat název proměnné, nastavovat její obor hodnot nebo upravovat formát hodnot.

Poslední část je označována jako „Documentation“ a je stejná jako v mapě. Tato část přebírá hodnoty vložené v mapě.

„Equation“

Záložka „rovnice“ (v originále „Equation“ dále, už jen „záložka rovnice“) vyobrazuje veškeré rovnice z celého modelu. Jedná se o matematické vyjádření celého modelu.

Rovnice jsou logicky sdruženy do bloků. Tyto bloky se dělí podle modulů. Každý modul je dále dělen na hladiny a každá hladina má přiřazené přítoky a odtoky. Po vyjádření všech hladin nastává prostor pro matematické vyjádření všech konvektorů.

Interface

Interface se nachází v software STELLA 10.0 na prvním místě vertikální lišty, ale vhodnější bude tuto utilitu představit jako poslední. Interface má hlavní význam pro představení modelu anebo pro představení výsledků simulace.

Horizontální lišta je v části interface do jisté míry shodná s horizontální lištou z modelu. Lišta je zachycena na obrázku 3.



Obrázek 3 - Nástroje interface (obrázek autora)

Část s otazníkem se nazývá tlačítko (v originále Button). Tlačítko umožňuje různé aktivity od informačního charakteru přes navigace až po spuštění story tellingu.

Následuje odkaz na přepínač (Switch). Za pomoci přepínače lze vypínat/zapínat různé části modelu.

Jak je patrné z obrázku 4, za pomoci ladiče (Knob Input Device) lze nastavovat hodnoty pro proměnné. Pro tuto utilitu je typický velký rozsah desetinných hodnot.



Obrázek 4 - Ladič (obrázek autora)

Následující utilitou je list (List Input Device). List sdružuje proměnné a jejich hodnoty. Proměnné v listu je vhodné sdružovat pomocí logických návazností. Za pomocí listů lze upravovat nastavení modelu na požadovaný stav.

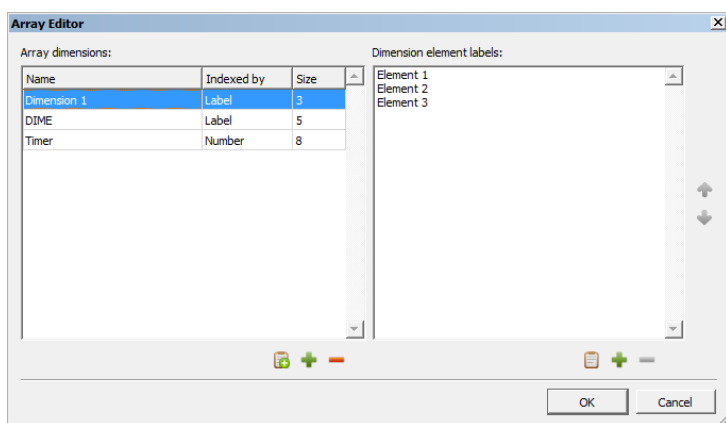
Poslední odlišnou utilitou je posuvník (Slider). Posuvník funguje na stejném principu jako ladič.

Pole

Pomocí funkce pole lze rozšiřovat pole proměnných na různé velikosti. Pole lze definovat následujícím způsobem: Pole $x[n]$ (KORN, 2013), kde x značí pole a n značí velikost pole. Další možností je vytvoření matice, kde lze Pole definovat jako Pole $A[n,m]$ (KORN, 2013), kde A je název matice, n počet sloupců a m počet řádků. Třetí možností je třídimenzionální pole, které lze popsat jako Pole $x[n],A[n, m]$ (KORN, 2013). Program STELLA umožňuje přidávání dimenzí dle potřeby. Je tedy možné se dostat až k x dimenzionálnímu poli.

Každá proměnná je určitým polem. Základní proměnná, tedy proměnná s velikostí pole se dá popsat jako Pole $x[1]$. Proměnnou je možné rozšířit na pole s větší hodnotou než jedna. Poté lze takovéto pole označit Pole $x[n]$, kde n může nabývat hodnot od jedné do nekonečna.

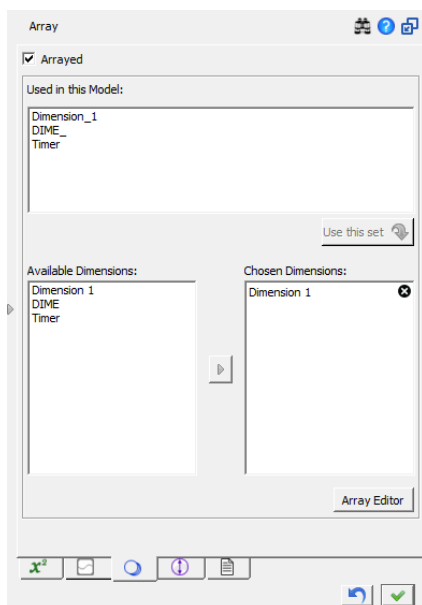
Software STELLA umožňuje vytvářet pole pomocí editorů polí (Array Editor), který je vidět na obrázku 5. Při vytváření polí jsou na výběr dva způsoby značení. Prvním způsobem je indexování. Alternativou je vytvoření názvů pro jednotlivá pole. Velikost pole se vybírá v části Velikost (Size), do které se zadá přirozené číslo větší než jedna. Software STELLA je vybavena možností kombinovat dimenze. Díky této utilitě je možno vytvářet 3D prostor, případně 4D prostor a takto lze pokračovat až do nekonečna.



Obrázek 5 - Editor pole (obrázek autora)

Rozšíření proměnné o pole je vidět na obrázku 6. V horní části je možné označit proměnnou jako proměnnou s polem. Po této operaci lze vybrat definovanou dostupnou dimenzi (v originále „Available Dimensions“). V pravé části jsou vidět vybrané dimenze (v originále „Chosen Dimensions“). Při výběru více než

jedné dostupné dimenze vzniká více dimenzionální prostor, jak už bylo popsáno výše.



Obrázek 6 - Vlastnosti proměnné, záložka pole (obrázek autora)

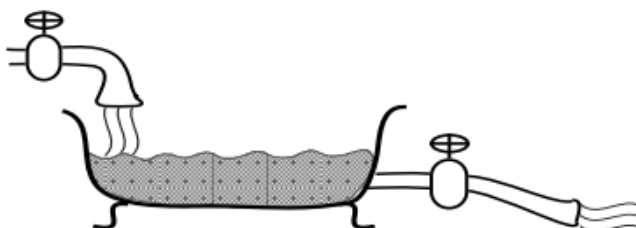
Pro práci s poli jsou v software STELLA implantovány vestavěné funkce (ISEE SYSTEMS):

- **MEAN** – vrací aritmetický průměr pole nebo výrazu. Syntaxe vestavěné proměnné je následující: „MEAN(<pole nebo výraz>)“.
- **RANK** – vrací číselný index z proměnné s polem, s daným pořadovým číslem. Syntaxe je následující: „RANK(<proměnná s polem>, <pořadové číslo>)“.
- **SELF** – využívá se pro odkazování na sebe samu. Používá se v kombinaci s vestavěnými proměnnými SIZE a PREVIOUS.
- **SIZE** – vrací velikost pole nebo výrazu. Syntaxe je následující: „SIZE(<pole nebo výraz >)“.
- **STDDEV** – vrací hodnotu standardní derivace elementů pro pole či výraz. Se syntaxí: „STDDEV(<pole nebo výraz >)“.
- **SUM** – vrací součet elementů pro pole či výraz. Syntaxe je následující: „SUM(<pole nebo výraz >)“.

Stavební prvky diagramu systémové dynamiky

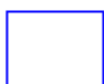
Sterman rozděluje stavební prvky diagramu systémové dynamiky na hladiny, přítoky, odtoky, regulátory a zdroje (STERMAN, 2000).

Na obrázku 7 Sterman popsal základní myšlenku diagramu systémové dynamiky. V levé horní části obrázku 7 je naznačen přítok. Střed obrázku 7 zobrazuje hladinu a pravá část znázorňuje odtok. Jak je z obrázku vidět přítok i odtok jsou vybaveny regulátory. Sterman krásně vystihl analogii diagramu systémové dynamiky s nádobou plnicí se tekutinou.



Obrázek 7 - Hlavní struktura (převzato (STERMAN, 2002))

V software STELLA je hladina reprezentována pomocí obdélníku, jak je vidět na obrázku 8. Obdélníkový tvar má připomínat kontejner, v originále se nazývá „Stock“ (v překladu kontejner).



Hladina

Obrázek 8 - Hladina (obrázek autora)

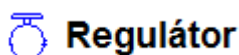
Program STELLA nerozlišuje odtok a přítok. Grafické vyjádření toku má připomínat potrubí, jak je vidět na obrázku 9. Toto znázornění vychází z reálné funkce potrubí, a to proudění látky ze zdroje do jiného uložení. Tok v software STELLA může mít dva různé zdroje. Za první zdroj se dá považovat zdroj na obrázku 11. Alternativou tohoto zdroje je Hladina.



Tok

Obrázek 9 - Tok (obrázek autora)

Regulátor má na starosti sílu toku. V software STELLA je regulátor implementován v toku. Na obrázku 10 je vidět samotný regulátor.



Regulátor

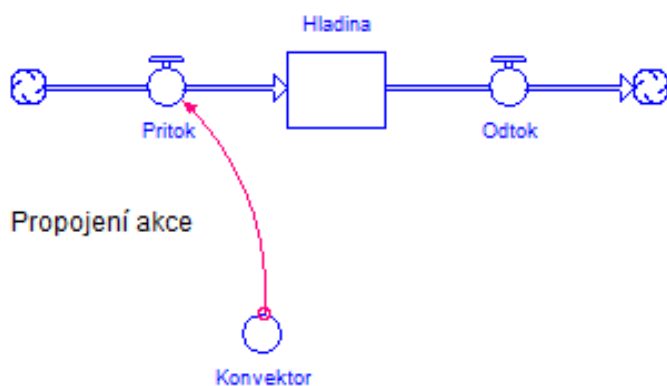
Obrázek 10 - Regulátor (obrázek autora)

Zdroj a odtok jsou v software STELLA značeny stejným grafickým obrazcem. Grafické vyobrazení zdroje a odtoku má připomínat mrak.

Zdroj nebo Odtok

Obrázek 11 - Zdroj nebo Odtok (obrázek autora)

Na obrázku 12 je vidět základní struktura diagramu v software STELLA. Konvektor a „Propojení akce“ nejsou na Stermanově analogii zobrazeny. Konvektor plní podpůrnou funkci pro regulátory. Účelem konvektoru je zjednodušovat rovnice v regulátorech. Zároveň lze jednu rovnici v konvektoru využít pro více regulátorů. Propojení akce slouží k logickému propojení mezi prvky v diagramu.



Obrázek 12 - Hlavní struktura 2 (obrázek autora)

Práce s daty

Software STELLA umožňuje různé operace s daty. Mezi hlavní operace patří import dat, export dat a prezentace dat.

Import dat

Importování dat z externích souborů je vhodné využívat například při testování modelu. Druhým důvodem může být převzetí reálných dat (například dat získaných z pozorování) a zkoumání funkcí modelu s reálnými daty.

Při testování se požadují různé druhy dat. Pomocí importů dat z externích souborů je možné vytvořit různé varianty dat. Může se jednat o data vhodná pouze pro začátek simulace, či o statická/dynamická data pro průběh celé simulace.

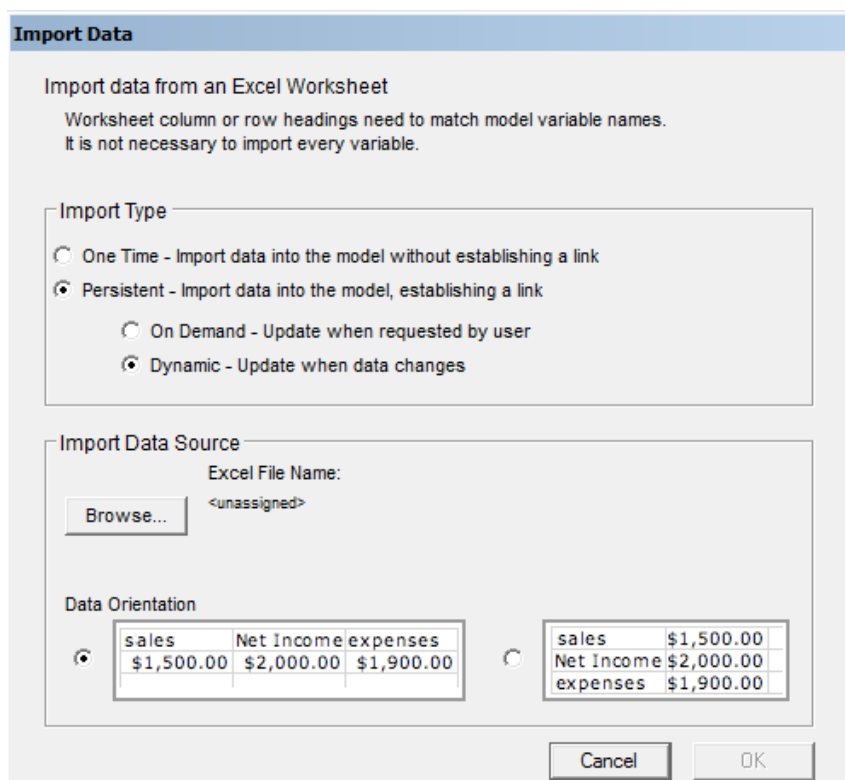
Importací reálných dat lze testovat model na relevantnost jeho výsledků. Pokud model projde testy tohoto druhu, lze jej využívat pro predikci situace. Další možností je zkoumání vztahů mezi elementy.

Na obrázku 13 je zachyceno okno pro importování dat.

Software STELLA umožňuje různé postupy při importaci dat. Tyto postupy je možné kategorizovat podle frekvence importování dat a uložení importovaných dat.

Frekvenci je možné použít při importaci dat pouze jednou anebo při kontinuální importaci dat. Pro kontinuální importaci lze vybrat z možností „na požádání“ a dynamické importace dat. Jak už názvy napovídají tak způsob „na požádání“ importuje data z externího souboru vždy, když o to uživatel požádá, zatímco dynamický způsob aktualizuje data vždy při změně dat.

Uložení dat pro importaci se dělí dle jejich uspořádání - respektive podle uspořádání proměnných v externím souboru. Proměnné mohou být seřazeny vertikálně nebo horizontálně.



Obrázek 13 - Import dat (obrázek autora)

Export dat

Export dat slouží pro další interpretace výsledků dosažených během simulace.

Frekvence exportu dat funguje v softwaru STELLA na stejném principu jako import a rozložení dat. Toto je patrné při porovnání obrázku 13 s obrázkem 14.

V exportu dat lze vybrat, které proměnné budou exportovány a v jakých intervalech. Interval lze zvolit ze tří možností: zaznamenat pouze hodnoty v jeden čas, zaznamenávat hodnoty pro každý x-tý čas, anebo pro každý DT modelu.

Export Data

Export Data to an Excel Worksheet

Export Type

- One Time - Export data from the model without establishing a link
- Persistent - Export data from the model, establishing a link
 - On Demand - Update when requested by user
 - Dynamic - Update when data changes

Export Data Source

- Export all model variables
- Export variables in table

Interval:

- One set of values
- Every Minutes/2
- Every DT - Export every intermediate value during the run
- Use table settings

Export Destination

Excel File Name:

<unassigned>

Data

- | sales | Net Income | expenses |
|------------|------------|------------|
| \$1,500.00 | \$2,000.00 | \$1,900.00 |
- | | |
|------------|------------|
| sales | \$1,500.00 |
| Net Income | \$2,000.00 |
| expenses | \$1,900.00 |

Obrázek 14 - Export dat (obrázek autora)

2.2.5 Simulace

„Hlavní rolí počítačové simulace v systémové dynamice je testování hypotéz o vztazích mezi strukturou systému a chováním systému.“ (HOVMAND, 2014).

„Simulace manipuluje s modelem v čase nebo prostoru, aby ho komprimovala a umožnila tak vnímat interakce, které by jinak nebyly zřejmé z důvodu jejich oddělení v čase či prostoru.“ SYSTEM THINKING – volně přeloženo

Je tedy možné říci, že simulace umožňuje zkoumat souvislosti mezi prvky modelu, které nemusejí být na první pohled zřejmé. Díky této vlastnosti lze simulací proniknout hlouběji do užších spojitostí v systému.

Simulace můžeme rozdělit na diskrétní a spojitě. Nelze jednoznačně zhodnotit, který druh simulace je vhodnější. Obě ze zmíněných simulací mají využití pro jiný druh modelu a následně pro požadovaný druh výsledku. Diskrétní simulace rozděluje chod modelu na určité časové úseky, které jsou předem stanoveny. Oproti tomu spojitě simulace tyto časové úseky co nejvíce zkracují. Teoreticky se velikost časového úseku při spojitě simulaci blíží nule (SYSTEMS-THINKING).

Duggan říká, že se simulace postupně rozdělila na dvě separovaná a distingovaná odvětví. Duggan rozděluje simulaci na dvě části: diskrétní simulaci (v originále discrete-event simulation (DES)) a spojitou simulaci (Duggan tyto simulace označuje jako přístupy systémové dynamiky (system dynamics approaches) (DUGGAN, 2006).

Ve své podstatě se nedá přesně určit hranice velikosti časového úseku mezi simulací diskrétní a simulací spojitou.

Barlas popisuje základní stavy proměnných v průběhu spojitě simulace. Průběh stavů proměnných je zobrazen na obrázku 15.

První možností je konstantní stav (první část obrázku 15) Tento stav nastává, pokud se hodnoty v proměnné během simulace nemění, tedy je její stav konstantní.

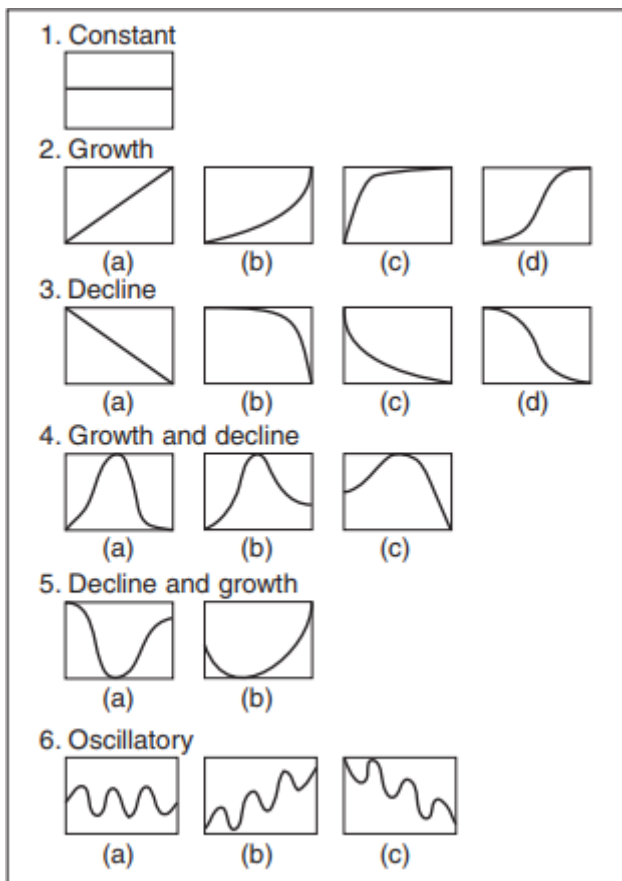
Za druhou možnost lze považovat růst hodnot proměnné (druhá část obrázku 15). Tento stav nastává, pokud v průběhu simulace hodnoty v proměnné rostou. V části d) je vidět stav, kdy hodnoty proměnné rostly a v jisté situaci přešly do konstantního stavu. Pokud tedy hodnoty proměnné alespoň v jeden okamžik simulace narůstaly a po dobu celé simulace nepoklesly, jedná se o proměnnou, kterou lze označit jako „proměnnou s hodnotami růstu“.

Třetí možností je klesání hodnot (třetí část obrázku 15). Jedná se zde o opak růstových hodnot.

Růst a pokles značí čtvrtou možnost. (čtvrtá část obrázku 15). Pro tento vývoj platí, že hodnoty proměnné musí během simulace jednou růst a zároveň alespoň jednou klesat. Další podmínkou je nutnost růstu jako první změny hodnot proměnné.

Pátou možnost umožňuje simulaci poklesu a růstu (pátá část obrázku 15). Pro tento vývoj platí první podmínka pro „růst a pokles“. Rozdíl je v druhé podmínce. Zde je nutné, aby hodnoty v průběhu simulace nejprve klesaly.

Poslední možností simulace je oscilace (šestá část obrázku 15). Oscilace je kombinace několika růstů a poklesů. V části 6a je vidět oscilace kolem konstantního čísla. V části 6b je vyobrazena oscilace s růstovou tendencí. V části 6c je znázorněna oscilace s klesající tendencí.



Obrázek 15 Možnosti hodnot proměnné (převzato (BARLAS, 2007)).

3 Cíle práce a metodika

V této kapitole bude přiblížen cíl práce, dílčí cíle a metodika práce.

3.1 Cíle práce

Hlavním cílem diplomové práce je aplikace nástrojů systémové dynamiky na řešení dopravních problémů. Cíl byl určen z důvodu možnosti aplikovat systémovou dynamiku na řešení dopravních problémů. V oblasti dopravy není systémová dynamika hojně využívána. Tento stav umožňuje poukázat na možnosti systémové dynamiky.

Dílčím cílem bylo vytvoření modelu systémové dynamiky zaměřeným na dopravu. Účelem tohoto cíle je naplnění cíle hlavního. Nejprve bylo nutné vytvořit funkční model systémové dynamiky, aby bylo možné zkoumat problémy v dopravě.

Důležitým bodem byla také znouvupoužitelnost modelu systémové dynamiky. V zásadě bylo vhodné situovat model systémové dynamiky způsobem, který bude snáze aplikovatelný na různá prostředí v dopravě.

Dalším dílčím cílem bylo rozšíření modelu systémové dynamiky o ukazatele, které mají spojitost s dopravními problémy. Tento dílčí cíl byl vybrán, z nutnosti popisu nekonzistentních stavů v dopravě vůči optimálním stavům.

Posledním cílem byla zvolena aplikace vytvořeného modelu na případovou studii. Následně by tak bylo možné poukázat na možnosti vytvořeného modelu.

3.2 Metodika

Metodika pro tvorbu modelu byla převzata od Sterman. Jedná se volný překlad (STERMAN, 2000).:

1. Vyjádření problému
 - Výběr tématu: Jaký je problém? A proč je to problém?
 - Klíčové proměnné: Co jsou klíčové proměnné? Jaké jsou klíčové koncepty?
 - Definice dynamického problému: Jak se klíčové části chovaly v minulosti.
2. Formulace dynamické hypotézy

- Prvotní hypotézy: Jaké jsou teorie o aktuálním problémovém chování?
 - Endogenní vliv.
 - Mapování: Vytvoření map pro aktuální hypotézy, klíčové proměnné a pro další přístupná data.
3. Sestavení simulačního modelu
- Specifikace struktury a rozhodovacích pravidel.
 - Odhady parametrů, vztahů chování a počátečních podmínek.
 - Testy.
4. Testování
- Porovnání s referenčním módem: Vytváří model problémové chování, které je adekvátní k účelu?
 - Robustnost v extrémních podmínkách: Je chování modelu realistické při extrémních podmínkách?
 - Citlivost: Jak je model citlivý na změnu parametrů.
5. Návrh postupu a hodnocení
- Specifikace scénáře: Jaké okolní podmínky se mohou objevit, a jsou zároveň schopné ovlivnit problém?
 - Navrh postupu: Jaká jsou nová rozhodovací pravidla, strategie a struktury, která by mohla být aplikována v reálném světě?
 - Co když analýza: Jaké jsou důsledky navržených postupů?
 - Citlivostní analýza: Jak robustní jsou doporučené postupy v různých scénářích?
 - Interakce postupů: Interagují postupy mezi sebou?

Některé části této metodiky budou mírně upraveny pro účel práce. Tato metodika je popsána jako obecná a proto je vhodné některé její části mírně poupravit, nebo je ponechat jen v teoretické rovině.

Coyle rozdělil metodiku do pěti částí (COYLE, 1996):

1. rozpoznání problému (koho to zajímá a proč),
2. porozumění problému a popis systému,
3. kvalitativní analýza,
4. simulace a modelování,
5. kontrola testování a návrhu.

Oba postupy se do určité míry prolínají. Postup Coyla je více obecný a není tedy příliš nutné vytvářet tolik úprav, jako u postupu Stermana. Na druhou stranu je Stermanův postup více podrobný. Právě kvůli podrobnosti byla pro diplomovou práci vybrána Stermanova metodika.

4 Praktická část

Praktická část práce se opírá o teoretické poznatky a metodiku. Jsou zde popsány proměnné a rovnice použité při modelování, samotný průběh simulace a problémy spojené s modelováním problematiky dopravních systémů. Dále jsou zde shrnuty výsledky modelování.

4.1 Společné proměnné

Společné proměnné lze rozdělit na proměnné základní a popisné. Základní proměnné zajišťují logiku modelu. Bez nich by simulace modelu nedávala smysl nebo by nebyla možná. Popisné proměnné plní funkci informativní. Jejich hlavní smysl je zakotven v popisu simulace.

4.1.1 Základní proměnné

Model je rozdělen na několik modulů. V každém z nich se nachází jeden dopravní uzel. Některé proměnné jsou společné pro všechny moduly. Jedná se například o proměnné, řídící čas, či výpočet dopravní zácpy.

Counter

Proměnná „Counter“ sčítá posledních osm vstupů do modulu z jedné strany. Funguje jako počítadlo dopravních prostředků, které přijely během určitého časového období.

Pro konvektor „Counter“ je typické, že řeší prvních sedm časových období zvlášť. Pro každé problematické časové období je tedy vytvořena vlastní část kódu.

Rovnice 1 vyhodnocuje, zdali součet aktuálního času a časové vzdálenosti mezi křižovatkami se rovná jedné. Pokud je tato podmínka splněna konvektor „Counter“ promítne hodnotu z navazujícího modulu z aktuálního času a odečte časovou vzdálenost. Pokud podmínka z Rovnice 1 není splněna, přejde „Counter“ na druhou část Rovnice 1 a tak dále. Po překonání problematického časového období se využívá poslední podmínka z Rovnice 1 po zbytek průběhu simulace.

Lze konstatovat, že prvních sedm podmínek v Rovnici 1 plní funkci stabilizace spuštění modelu a poslední podmínka nastupuje ve stavu, kdy simulace probíhá stabilně.

```

IF (TIME + Range_1) = 1 THEN
HISTORY(Circular_connection_2, TIME -Range_1)
ELSE
IF (TIME + Range_1) = 2 THEN
HISTORY(Circular_connection_2, TIME -Range_1) +
HISTORY(Circular_connection_2, TIME -1 -Range_1)
ELSE
IF (TIME + Range_1) = 3 THEN
HISTORY(Circular_connection_2, TIME -Range_1) +
HISTORY(Circular_connection_2, TIME -1 -Range_1) +
HISTORY(Circular_connection_2, TIME -2 -Range_1)
ELSE
IF (TIME + Range_1) = 4 THEN
HISTORY(Circular_connection_2, TIME -Range_1) +
HISTORY(Circular_connection_2, TIME -1 -Range_1) +
HISTORY(Circular_connection_2, TIME -2 -Range_1) +
HISTORY(Circular_connection_2, TIME -3 -Range_1)
ELSE
IF (TIME + Range_1) = 5 THEN
HISTORY(Circular_connection_2, TIME -Range_1) +
HISTORY(Circular_connection_2, TIME -1 -Range_1) +
HISTORY(Circular_connection_2, TIME -2 -Range_1) +
HISTORY(Circular_connection_2, TIME -3 -Range_1) +
HISTORY(Circular_connection_2, TIME -4 -Range_1)
ELSE
IF (TIME + Range_1) = 6 THEN
HISTORY(Circular_connection_2, TIME -Range_1) +
HISTORY(Circular_connection_2, TIME -1 -Range_1) +
HISTORY(Circular_connection_2, TIME -2 -Range_1) +
HISTORY(Circular_connection_2, TIME -3 -Range_1) +
HISTORY(Circular_connection_2, TIME -4 -Range_1) +
HISTORY(Circular_connection_2, TIME -5 -Range_1)
ELSE
IF (TIME + Range_1) = 7 THEN
HISTORY(Circular_connection_2, TIME -Range_1) +
HISTORY(Circular_connection_2, TIME -1 -Range_1) +
HISTORY(Circular_connection_2, TIME -2 -Range_1) +
HISTORY(Circular_connection_2, TIME -3 -Range_1) +
HISTORY(Circular_connection_2, TIME -4 -Range_1) +
HISTORY(Circular_connection_2, TIME -5 -Range_1) +
HISTORY(Circular_connection_2, TIME -6 -Range_1)
ELSE
IF (TIME + Range_1) > 7 THEN
HISTORY(Circular_connection_2, TIME -Range_1) +
HISTORY(Circular_connection_2, TIME -1 -Range_1) +
HISTORY(Circular_connection_2, TIME -2 -Range_1) +
HISTORY(Circular_connection_2, TIME -3 -Range_1) +

```

```

HISTORY(Circular_connection_2, TIME -4 -Range_1) +
HISTORY(Circular_connection_2, TIME -5 -Range_1) +
HISTORY(Circular_connection_2, TIME -6 -Range_1) +
HISTORY(Circular_connection_2, TIME -7 -Range_1)
ELSE 0

```

(1)

Changer

Konvektor „*Changer*“ slouží pro změnu časového rozložení. Tato proměnná odpovídá na otázku, jak moc konvektor „*Timer_X*“ pootočí s časovým rozložením dopravního uzlu. Využití těchto dvou konvektorů je více rozepsáno v kapitole 4.5 Zelená vlna.

Timer

„*Timer*“ určuje reakce světelné křižovatky. Každý vstupní tok reaguje na určitou dimenzi konvektoru „*Timer*“. „*Timer*“ je osmi-dimenzionální konvektor. Počet dimenzí vychází ze základní logiky modelu. V rovnici 2 je vidět, že proměnná „*Timer*“ může nabývat celé hodnoty od 0 do 7. Každá hodnota z této množiny ovlivňuje chování křižovatky.

```

Timer_1[1] = (1 + Changer) mod 8
Timer_1[2] = (2 + Changer) mod 8
Timer_1[3] = (3 + Changer) mod 8
Timer_1[4] = (4 + Changer) mod 8
Timer_1[5] = (5 + Changer) mod 8
Timer_1[6] = (6 + Changer) mod 8
Timer_1[7] = (7 + Changer) mod 8
Timer_1[8] = (0 + Changer) mod 8

```

(2)

Range

Konvektor „*Range*“ počítá časovou vzdálenost mezi křižovatkami. Časová vzdálenost je roznásobena hodnotou dva. Tato úprava byla nutná z důvodu rozdělení jedné časové jednotky, poloviny.

Proměnná „*Range*“ se využívá pro stanovení vzdálenosti. Tento atribut má blízkou souvislost s proměnnou „*Counter*“.

Permeability

„*Permeability*“ určuje, kolik dopravních prostředků je schopno projet dopravním uzlem během jednoho časového úseku. Tento konvektor plní funkci pro stanovení maximální propustnosti křižovatky.

Traffic_jam

Hladina „*Traffic_jam*“ značí počet dopravních prostředků v dopravní zácpě. Tato proměnná slouží pro jako „brána“ do dopravního uzlu.

Input_to_traffic_jam

Proměnná „*Input_to_traffic_jam*“ jak už název napovídá označuje přítok do proměnné „*Traffic_jam*“. Pro tok „*Input_to_traffic_jam*“ platí rovnice 3. Pokud časová hodnota modula osm bude shodná s hodnotou proměnné „*Timer*“, pak hodnota z konvektoru „*Counter*“ přiteče do hladiny „*Traffic_jam*“.

$$\text{ROUND (IF (TIME MOD 8 = Timer}_1[3]) \text{ THEN Counter}_1 \text{ ELSE 0)} \quad (3)$$

Sum_of_Output

Konvektor „*Sum_of_Output*“¹ sečte příslušné hodnoty z odtoku „*Output_X*“. Tento konvektor slouží pro propojení dvou závislých modulů. „*Sum_of_Output*“ se dá považovat za proměnnou, která má zajistit předávání zásadních informací mezi moduly. Toto provázání je vytvořeno pomocí zrcadlení proměnné. Zrcadlo proměnné „*Sum_of_Output*“ je vloženo do závislého modulu. V propojeném modulu je konvektor „*Sum_of_Output*“ propojen s proměnnou „*Counter*“.

Random_number

Konvektor „*Random_number_X*“ generuje pseudonáhodné číslo od hodnoty nula až po hodnotu jedna. Tento jev lze vidět na rovnici 3. Tato proměnná stanovuje logiku rozdělení dopravních prostředků.

$$\text{RANDOM}(0,1) \quad (3)$$

Generování pseudonáhodných čísel vyžaduje podrobnější zkoumání z důvodu chyby zaokrouhlených čísel. V modelu se uplatňují práva pro vygenerovaná pseudonáhodná čísla. Výsledné vygenerované číslo se nesmí rovnat hodnotě 0,5. Při hodnotě 0,5 v modelu vznikaly hodnoty, které neodpovídaly předpokladům. Platí zde následující: pokud je hodnota pseudonáhodného čísla rovna 0,5, korekce ji upraví na hodnotu 0,51. Z tohoto důvodu byla vytvořena proměnná „*Correction_of_Random_number_1*“.

¹ V kruhovém dopravním uzlu funkci této proměnné plní „*Output*“. Tento rozdíl vychází z nepotřeby sčítat výstupy v kruhovém dopravním uzlu.

Rovnice 4 zobrazuje základní úpravu. Rovnice 3 vygeneruje náhodné číslo od hodnoty nula až po hodnotu jedna. Rovnice 4 vyhodnotí, zdali je konvektor „*Random_number_1*“ roven 0,5. Pokud nastane tento stav, konvektor „*Correction_of_Random_number_1*“ určí svojí hodnotu jako 0,51. V opačném případě převezme hodnotu z proměnné „*Random_number_1*“.

```
Correction_of_Random_number_1 = IF Random_number_1 = 0.5 THEN 0.51 ELSE  
Random_number_1 (4)
```

Random_number_for_Input

Další částí modelu generující pseudonáhodná čísla je konvektor „*Random_number_for_Input_X*“. Tato proměnná generuje dopravní prostředky vstupující do systému z vnějších systémů.

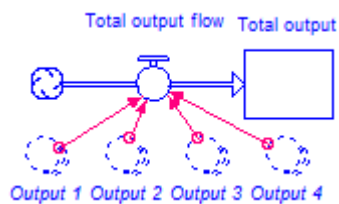
Tento konvektor je hybnou silou modelu a to z důvodu funkce naplnění modelu dopravními prostředky. Na začátku simulace se v modelu nenachází žádný dopravní prostředek. Během prvních kroků simulace se model začne plnit dopravními prostředky z těchto konvektorů. Tato proměnná může také plnit funkci testování zátěže systému - tedy zkoumat, jak se bude systém chovat při změně tlaku dopravních prostředků. Aby tuto funkci mohla plnit korektně, má nastaveno rozpětí pomocí dvou konvektorů zmíněných v rovnici 5.

```
RANDOM(Minimum_of_random_number_3,Maximum_of_random_number_3) (5)
```

Konvektor „*Maximum_of_random_number*“ značí maximální možný počet přítoků dopravních jednotek z jiných systémů. Tento konvektor spolu s konvektorem „*Minimum_of_random_number*“ koriguje počet dopravních prostředků vstupujících do systému. Proměnná „*Minimum_of_random_number*“ určuje minimální možný počet přítoků dopravních jednotek z jiných systémů.

4.1.2 Popisné proměnné

V modelu se nachází informační část, která vykresluje situaci probíhající v simulaci. Následující proměnné jsou společné pro všechny moduly.



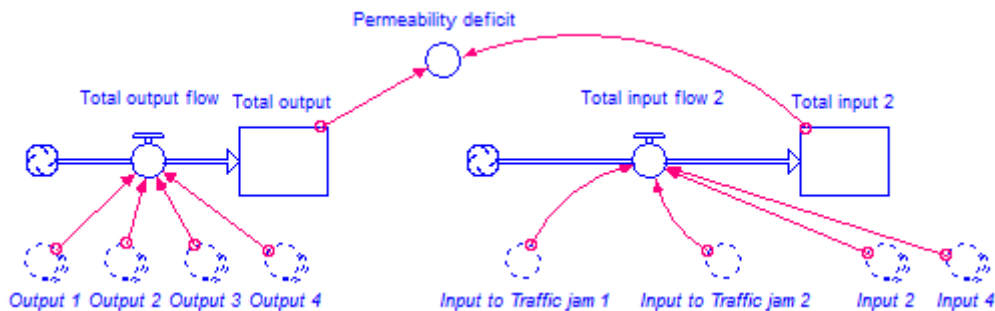
Obrázek 16 - Celkový vstup (obrázek autora)

$$\text{SUM}(\text{Input}_1)+\text{SUM}(\text{Input}_3)+\text{SUM}(\text{Input}_2)+\text{SUM}(\text{Input}_4) \quad (6)$$

Na obrázku 16 je zachyceno počítadlo vstupů do křižovatky. Jedná se součet vstupů z okolních systémů (těch, které nejsou v modelu zatížené hladinou „Traffic_jam_X“) a součet vstupů z dopravních kolon. Tok „Total_input_flow“ ukazuje přítok do křižovatky v každém čase, toto je vyjádřeno rovnicí 6. „Total_input“ určuje, jak velký byl přítok do dopravního uzlu od počátku simulace až po současný čas.

V levé části obrázku 17 je zachyceno počítadlo odtoků. Hladina „Total_output“ ukládá hodnotu celkového odtoku z modulu. Tok „Total_output_flow“ ukazuje odtok ze systému v každé časové jednotce. Tento výpočet je vyjádřen rovnicí 7.

V pravé části obrázku 17 je vidět počítadlo vstupů do modulu. Hladina „Total_input_2“ sčítá celkový přítok do modulu. Tok „Total_input_flow_2“ zobrazuje aktuální hodnotu přítoku do modulu, toto je vyjádřeno rovnicí 8.



Obrázek 17 - Celkový výstup a celkový vstup (obrázek autora)

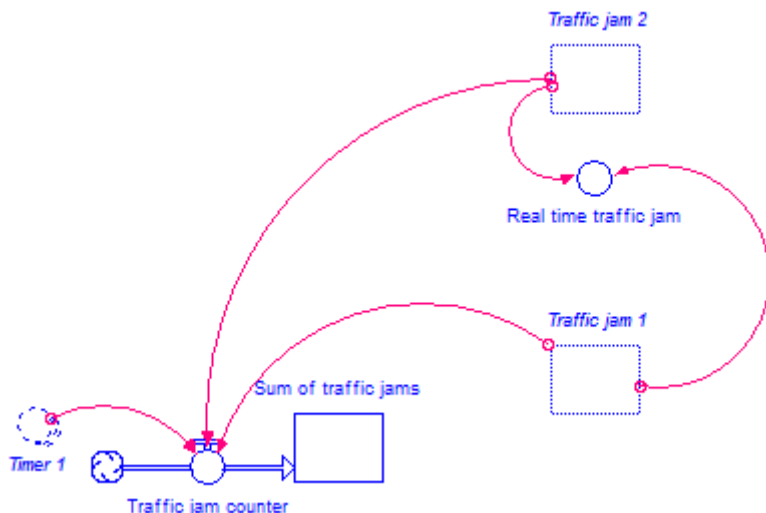
$$\text{SUM}(\text{Output}_1)+\text{SUM}(\text{Output}_2)+\text{SUM}(\text{Output}_3)+\text{SUM}(\text{Output}_4) \quad (7)$$

$$\text{Input_to_Traffic_jam_1}+\text{Input_to_Traffic_jam_2}+\text{SUM}(\text{Input}_3)+\text{SUM}(\text{Input}_4) \quad (8)$$

Konvektor „Permeability_deficit“ ukazuje nedostatečnou/dostatečnou propustnost dopravního uzlu. Pokud je hodnota proměnné „Permeability_deficit“ nulová jedná se o stav simulace, kdy je propustnost dostatečná. V případě, že hodnota konvektoru „Permeability_deficit“ je rovna jedné, nebo větší než jedna, je

propustnost nedostatečná. Rovnice proměnné „Permeability_deficit“ je vyobrazena rovnicí 9.

$$\text{Total_input_2} - \text{Total_output} \quad (9)$$



Obrázek 18 - Dopravní zácpa (obrázek autora)

Konvektor „Real_time_traffic_jam“ zobrazuje aktuální počet dopravních jednotek stojících v dopravní zácpe. Tato proměnná je vyjádřena rovnicí 10. Hladina „Sum_of_traffic_jams“ ukazuje celkový počet dopravních jednotek uvíznutých v dopravní zácpe. Přítok „Traffic_jam_counter“ je vyjádřen rovnicí 11. Jak je patrné z rovnice 11, tak informace o dopravních kolonách se aktualizují vždy v určitý čas.

$$\text{Traffic_jam_1} + \text{Traffic_jam_2} \quad (10)$$

$$\text{IF TIME MOD } 8 = \text{Timer_1}[3] \text{ THEN Traffic_jam_1 ELSE IF TIME MOD } 8 = \text{Timer_1}[2] \text{ THEN Traffic_jam_2 ELSE } 0 \quad (11)$$

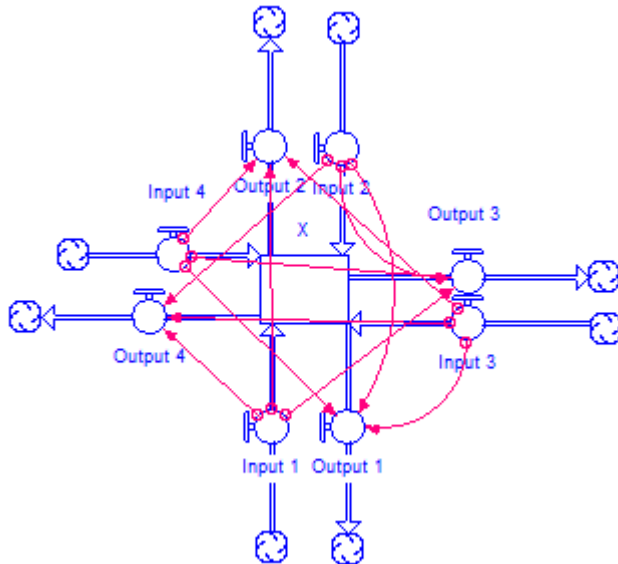
4.2 Typy křižovatek

V modelu se nacházejí tři základní typy křižovatek: křížová křižovatka, křižovatka do tvaru písmene T a Kruhový objezd.

4.2.1 Křížová křižovatka

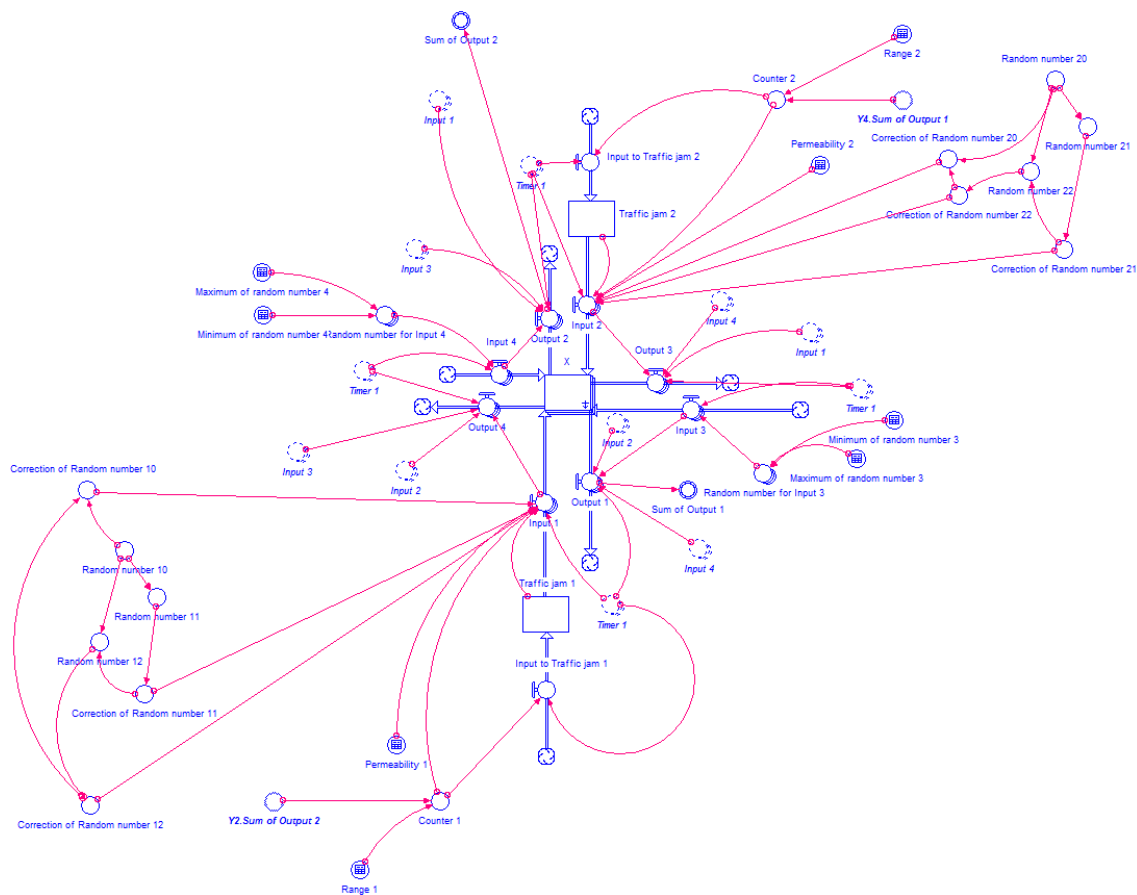
Prvním druhem dopravního uzlu je křižovatka ve tvaru kříže. Jedná se o dopravní uzel se čtyřmi vstupy a čtyřmi výstupy.

Na obrázku 19 je vidět základ pro sestavení křižovatky do tvaru kříže. Obrázek 19 odpovídá logickému uspořádání křižovatky. V této části se předpokládá, že veškeré vstupy/výstupy jsou výstupy/vstupy z/do jiných systémů.



Obrázek 19 - Dopravní uzel, prvotní návrh (obrázek autora)

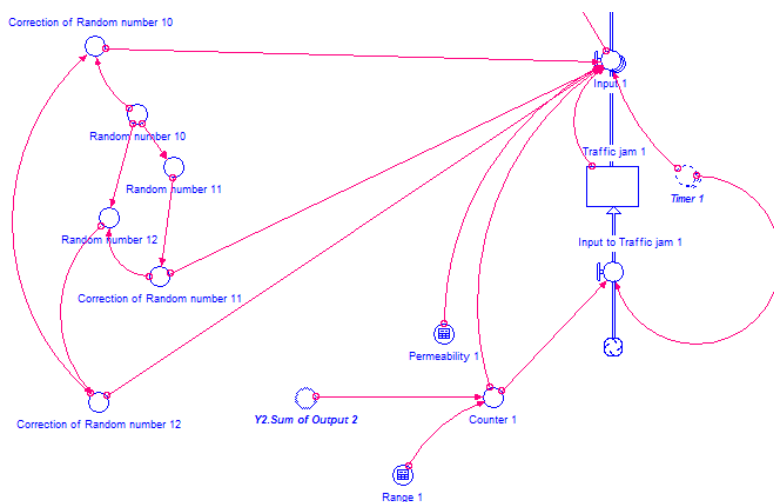
Pro každý „Input“ platí, že je přítokem do křižovatky. Analogicky platí pro každý „Output“, že je výstupem z křižovatky. Hladina „X“ funguje jako kontrolní mechanismus správného fungování modelu. Při žádoucí funkcionalitě modelu se předpokládá hodnota proměnné „X“ rovna nule.



Obrázek 20 - Dopravní uzel ve tvaru kříže (obrázek autora)

Na obrázku 20 je zobrazen celý mechanismus křížového dopravního uzlu. Střed obrázku okolo hladiny „X“ je velmi podobný původnímu návrhu, který je znázorněn na obrázku 19. Jednotlivé části jsou popsány níže. Pro pochopení daného modelu bylo potřeba model mechanicky rozdělit na části.

Vstup a dělení



Obrázek 21 - Vstup do dopravního uzlu ve stavu kříže (obrázek autora)

Na obrázku 21 jsou zachyceny mechanismy podporující rozhodování pro vstup do dopravního uzlu. V pravé části se nachází hladina „*Traffic_jam_1*“, v které jsou uskladněny dopravní jednotky, které nemohly projít křižovatkou. Konvektor „*Permeability*“ značí množství dopravních prostředků, které jsou schopny projít dopravním uzlem v jedné časové sekvenci. Proměnná „*Input_to_Traffic_jam_1*“ je vyjádřena rovnicí 12.

$$\text{ROUND} (\text{IF} (\text{TIME MOD } 8 = \text{Timer}_1[5]) \text{ THEN Counter}_1 \text{ ELSE } 0) \quad (12)$$

Přítok „*Input_X*“ se stará o průchod dopravních prostředků dopravním uzlem. Jak je evidentní z rovnice 13, rovnice 14 a rovnice 15, křižovatkou nemůže projít více dopravních prostředků, než je nastavena hodnota konvektoru „*Permeability*“. Dále rovnice udává, že pokud je součet hladiny „*Traffic_jam_1*“ a konvektoru „*Counter_X*“ větší než hodnota konvektoru „*Permeability*“, do křižovatkou projede jen počet dopravních prostředků roven konvektoru „*Permeability*“ a zbytek zůstane v hladině „*Traffic_jam_X*“. V druhé alternativě součet hladiny „*Traffic_jam_X*“ a konvektoru „*Counter_X*“ je roven nebo menší, než hodnota konvektoru „*Permeability*“, projdou křižovatkou všechny dopravní prostředky.

$$\begin{aligned} &\text{ROUND}(\\ &\text{IF} (\text{TIME MOD } 8 = \text{Timer}_1[5]) \text{ THEN} \\ &\text{IF} \text{Permeability}_1 > (\text{Traffic_jam}_1 + \text{Counter}_1) \text{ THEN} \\ &(\text{Traffic_jam}_1 + \text{Counter}_1) * \text{Correction_of_Random_number}_{10} \text{ ELSE} \\ &\text{Permeability}_1 * (\text{Correction_of_Random_number}_{10}) \text{ ELSE } 0 \end{aligned} \quad (13)$$

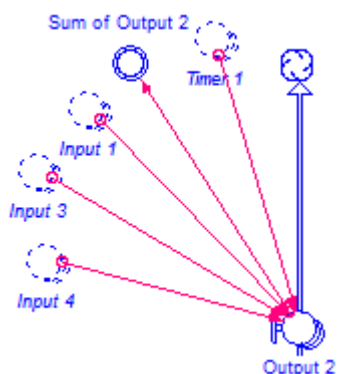
$$\begin{aligned} &\text{IF} (\text{TIME MOD } 8 = \text{Timer}_1[5]) \text{ THEN} \\ &\text{IF} \text{Permeability}_1 > (\text{Traffic_jam}_1 + \text{Counter}_1) \text{ THEN} \\ &(\text{Traffic_jam}_1 + \text{Counter}_1) * \text{Correction_of_Random_number}_{10} \text{ ELSE} \\ &\text{Permeability}_1 * (\text{Correction_of_Random_number}_{11}) \text{ ELSE } 0 \end{aligned} \quad (14)$$

$$\begin{aligned} &\text{IF} (\text{TIME MOD } 8 = \text{Timer}_1[5]) \text{ THEN} \\ &\text{IF} \text{Permeability}_1 > (\text{Traffic_jam}_1 + \text{Counter}_1) \text{ THEN} \\ &(\text{Traffic_jam}_1 + \text{Counter}_1) * \text{Correction_of_Random_number}_{10} \text{ ELSE} \\ &\text{Permeability}_1 * (\text{Correction_of_Random_number}_{12}) \text{ ELSE } 0 \end{aligned} \quad (15)$$

Výstup

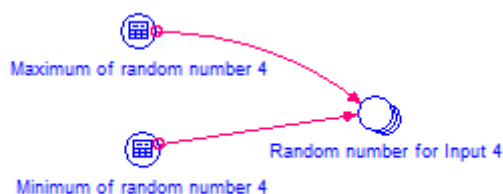
Levá část obrázku 21 zastává funkci přerozdělení dopravních jednotek. S tokem „*Input*“ komunikují pouze konvektory „*Correction_of_Random_number_X*“. Proměnné „*Random_number_X*“ generují náhodná čísla, zatímco konvektory „*Correction_of_Random_number_X*“ zajišťují konzistentní výstupy. Hlavním cílem

korektivních konvektorů je odladit nedostatky z pseudonáhodně generovaných čísel.



Obrázek 22 - Výstup z dopravního uzlu ve tvaru kříže (obrázek autora)

Obrázek 22 značí funkci výstupů. Proměnná „*Output_X*“ soustřeďuje všechny výstupy jedním směrem. Konvektor „*Sum_of_output_X*“ „spojuje“ aktuální dopravní uzel s dalším uzlem.

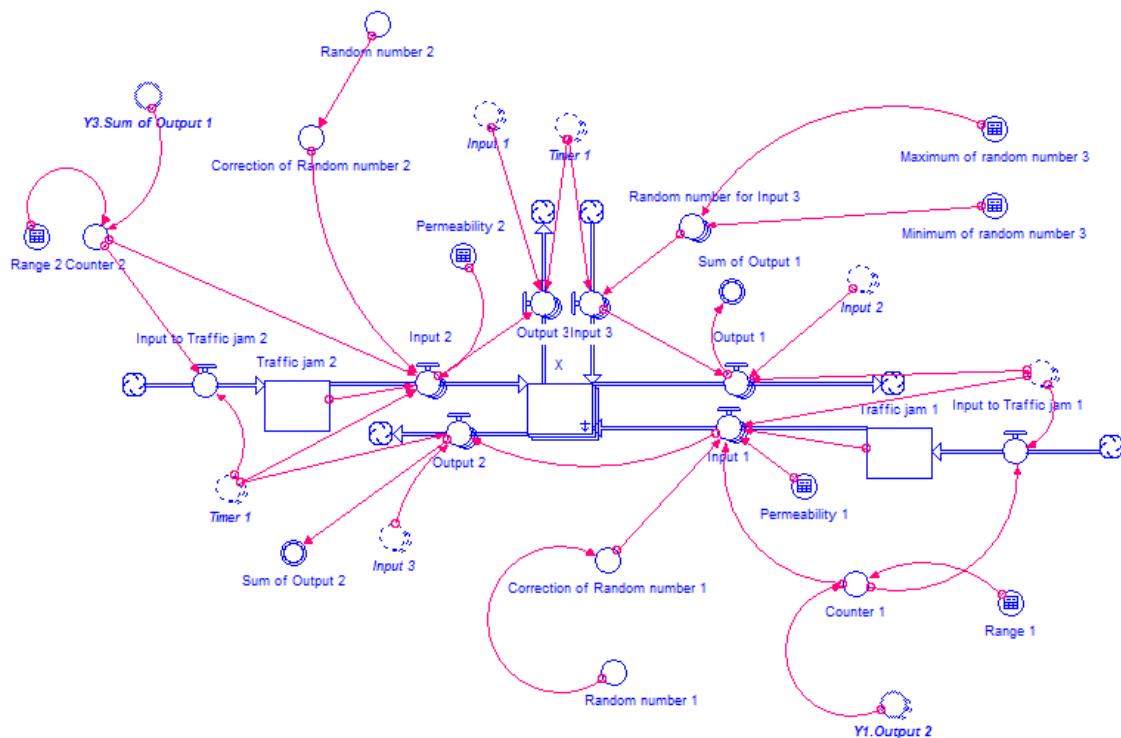


Obrázek 23 - Vstup z externího systému (obrázek autora)

0 vstupech z jiných systémů rozhoduje proměnná „*Random_number_for_Input_X*“, jejíž minimální hranici určuje konvektor „*Minimum_of_random_number_X*“, zatímco maximální hranici určuje „*Maximum_of_random_number_X*“.

4.2.2 Dopravní uzel ve tvaru písmena T

Dopravní uzel do písmene T je zobrazen na obrázku 24. Na obrázku je naznačena zjednodušená verze křížové křižovatky.



Obrázek 24 - Dopravní uzel ve tvaru písmena T (obrázek autora)

Zásadním rozdílem dopravního uzlu ve tvaru písmene T oproti křižovatce křížového tvaru je menší počet přítoků a odtoků. Díky tomuto rozdílu je možné z modelu vyjmout velkou část konvektorů. Velká část proměnných typu korekce pseudonáhodných čísel v tomto modelu postrádá smysl. Rozdělení toku dopravních prostředků řeší pomocí jednoho vygenerovaného pseudonáhodného čísla pro jeden přítok. Poté lze rozdělení dopravních prostředků popsat rovnicí 16 a rovnicí 17.

$$\begin{aligned}
 & \text{ROUND}(\text{TIME MOD } 8 = \text{Timer}_1[3]) \text{ THEN} \\
 & \text{IF } \text{Permeability}_1 > (\text{Traffic_jam}_1 + \text{Counter}_1) \text{ THEN} \\
 & (\text{Traffic_jam}_1 + \text{Counter}_1) * \text{Correction_of_Random_number}_1 \text{ ELSE} \\
 & \text{Permeability}_1 * (\text{Correction_of_Random_number}_1) \text{ ELSE } 0 \tag{16}
 \end{aligned}$$

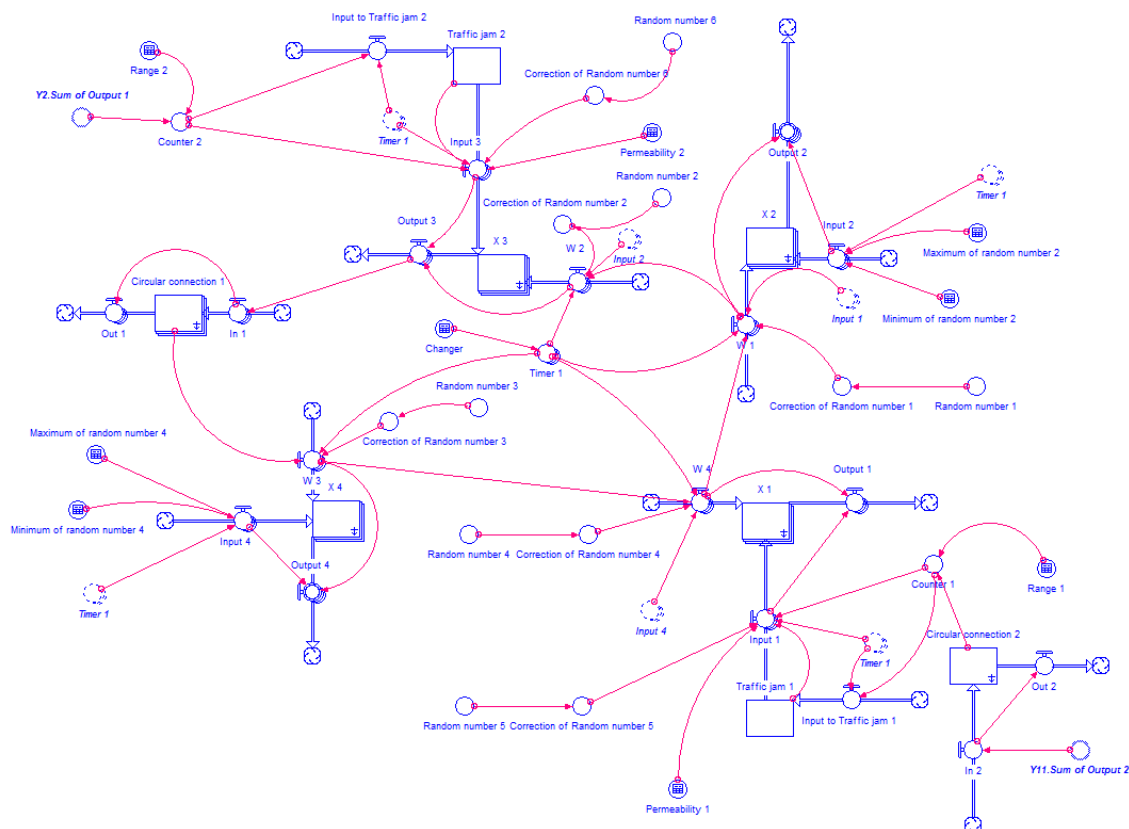
$$\begin{aligned}
 & \text{ROUND}(\text{TIME MOD } 8 = \text{Timer}_1[3]) \text{ THEN} \\
 & \text{IF } \text{Permeability}_1 > (\text{Traffic_jam}_1 + \text{Counter}_1) \text{ THEN} \\
 & (\text{Traffic_jam}_1 + \text{Counter}_1) * (1 - \text{Correction_of_Random_number}_1) \text{ ELSE} \\
 & \text{Permeability}_1 * (1 - \text{Correction_of_Random_number}_1) \text{ ELSE } 0 \tag{17}
 \end{aligned}$$

Jak je vidět z rozdílu rovnice 16 a rovnice 17, první rovnice převezme korekované pseudonáhodné číslo a roznásobí jím hodnotu z hladiny „Traffic_jam_X“, nebo kombinaci konvektoru „Counter_X“ s hladinou „Traffic_jam_X“. Druhá rovnice provede stejnou operaci až na jednu drobnost. Od hodnoty jedna

odečte hodnotu konvektoru „*Correction_of_Random_number_X*“. Tímto krokem se rozloží veškeré možné procházející dopravní prostředky.

4.2.3 Kruhový objezd

Kruhová křižovatka je zachycena na obrázku 25.



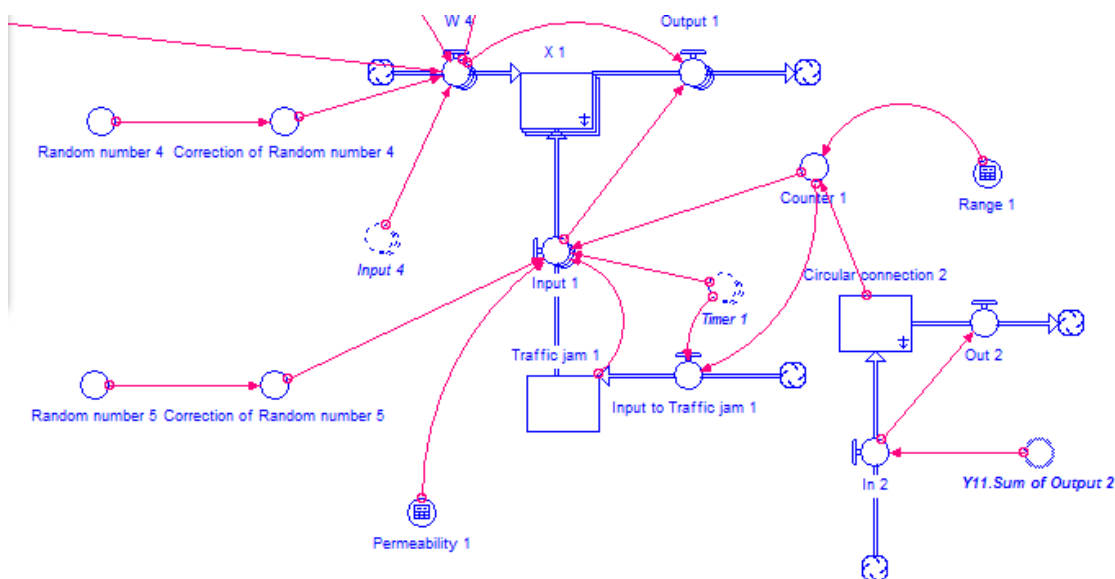
Obrázek 25 - Kruhový dopravní uzel (obrázek autora)

Na obrázku 26 je vidět část světelného kruhového dopravního uzlu. Hladina „*X_1*“ plní funkci vstupu do křižovatky. Dále tato hladina plní kontrolní funkci, kde pokud je model nastaven podle prvotních předpokladů, tak hladina „*X_1*“ nabývá pouze nulové hodnoty.

Odtok „*Output_1*“ slouží jako proměnná, přes kterou odcházejí dopravní prostředky z kruhového objezdu. Další funkcí tohoto toku je odstraňování duplikátních dopravních prostředků z modelu. Duplikace vznikají v zásadě, kvůli postavení modelu kruhové křižovatky. Důvodem je nemožnost přechodu dopravních jednotek mezi dimenzemi. Proto se v jeden čas vytvoří duplikace

dopravních jednotek ve správné dimenzi a původní přebytečné dopravní jednotky odtečou tokem „*Output_X*“.

Za přímý vstup do této části se dají považovat dva přítoky „*W_4*“ a „*Input_1*“. Přes přítok „*Input_1*“ přicházejí dopravní prostředky z provázaného modulu. Zatímco přítokem „*W_4*“ procházejí dopravní prostředky ze zbytku kruhového objezdu.



Obrázek 26 - Vstup do dopravního kruhového uzlu (obrázek autora)

Kruhová křižovatka je na rozdíl od zbylých křižovatek v modelu modelována v pěti dimenzích. První dimenze slouží pro odchod dopravních prostředků z modulu a zároveň vyrovnává duplikaci dopravních prostředků. Zbylé čtyři dimenze slouží pro průchod kruhovým dopravním uzlem. Platí, že pro každý vstup je vymezena jedna dimenze. Rozdělení dimenzí vycházelo z potřeby určit, který dopravní prostředek a kde vjel do křižovatky. Díky tomuto opatření nebyl problém nastavit pravidla pro pohybování se kruhovým objezdem tak, aby odpovídaly realitě. Další nespornou výhodou tohoto rozdělení je přehlednost modelu.

První dimenze toku „*Input_X*“, který se nazývá „*Leave*“ je popsána Rovnicí 18. Předpokládá se, že určitý podíl dopravních prostředků opustí kruhovou křižovatku prvním možným výjezdem. Tento podíl je určen konvektorem „*Correction_of_Random_number_5*“.

Zbylé dimenze („*Z1*“, „*Z2*“, „*Z3*“, „*Z4*“) plní funkci vstupu do kruhové křižovatky (pro proměnnou „*Input_X*“). Pro každou proměnnou „*Input_X*“ je

vyhrazena jedna z dimenzí „ZX“. Dále platí, že každý číselný znak dimenze „ZX“ je shodný pro každou proměnnou „Input_X“, neboli každá proměnná „Input_X“ má vyhrazenou dimenzi „ZX“ a číselný znak místo „X“ je shodný.

```
ROUND(
IF ( TIME MOD 8 = Timer_1[2]) THEN
IF Permeability_1 > (Traffic_jam_1+Counter_1) THEN
(Traffic_jam_1+Counter_1) * Correction_of_Random_number_5 ELSE
Permeability_1*(Correction_of_Random_number_5) ELSE 0 )
```

 (18)

```
IF ( TIME MOD 8 = Timer_1[2]) THEN
IF Permeability_1 > (Traffic_jam_1+Counter_1) THEN
(Traffic_jam_1+Counter_1) *(1- Correction_of_Random_number_5) ELSE
Permeability_1*(1- Correction_of_Random_number_5) ELSE 0 )
```

 (19)

Odtok „Output_X“ má v první dimenzi „Leave“ rovnici 20. Z této rovnice a z obrázku 26 je patrné, že se jedná o součet dopravních prostředků, které opustí kruhový objezd první výjezdem a dopravní prostředky, které opustí kruhový objezd druhým a vyšším výjezdem. Dimenze „ZX“ slouží pro odstraňování duplicit v modelu.

$$\text{Input}_1[\text{Leave}] + W_4[\text{Leave}] \quad (20)$$

$$\text{Input}_1[\text{Z1}] \quad (21)$$

$$W_4[\text{Z2}] \quad (22)$$

$$W_4[\text{Z3}] \quad (23)$$

$$W_4[\text{Z4}] \quad (24)$$

Přítoky „W1“, „W2“, „W3“, „W4“ (dále už jen „WX“) mají vždy dvě z pěti dimenzí nulové. Nulové hodnoty zabraňují cyklení dopravních prostředků v kruhovém dopravním uzlu. Cyklení je zabráněno kvůli popisu reality.

První dimenze „Leave“ přítoku „W_1“ se dá rozdělit na několik částí. Každá část plní určitou funkci. První část určuje, kolik dopravních prostředků vstoupivších proměnnou „Input_1“ opustí kruhovou křižovatkou výstupem „Output_2“. Druhá část propustí zbylé dopravní prostředky v kruhové křižovatce, které vstoupily proměnnou „Input_3“. Poslední část zobrazuje, kolik dopravních prostředků vstoupivších přítokem „Input_4“ opustí kruhový dopravní uzel výstup „Output_2“.

```
ROUND(
IF TIME MOD 8 = Timer_1[2] THEN Input_1[Z1]*Correction_of_Random_number_1
ELSE
```

IF TIME MOD 8 = Timer_1[3] THEN W_4[Z3] ELSE
 IF TIME MOD 8 = Timer_1[5] THEN W_4[Z4]*(Correction_of_Random_number_1)
 ELSE 0) (25a)

0 (25b)

0 (25c)

ROUND(IF TIME MOD 8 = Timer_1[2] THEN Input_1[Z1]*(1 -
 Correction_of_Random_number_1) ELSE 0) (25d)

ROUND(IF TIME MOD 8 = Timer_1[5] THEN W_4[Z4]*(1-
 Correction_of_Random_number_1) ELSE 0) (25e)

V zásadě lze říci, že každá proměnná „W_X“ funguje na stejném principu. Zásadní rozdíl, je možné nalézt v posunu dimenzí s nulovými hodnotami. Tento rozdíl je vidět při porovnání soustavy rovnic 25 se soustavou rovnic 26.

ROUND(IF TIME MOD 8 = Timer_1[5] THEN
 Input_2[Z2]*Correction_of_Random_number_2 ELSE
 IF TIME MOD 8 = Timer_1[3] THEN HISTORY(W_1[Z1], TIME - 1)
 *(Correction_of_Random_number_2) ELSE
 IF TIME MOD 8 = Timer_1[7] THEN HISTORY(W_1[Z4], TIME - 2) ELSE 0) (26a)

ROUND(IF TIME MOD 8 = Timer_1[3] THEN HISTORY(W_1[Z1], TIME - 1) *(1-
 Correction_of_Random_number_2) ELSE 0) (26b)

ROUND(IF TIME MOD 8 = Timer_1[5] THEN Input_2[Z2]*(1-
 Correction_of_Random_number_2) ELSE 0) (26c)

0 (26d)

0 (26e)

Problém kruhové vazby je v tomto modulu řešen rozetnutím kruhu mezi odtokem „Output_3“ a přítokem „W_3“. Toto přetnutí má za následek, že dopravní prostředky procházejících přes proměnnou „Output_3“ se zdrží o jednu časovou jednotku. Po tomto zásahu se musí odkazovat na jejich hodnoty s posunutím času. V modelu to má za následek, že se proměnné odkazují na liché i sudé dimenze konvektoru „Timer_1“. Pokud by se zde nemusela přetínat kruhová vazba a zároveň by musela být přetnuta jinde v modelu, tak by proměnné odkazovaly pouze na sudé dimenze proměnné „Timer_1“.

Rovnice 27 značí vstup do hladiny „Circular_connection_1“. Tato hladina přebírá veškeré hodnoty z proměnné „Output_3“. V hladině „Circular_connection_1“ se ukládají hodnoty z předchozího kroku proměnné „Output_3“. Tato vlastnost je zaručena výstupem „Out_1“ s rovnicí 28. V rovnici 28 model vyhodnotí aktuální čas.

Pokud je čas simulace roven jedné, z hladiny „Circular_connection_1“ nic neodteče. Pokud je čas simulace jiný než jedna, tak z hladiny „Circular_connection_1“ odteče hodnota rovna přítoku „In_1“ z minulé časové části. Hladina „Circular_connection_1“ je pětidimenzionální hladinou z důvodu zachování logiky celého modulu.

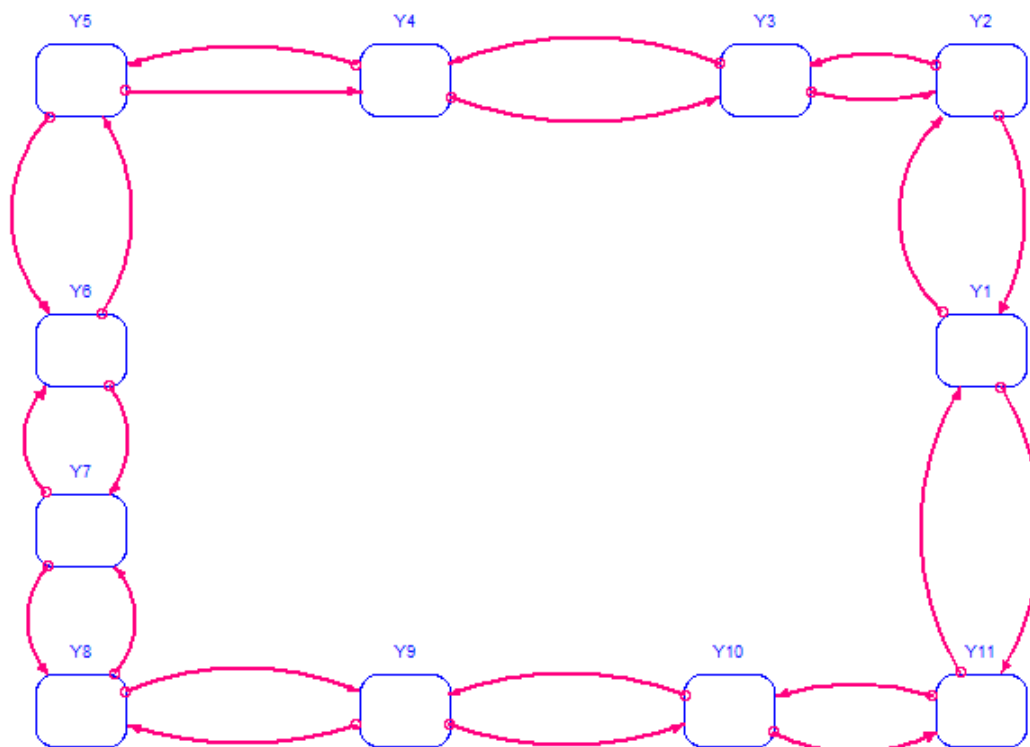
Output_3 (27)

IF (TIME = 1)THEN 0 ELSE HISTORY(In_1, TIME -1) (28)

4.3 Propojení městského objezdu

Model simuluje vnější dopravní okruh v Hradci Králové. Převod dopravního okruhu na model je vyobrazen na obrázku 27. Tento převod se skládá z jedenácti modulů. V každém modulu je namodelována jedna křižovatka.

Modul „Y_1“ simuluje kruhový objezd se světelným značením (tento dopravní uzel byl vybrán jako první z důvodu jeho unikátnosti v dopravním okruhu města Hradec Králové). Následující dopravní uzel „Y_2“ leží severozápadním směrem od dopravního uzle „Y_1“. Následující dopravní uzly jsou logicky pospojovány dle reálného rozložení.



Obrázek 27 - Vnější okruh (obrázek autora)

Moduly jsou propojeny za pomoci konvektorů „*Sum_of_output*“. Tento konvektor „přemostňuje“ informace mezi moduly. Následně se s těmito informacemi pracuje jako s hodnotami dopravních jednotek vygenerovaných za pomoci generátoru pseudonáhodných čísel.

4.4 Simulace

V kapitole Simulace se bude nahlížet na parametry simulace a následně na výsledky simulace.

4.4.1 Parametry simulace

Časový start simulace byl nastaven na hodnotu jedna. Konečný stav simulace byl nastaven na hodnotu pět set. Toto časové rozpětí se ukázalo jako dostatečně dlouhé, aby vypovídalo o stavech v modelu během simulace. Kratší časové rozpětí by nemělo považovanou vypovídající hodnotu. Zatímco delší časové rozpětí by mělo za následek složitější pochopení výsledků simulace.

Krokování simulace bylo nastaveno na hodnotu jedna. Toto nastavení se ukázalo jako optimální. Jak kratší, tak delší krokování by nemělo smysl.

Jako integrační metoda byla ponechána „Euler’s method“. Změna na metodu „Runge-Kutta 2“, nebo „Runge-Kutta 4“ by neměla na výsledky simulace žádný dopad. Z tohoto důvodu byla ponechána defaultní metoda.

Časovou jednotku bylo nutné pro simulaci vytvořit z důvodu její absence v základních možnostech software STELLA. Vytvořená časová jednotka se jmenuje „*Minutes/2*“. Tento název vychází z nutnosti rozdělení času na poloviny a z důvodu kruhových vazeb.

4.4.2 Výsledky simulace

Výstupem ze simulace jsou grafy popisující její průběh. V grafech se objevují důležité informace o provozu přes základní ukazatele a průřez problematiky dopravní zácpy až po podrobnější informace o dopravní zácpě.

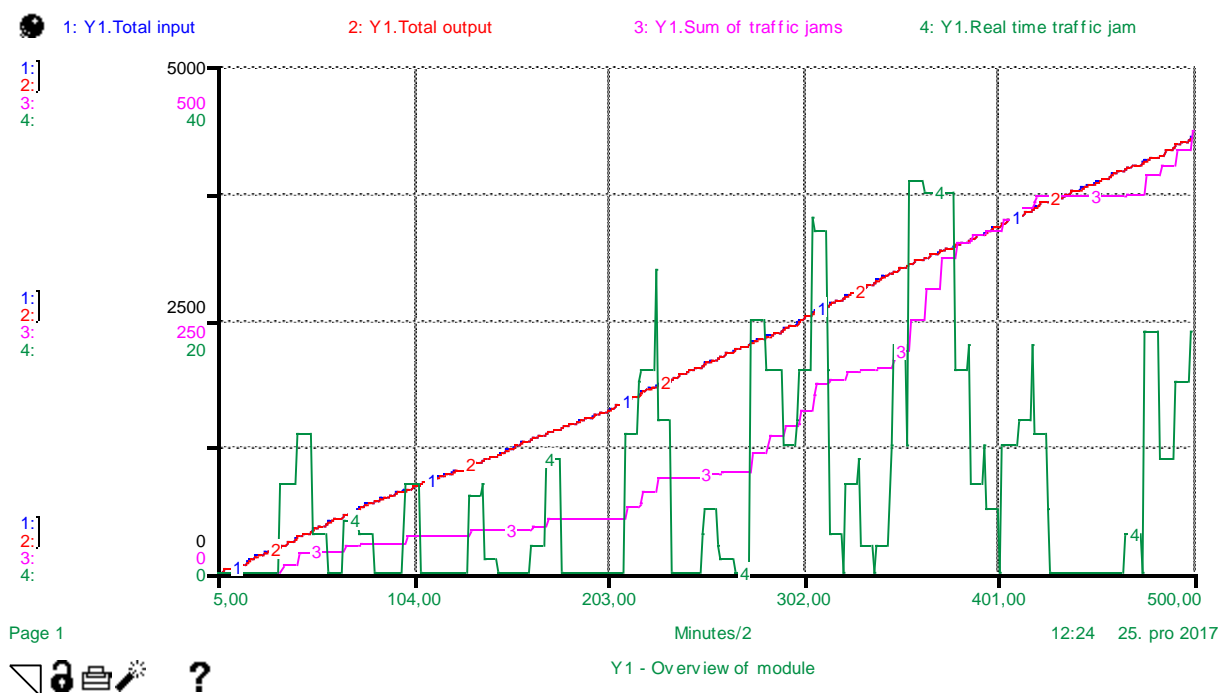
Vybrané grafy značí základní informace o stavech proměnných v modelu. Informace zachycené v grafech mají silnou vypovídající hodnotu o stavech v celém modelu. Pro každé zkoumané odvětví je vytvořeno jedenáct grafů. Tedy pro každý

modul je v jednom odvětví vytvořen jeden graf. Tento postup vychází ze spojitosti mezi jednotlivými moduly.

Přehled modulů

Interface je tvořen několika grafy a listy s proměnnými.

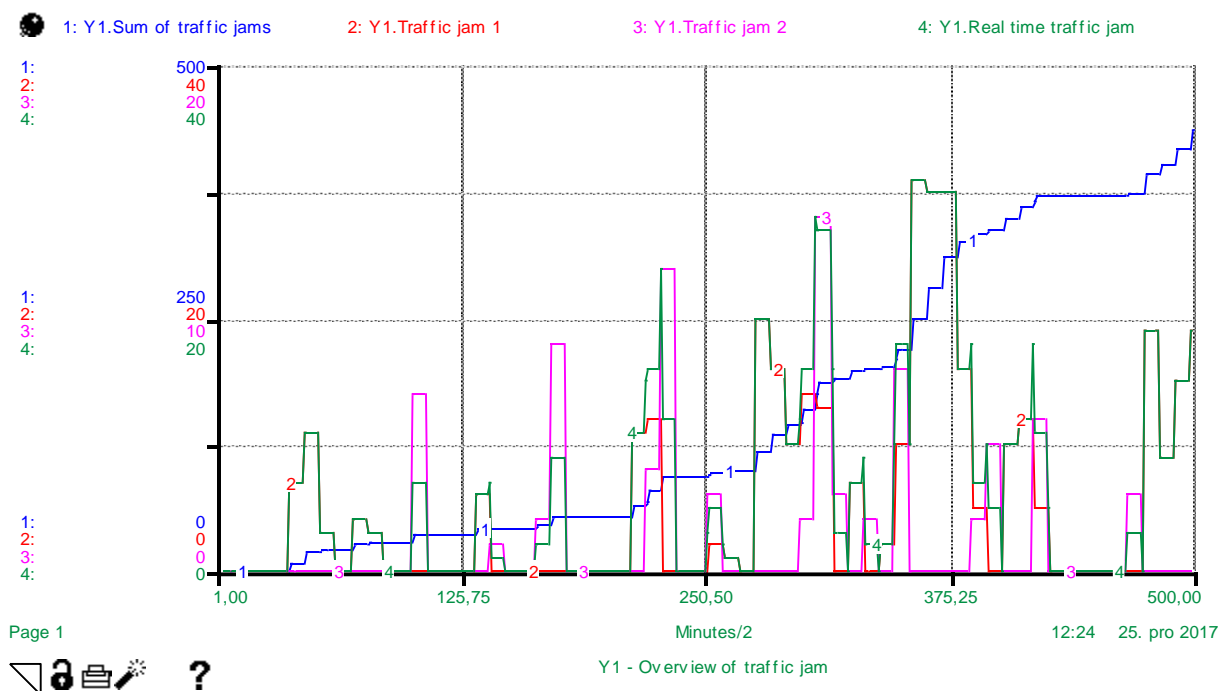
První graf ukazuje základní charakteristiky křižovatek (pro každou křižovátku existuje vlastní graf a to platí pro všechny grafy). Zkoumají se zde vstupy do křižovátky, výstupy z křižovátky, celkový počet aut v dopravní zácpě a aktuální hodnota dopravní zácpy. Zásadní rozdíl nastává mezi prvním grafem (z kruhového objezdu) a zbylými grafy. Na prvním grafu je znázorněna situace, kde některá auta se zdrží v křižovatce (odpovídá realitě. V dalších grafech tato situace nenastává (také odpovídá realitě).



Obrázek 28 Graf - Celkový přehled modulu (obrázek autora)

Přehled dopravních kolapsů

Grafy v této části popisují stav dopravní zácpy. Na obrázku 29 je zobrazen průběh stavů dopravních komplikací během simulace. Na ose „Y1.Sum_of_traddic_jams“ je znázorněna stoupající tendence této proměnné. Další vlastnost této osy je reakce na změny proměnných „Traffic_jam_1“ a „Traffic_jam_2“

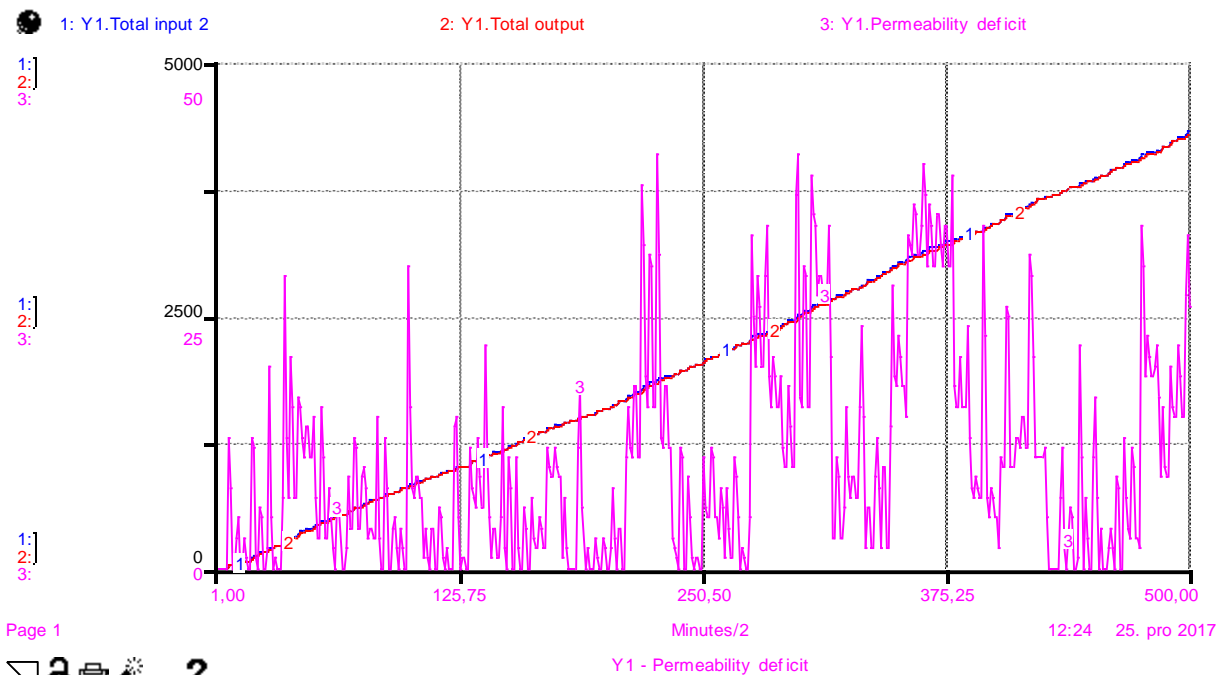


Obrázek 29 Graf - přehled dopravních zácp (obrázek autora)
Nedostatek propustnosti

Tento jev popisuje stav propustnosti křižovatky. V grafech jsou zobrazeny vstupy a výstupy do modulů. Dále je zde vidět nedostatek v propustnosti.

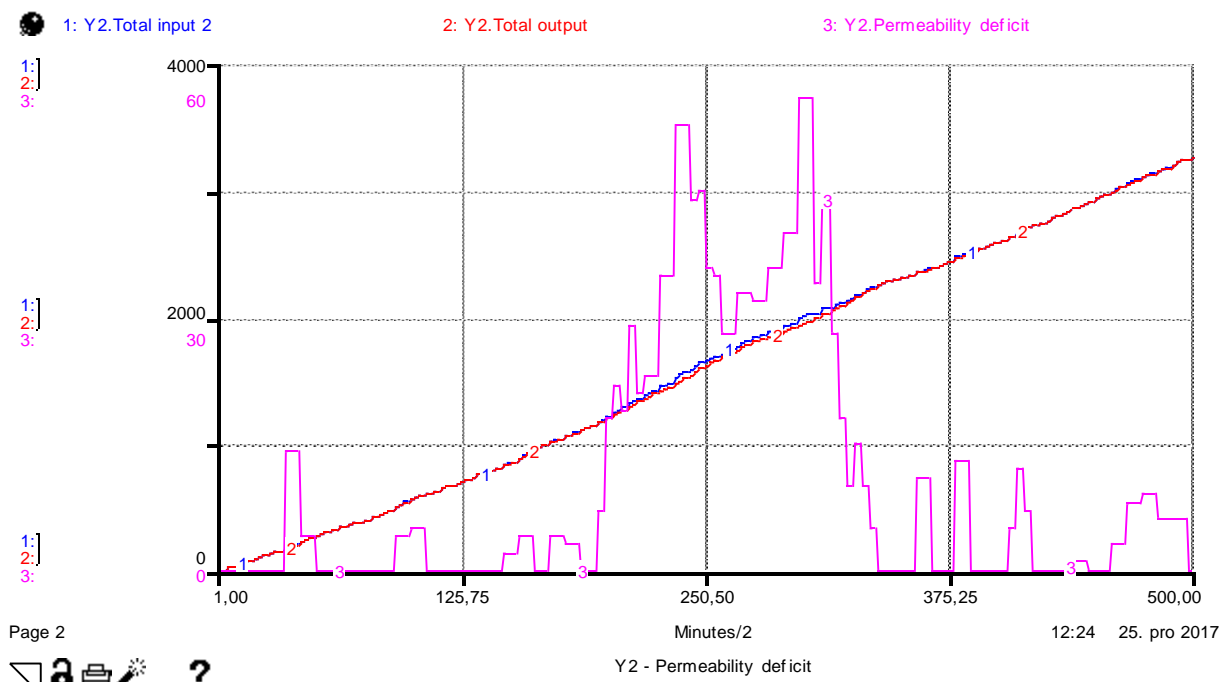
V těchto grafech je ukázán problém s kruhovou vazbou. V prvním grafu jsou hodnoty pro proměnnou „Permeability_deficit“ ostré. V ostatních grafech mají jiný tvar. Tento rozdíl je způsoben z důvodu dělení času.

Na obrázku 30 lze sledovat provázanost proměnných. V případě, že hodnoty proměnné „Permeability_deficit“ jsou výrazně vysoké, nastává převis hodnot proměnné „Total_input_2“ nad proměnnou „Total_output“. Tento převis je možné pozorovat ve třetím kvartále.



Obrázek 30 Graf - Nedostatek propustnosti 1 (obrázek autora)

Více viditelný převis lze pozorovat na obrázku 31. Převis je pozorovatelný na pomezí druhého a třetího kvartálu.



Obrázek 31 Graf - Nedostatek propustnosti 2 (obrázek autora)

4.5 Zelená vlna

Zelená vlna se dá v modelu simulovat synchronizací křižovatek za předpokladu, že mezi křižovatkami není žádná vzdálenost - tedy přesun z jedné křižovatky do druhé nezabere žádnou časovou jednotku. Hodnoty proměnných „*Timer*“ se nastaví na hodnoty odpovídající hladkému průběhu.

Řekněme, že prvotní „*Timer*“ se nachází v modulu „*Y1*“. „*Y1.Timer_1*“ nabývá základní hodnoty (1,2,3,4,5,6,7,0), respektive „*Y1.Changer*“ má hodnotu nula. Dále předpokládáme, že dopravní prostředky vjedou v časové hodnotě rovné druhé pozici z „*Y1.Timer_1*“ tedy v čase 2. Dále je nutné vědět, že do modulu „*Y2*“ přijíždí dopravní prostředky v čase, který je uložen v „*Y2.Timer_1*“ na třetí pozici. Tím pádem na třetí pozici v proměnné „*Y2.Timer_1*“ musí být číslo 2, tedy „*Y2.Changer*“ musí být nastaven na hodnotu 7 atd. Křižovatka do tvaru T přebírá dopravní prostředky vždy v čase uloženém na třetí pozici, křižovatka do kříže v čase uloženém na páté pozici.

Díky těmto předpokladům se zelená vlna dá vyjádřit rovnicemi:

$$Y2.Timer_1[3] = Y1.Timer_1[2] \quad (29a)$$

$$Y3.Timer_1[5] = Y2.Timer_1[3] \quad (29b)$$

$$Y4.Timer_1[5] = Y3.Timer_1[5] \quad (29c)$$

$$Y5.Timer_1[3] = Y4.Timer_1[5] \quad (29d)$$

$$Y6.Timer_1[3] = Y5.Timer_1[3] \quad (29e)$$

$$Y7.Timer_1[5] = Y6.Timer_1[3] \quad (29f)$$

$$Y8.Timer_1[5] = Y7.Timer_1[5] \quad (29g)$$

$$Y9.Timer_1[5] = Y8.Timer_1[5] \quad (29h)$$

$$Y10.Timer_1[3] = Y9.Timer_1[5] \quad (29ch)$$

$$Y11.Timer_1[5] = Y10.Timer_1[3] \quad (29i)$$

Pokud předešlé rovnice platí, tak nastává zelená vlna za předpokladu, že mezi křižovatkami je nulová časová vzdálenost.

Rozšíření rovnic o časovou vzdálenost mezi křižovatkami:

$$Y2.Timer_1[3] = (Y1.Timer_1[2] + Y2.Range_1) \text{ mod } 8 \quad (30a)$$

$$Y3.Timer_1[5] = (Y2.Timer_1[3] + Y3.Range_1) \text{ mod } 8 \quad (30b)$$

$$Y4.Timer_1[5] = (Y3.Timer_1[5] + Y4.Range_1) \text{ mod } 8 \quad (30c)$$

$$Y5.Timer_1[3] = (Y4.Timer_1[5] + Y5.Range_1) \bmod 8 \quad (30d)$$

$$Y6.Timer_1[3] = (Y5.Timer_1[3] + Y6.Range_1) \bmod 8 \quad (30e)$$

$$Y7.Timer_1[5] = (Y6.Timer_1[3] + Y7.Range_1) \bmod 8 \quad (30f)$$

$$Y8.Timer_1[5] = (Y7.Timer_1[5] + Y8.Range_1) \bmod 8 \quad (30g)$$

$$Y9.Timer_1[5] = (Y8.Timer_1[5] + Y9.Range_1) \bmod 8 \quad (30h)$$

$$Y10.Timer_1[3] = (Y9.Timer_1[5] + Y10.Range_1) \bmod 8 \quad (30ch)$$

$$Y11.Timer_1[5] = (Y10.Timer_1[3] + Y11.Range_1) \bmod 8 \quad (30i)$$

Dopočítání proměnné „Changer“

$$Y2.Changer = (((Y1.Timer_1[2] + Y2.Range_1) \bmod 8) - Y2.Timer_1[3]) \bmod 8 \quad (31a)$$

$$Y3.Changer = (((Y2.Timer_1[2] + Y3.Range_1) \bmod 8) - Y3.Timer_1[3]) \bmod 8 \quad (31b)$$

$$Y4.Changer = (((Y3.Timer_1[2] + Y4.Range_1) \bmod 8) - Y4.Timer_1[3]) \bmod 8 \quad (31c)$$

$$Y5.Changer = (((Y4.Timer_1[2] + Y5.Range_1) \bmod 8) - Y5.Timer_1[3]) \bmod 8 \quad (31d)$$

$$Y6.Changer = (((Y5.Timer_1[2] + Y6.Range_1) \bmod 8) - Y6.Timer_1[3]) \bmod 8 \quad (31e)$$

$$Y7.Changer = (((Y6.Timer_1[2] + Y7.Range_1) \bmod 8) - Y7.Timer_1[3]) \bmod 8 \quad (31f)$$

$$Y8.Changer = (((Y7.Timer_1[2] + Y8.Range_1) \bmod 8) - Y8.Timer_1[3]) \bmod 8 \quad (31g)$$

$$Y9.Changer = (((Y8.Timer_1[2] + Y9.Range_1) \bmod 8) - Y9.Timer_1[3]) \bmod 8 \quad (31h)$$

$$Y10.Changer = (((Y9.Timer_1[2] + Y10.Range_1) \bmod 8) - Y10.Timer_1[3]) \bmod 8 \quad (31ch)$$

$$Y11.Changer = (((Y10.Timer_1[2] + Y11.Range_1) \bmod 8) - Y11.Timer_1[3]) \bmod 8 \quad (31i)$$

Výpočet nastavení zelené vlny lze vyjádřit pomocí rovnice 32. Předpokladem pro tuto rovnici je situace, ve které se proměnné v některém modulu považují za neměnné. Tedy jeden modul se vybere jako modul se základním nastavením a od něj se dopočítá zbylé nastavení ostatních modulů. Aby rovnice 32 byla platná, je nutné vybrat jako základní modul „Y1“. Alternativou je přeznačit moduly. Ve své podstatě je vhodné začít od modulu s nejnižším číslem.

$$Y_a.Changer = \left(((Y_{a-1}.Timer[c] + Y_a.Range_1) \bmod 8) - Y_a.Timer_1[b] \right) \bmod 8 \quad (32)$$

Proměnná „a“ značí číslo aktuálního modulu, proměnná „b“ je dimenze aktuálního konvektoru „Timer“, proměnná „c“ je dimenze konvektoru „Timer“ používaného v předcházejícím modulu.

5 Shrnutí výsledků

Model jako takový je funkční, odráží-li realitu na předpokládané úrovni. Není kopií reality, ale zároveň vytváří dostatek relevantních informací. Při modelování byla sestavena nová koncepce systému systémové dynamiky v dopravě. Přerozdělení dopravních prostředků za využití polí se dá považovat za unikátní přístup. Nevýhodou tohoto konceptu je složitost odkazování se na pole. Na druhou stranu se tomuto problému nelze vyhnout za předpokladu využití polí.

Rozdělení času pomocí modula je poměrně zajímavé řešení. Celý model se musí řídit modulem osmi. Pokud by software STELLA umožňoval kruhové vazby, stačilo by modulo čtyř. Tento problém zanesl do modelu významný prvek obcházení kruhových vazeb.

Rozložení modelu do modulů, kde každý modul obsahuje jednu křížovátku, je poměrně zajímavý postup. Díky tomuto rozdělení je zajištěna znouvopoužitelnost a zároveň je podpořeno pochopení modelu jako takového.

Pomocí modelu lze zkoumat jeden z nejzávažnějších problémů dnešní dopravy. Dopravní kolapsy jsou k vidění téměř v každém větším městě několikrát denně. V modelu lze tento městský neduh zkoumat a zároveň lze upravovat různá nastavení tak, aby se dopravní kolony povedlo eliminovat, nebo alespoň zmírnit.

Měněním vstupů do modelu lze simulovat neobvyklé situace, kdy přes dopravní síť prochází více dopravních prostředků, než je obvyklé. Případně lze upravovat rozhodování řidičů, kdy je preferována jedna trasa více než ostatní.

Model nabízí různé informace o dění průběhu simulace. Poměrně velká část modelu je věnována právě ukazatelům zkoumajícím průběh simulace. Tyto ukazatele jsou převedeny do celků v podobě grafů. Informativní část modelu lze rozšířit či upravit podle požadavků zkoumání. V této práci se zkoumá například aktuální počet dopravních prostředků v koloně, či celkový počet, anebo deficit propustnosti dopravních uzlů.

6 Problémy

Během tvorby modelu nastalo několik problémů. Veškeré zásadní problémy spočívaly v limitacích software STELLA. Za nejzásadnější problém se dá považovat nemožnost vytvoření přímé kruhové vazby. Méně limitujícím problémem je nemožnost nastavení hodnot pro simulace v neexistujícím čase. Za nejméně limitující, ale stále poměrně zásadní problém lze považovat nemožnost úpravy definičního oboru proměnné.

6.1 Nemožnost kruhové vazby

Kruhová vazba se v modelu nachází dvakrát. Obě dvě kruhové vazby popisují stav reality. Nejedná se tedy o nezamýšlené kruhové vazby.

První kruhová vazba vznikla v modulu „Y1“. Kruhový objezd bez kruhové vazby by namodelovat nešel. Dalo by se říci, že se zde o kruhovou vazbu nejedná, protože žádná z dimenzí není propojena úplně dokola.

Problém byl vyřešen pomocí hladiny „Circular_connection_1“.

Druhá kruhová vazba vzniká při uzavření okruhu propojením jednotlivých modulů. Zde se problém řeší pomocí hladiny „Circular_connection_2“. Tato hladina byla implementována do modulu „Y1“ a mohla by být implementována do kteréhokoliv modulu. Na chování modelu by toto nemělo žádný dopad.

Tento problém na rozdíl od prvního problému zanechává na modelu určité nedostatky. Projevem této problematiky je nutnost upravit čas. Model byl původně navržen tak, aby v každé časové jednotce probíhaly operace. V důsledku kruhových vazeb se musel čas rozdělit na dvě části. Nyní platí, že každá časová jednotka má dvě části. Například čas 1 a čas 2 se považují za stejný časový úsek. Výsledkem je, že model potřebuje na simulaci určité časové relace dvojnásobný počet dt , než je simulovaný čas. Dalším projevem je situace v grafu popsáném v části Permeability deficit.

Určitě by bylo pozitivní změnou, kdyby software STELLA umožňoval propojení pomocí kruhové vazby například v rozšířeném nastavení programu, či zeptáním se uživatele v průběhu modelování.

6.2 Vracení nenulových čísel z funkce „HISTORY“

Pokud se pomocí této funkce model odvolává na historii proměnných v čase, ve kterém neexistují, tak vrátí nenulové číslo. Tato situace vedla k nesprávnému fungování modelu na začátku simulace.

Příkladem proměnné s tímto problémem může být uveden konvektor „Counter“. Původní rovnice 33 se ukázala jako chybná. V prvních několika krocích modelu rovnice vracela špatné hodnoty (v tomto případě se jednalo o nenulové hodnoty). Pro prvotní sestavení rovnic se vycházelo z premisy, že před spuštěním modelu jsou veškeré hodnoty nulové. Tento předpoklad vznikl z domněnky, že v neexistujícím čase neexistují proměnné a tím pádem odvoláním se na neexistující proměnnou se vrátí nulová hodnota. Simulace ukázala, že původní premisa byla chybná. Software STELLA je nastaven tak, že v neexistujícím čase existují proměnné a mají hodnoty přiřazené. Neboli pokud se proměnná odvolá na hodnoty z neexistujícího času, software je do ní generuje. Software tak vytvoří z neexistujícího času existující. Tato funkce má jistě své opodstatnění pro tvorbu modelů systémové dynamiky, nicméně pro tento případ byla nevhodná a musela se upravit rovnice proměnné „Counter“. Negativem STELLY 10.0 je nemožnost nastavit hodnoty pro proměnné před startem simulace.

```
HISTORY(Circular_connection_2, TIME -1 -Range_1) +  
HISTORY(Circular_connection_2, TIME -2 -Range_1) +  
HISTORY(Circular_connection_2, TIME -3 -Range_1) +  
HISTORY(Circular_connection_2, TIME -4 -Range_1) +  
HISTORY(Circular_connection_2, TIME -5 -Range_1) +  
HISTORY(Circular_connection_2, TIME -6 -Range_1) +  
HISTORY(Circular_connection_2, TIME -7 -Range_1) +  
HISTORY(Circular_connection_2, TIME -8 -Range_1) (33)
```

Situace měla různé možnosti řešení. Jednou z možností bylo změnit časové nastavení modelu. Další možnou alternativou bylo manuálně upravit kód v tocích nebo také upravit kód v konvektoru „Counter“.

Změna času

Premisa změnu času spočívala ve změně neexistujícího času na čas existující. Časové rozpětí modelu se změnilo z rozpětí nula až X, na mínus osm až X. Následně byly upraveny podmínky pro vstup do modelu. Platilo tedy, že několik prvních

problematických časových úseků plnilo funkci generátoru nul pro problematické proměnné. Tento způsob byl sice plně funkční, ale metoda, kterou model plnil funkci, nebyla velkým pozitivem.

Úprava kódu v tocích

Úprava kódu v tocích zůstala jen v teoretické rovině. Tento stav zapříčinila složitost aplikace tohoto řešení. Bylo by potřeba stanovit stejné podmínky jako u posledního způsobu a vytvořit nové vazby mezi proměnnými. Další nevýhodou byla stavba modelu, ve kterém je několikanásobně více toků než konvektorů „Counter“.

Změna konvektoru „Counter“

Změna konvektoru „Counter“ se nakonec ukázala jako nejlepší alternativa. „Counter“ se musel upravit pomocí osmi podmínek, kde se pro prvních sedm časových jednotek musely nastavit speciální podmínky. Viz proměnná „Counter“.

6.3 Zaokrouhlovací mechanismus

Posledním zjištěným problémem je nemožnost změnit zaokrouhlovací mechanismus v software STELLA. Problémem jsou nekonzistentní hodnoty po zaokrouhlení. Tento jev nastává při roznásobení lichého čísla s hodnotou 0,5 a následným zaokrouhlením.

Řešení tohoto problému během modelování bylo jednoduché. Jediným negativním dopadem tohoto řešení byl vysoký nárůst konvektorů.

Pro řešení tohoto problému a jemu podobných by bylo vhodné rozšířit software STELLA o možnost stanovování globálních pravidel. Podle názoru autora by bylo vhodné tuto utilitu implementovat v novějších softwarech od společnosti isee systems.

Přínosem by byla možnost v globálních pravidlech omezovat proměnné anebo skupiny proměnných za pomoci podmínek - případně možnost odstranění některých hodnot z oboru hodnot, anebo z definičního oboru hodnot.

V tomto modelu by tato možnost měla jen estetický dopad. Využitelnost v modelování systémové dynamiky by ale globální podmínky měly dost výraznou.

7 Závěry a doporučení

Model plní předpokládanou funkci. Největším problémem při tvoření celého modelu byla nutnost obejít kruhové vazby. Díky tomuto zásahu má model dvojnásobnou výpočetní náročnost, než bylo předpokládáno. Druhý závažný problém nastal při práci s nereálným časem. Naštěstí se tuto nedokonalost povedlo opravit a na simulaci tento problém neměl žádný vliv.

Modelem práce je možné reálně simulovat dopravu. Byly vytvořeny tři druhy dopravních uzlů. V práci se podařilo poukázat na možnosti modelů systémové dynamiky. Model je unikátní, jak svojí stavbou, tak hloubkou či znovupoužitelností.

V minulosti také nebyl publikován podobný model systémové dynamiky s dopravní tematikou. Tato struktura modelu by mohla mít dopad na využití modelů systémové dynamiky v budoucnu.

Pomocí vytvoření modelu dopravního okruhu města Hradec Králové se poukazuje na jednoduchost znovupoužití modelu na jiné dopravní problémy. Přidávání či ubírání dopravních uzlů nepožaduje složitou remodelizaci, ale pouze mírné zásahy.

Při postupu modelování by zásadně pomohla úprava modelovacího nástroje od společnosti isee systems. Pokud by totiž v modelovacím nástroji bylo možno vytvářet záměrné kruhové vazby, model by snížil svojí výpočetní náročnost o polovinu. Prezentace výsledků simulace by pak byla přehlednější.

Posledním doporučením je možnost nastavení výchozích hodnot pro nereálný čas simulace. Tato možnost by určitě pomohla i jiným modelům. Důsledkem by byla možnost ovlivňovat předcházející hodnoty simulace před jejím spuštěním.

8 Seznam použité literatury

BARLAS, Yaman. System dynamics: systemic feedback modeling for policy analysis. *System*, 2007, 1: 59. Dostupné na: http://gu.friends-partners.org/Global_University/Global%20University%20System/List%20Distributions/2012/MTI2350_20121123/SYSTEM%20DYNAMICS_%20SYSTEMIC%20FEEDBACK%20MODELING%20FOR%20POLICY%20ANALYSIS%20copy.pdf

BUREŠ, Vladimír a Pavel ČECH, 2008. Systémové vědy a teorie. Hradec Králové: Gaudeamus. ISBN 978-80-7041-155-1.

BUREŠ, Vladimír, 2006. Systémové myšlení a teorie systémů. Hradec Králové: Gaudeamus. ISBN 80-704-1344-1.

BUREŠ, Vladimír, 2009. Systems thinking and systems sciences. Hradec Králové: Gaudeamus. ISBN 978-80-7041-467-5.

BUREŠ, Vladimír, 2011. Systémové myšlení pro manažery. Praha: Professional Publishing. ISBN 978-80-7431-037-9.

CASTELLACCI, Fulvio. Co-evolutionary growth: A system dynamics model. *Economic Modelling*, 2017. Dostupné na: <https://www.sciencedirect.com/science/article/pii/S0264999317304133>

COYLE, R. G. System dynamics modelling: a practical approach. Boca Raton, Fla.: Chapman & Hall/CRC, 2001. ISBN 04-126-1710-2.

DOLEŽAL, Jan, Pavel MÁCHAL a Branislav LACKO. Projektový management podle IPMA. Praha: Grada, 2009. Expert (Grada). ISBN 978-80-247-2848-3.

DRMOLA, Jakub; HUBIK, Tomas. Kessler syndrome: System dynamics model. *Space Policy*, 2018. Dostupné na: <https://www.sciencedirect.com/science/article/pii/S0265964617300966>

DUGGAN, James. A Comparison of Petri Net and System Dynamics Approaches for Modelling Dynamic Feedback Systems. In: 24th International Conference of the Systems Dynamics Society. 2006. Dostupné na: <https://aran.library.nuigalway.ie/handle/10379/4040>

FAN, Chihhao, et al. Modeling computer recycling in Taiwan using system dynamics. *Resources, Conservation and Recycling*, 2018, 128: 167-175. Dostupné na: <https://www.sciencedirect.com/science/article/pii/S0921344916302397>

FONTES, Cristiano Hora de O., et al. Sustainable and renewable energy supply chain: A system dynamics overview. *Renewable and Sustainable Energy Reviews*, 2018, 82: 247-259. Dostupné na: <http://www.sciencedirect.com/science/article/pii/S1364032117312844>

FORD, Frederick Andrew. Modeling the environment: an introduction to system dynamics models of environmental systems. Island Press, 1999. ISBN 1-55963-601-7.

FORRESTER, Jay W. System dynamics, systems thinking, and soft OR. System Dynamics Review, 1994, 10.2-3: 245-256. [cit. 3. dubna 2018]. Dostupné na: <http://web.a.ebscohost.com.ezproxy.techlib.cz/ehost/pdfviewer/pdfviewer?vid=1&sid=17bb6e80-04be-4476-af74-37c6fc7e2b71%40sessionmgr4009>

HOVMAND, Peter S. Community based system dynamics. New York: Springer, 2014. ISBN 978-1-4614-8763-0.

ISEE SYSTEMS, [cit. 3. dubna 2018]. Webová stránka [Online] <http://iseesystems.com/>

JANÍČEK, Přemysl a Jiří MAREK, 2013. Expertní inženýrství v systémovém pojetí. Praha: Grada. Expert (Grada). ISBN 978-80-247-4127-7.

JANIŠOVÁ, Dana a Mirko KŘIVÁNEK, 2013. Velká kniha o řízení firmy: [praktické postupy pro úspěšný rozvoj]. Praha: Grada. ISBN 978-80-247-4337-0.

KORN, Granino A. Advanced Dynamic-System Simulation: Model Replication and Monte Carlo Studies. 2013. ISBN 11-183-9735-5.

OSSIMITZ, Günther. 2000. Teaching System Dynamics and Systems Thinking in Austria and Germany. Dostupné na: <http://www.systemdynamics.org/conferences/2000/PDFs/ossimitz.pdf>

PELÁNEK, Radek, 2011. Modelování a simulace komplexních systémů: jak lépe porozumět světu. Brno: Masarykova univerzita. ISBN 978-80-210-5318-2.

RICHMOND, Barry. 1994. System Dynamics/Systems Thinking: Let's Just Get On With It Delivered at the 1994 International Systems Dynamics Conference. Dostupné na: <http://iseesystems.com/resources/articles/download/lets-just-get-on-with-it.pdf>

SENGE, Peter M.; STERMAN, John D. Systems thinking and organizational learning: Acting locally and thinking globally in the organization of the future. European journal of operational research, 1992, 59.1: 137-150. Dostupné na: <https://www.sciencedirect.com/science/article/pii/037722179290011W>

STERMAN, John D. Business dynamics systems thinking and modeling for a complex world. [Nachdr.]. Boston: Irwin/McGraw-Hill, 2000. ISBN 007238915x.

SYSTEMS-THINKING [online], [cit. 2018-04-25]. Dostupné z: <http://systems-thinking.org/>

VLČEK, Jaroslav, 1999. Systémové inženýrství. Praha: České vysoké učení technické. ISBN 80-010-1905-5.

VOJTKO, Viktor. 2005. Co je systémové myšlení?. Dostupné na http://sciencedynamics.net/images/Articles/CZ/sys_think.pdf

ZELENKA, Josef a Martina PÁSKOVÁ. Výkladový slovník - kognitivní věda. Hradec Králové: Gaudeamus (nakladatelství), 2005. ISBN 80-7041-355-7.

Zadání diplomové práce

Autor: Bc. Marek Zanker

Studium: I1600848

Studijní program: N6209 Systémové inženýrství a informatika

Studijní obor: Informační management

Název diplomové práce: Možnosti řešení dopravních problémů pomocí nástrojů systémové dynamiky

Název diplomové práce AJ: Possible solutions of traffic problems with the help of system dynamics tools

Cíl, metody, literatura, předpoklady:

Vytvoření alternativ pro převod dopravních systémů na diagram hladin a toků, porovnání a navržení úprav na případové studii.

Bude upřesněno při konzultacích se studentem.

Garantující pracoviště: Katedra informačních technologií,
Fakulta informatiky a managementu

Vedoucí práce: doc. Ing. Vladimír Bureš, Ph.D., MBA

Datum zadání závěrečné práce: 21.10.2014