

Vysoká škola ekonomická v Praze
Fakulta informatiky a statistiky



SimulTrek: simulační aplikace v MS Excel

BAKALÁŘSKÁ PRÁCE

Studijní program: Matematické metody v ekonomii

Studijní obor: Ekonometrie

Autor: Dominik Papírník

Vedoucí práce: prof. Ing. Mgr. Martin Dlouhý, Dr., MSc

Praha, květen 2024

Prohlášení

Prohlašuji, že jsem bakalářskou práci *SimulTrek: simulační aplikace v MS Excel* vypracoval samostatně za použití v práci uvedených pramenů a literatury.

V Praze dne 5. 5 2024

.....

Podpis studenta

Poděkování

Rád bych poděkoval panu prof. Ing. Mgr. Martinu Dlouhém, Dr., MSc. za čas, trpělivost, rady a odborné vedení mé bakalářské práce. Dále bych chtěl poděkovat všem uživatelům stránky <https://stackoverflow.com>, kteří odpovídají na dotazy a pomáhají řešit problémy ostatních uživatelů.

Abstrakt

Tato bakalářská práce se zabývá vývojem jednoduchého simulačního programu zaměřeného na modelování diskretních událostí. Doplněk nazvaný SimulTrek je navržený pro studenty a začátečníky v oblasti simulace a je integrován v MS Excel, což zajišťuje snadnou dostupnost a použitelnost pro uživatele s různými úrovněmi dovedností práce s počítačem. Doplněk nabízí intuitivní uživatelské rozhraní, a flexibilní možnosti nastavení simulace, které umožňují modelování různých scénářů a experimentování s parametry. Díky své jednoduchosti, je doplněk SimulTrek vhodným nástrojem pro akademické použití. Práce obsahuje podrobný popis funkcí doplňku SimulTrek, stejně jako návod na jeho použití a zhodnocení jeho přínosů pro uživatele.

Klíčová slova

Simulace, diskretní události, vzdělávání, modelování, integrace s MS Excel

Abstract

This bachelor thesis is focuses on the development of a simple simulation program aimed at modeling discrete events. The program named SimulTrek is designed for students and beginners in the field of simulation and is integrated with the MS Excel spreadsheet software, ensuring easy accessibility and usability for users with varying levels of computer skills. The program offers an intuitive user interface and flexible simulation settings, allowing for modeling of different scenarios and experimentation with parameters. Thanks to its simplicity, SimulTrek is suitable for academic use. The thesis includes a detailed description of SimulTrek's features, as well as a users handbook and an evaluation of its benefits.

Keywords

Simulation, discrete events, education, modeling, MS Excel integration

Obsah

Úvod	13
1 Teoretický úvod	15
1.1 Simulační modelování a jeho význam	16
1.2 Simulační programování a jeho využití	17
1.3 Diskrétní simulace událostí	18
1.4 VBA (Visual Basic for Applications) jako prostředí pro tvorbu simulačních programů	19
2 Analýza požadavků	21
2.1 Identifikace požadavků na simulační program	21
2.2 Specifikace funkcionalit programu SimulTrek	23
2.2.1 Distribuční typy	23
2.2.2 Zdroje	26
2.2.3 Aktivita Delay a její specializace	27
2.3 Stanovení cílového uživatelského segmentu	27
3 Návrh a architektura programu	29
3.1 Popis architektury simulačního programu SimulTrek	29
3.2 Návrh uživatelského rozhraní	31
4 Implementace	35
4.1 Popis klíčových částí kódu	35
4.1.1 Vytváření modelu pomocí skriptu	36
4.1.2 Průběh simulace	37
4.2 Testování a ladění programu SimulTrek	40
5 Příklady využití	41
5.1 Modely hromadné obsluhy	41
5.2 Modely řízení zásob	44
5.3 Simulace křižovatky	46
Závěr	49
Seznam použité literatury	51
A Průvodce doplňku SimulTrek	55
B VBA funkce pro generování hodnot distribucí	57

Seznam obrázků

2.1	Menu pro nastavení Delay akce.	22
5.1	Model jedné obslužní linky v SimulTrek.	42
5.2	Model paralelně uspořádané linky v SimulTrek.	43
5.3	Model sériově uspořádané linky v SimulTrek.	44
5.4	Model sériově uspořádané linky v SimulTrek s omezenou fortanou.	44
5.5	Model řízení zásob pro jeden stánek v doplňku SimulTrek.	45
5.6	Model křižovatky v doplňku SimulTrek	47
A.1	Průvodce doplňku SimulTrek v angličtině	55

Úvod

Simulace je v dnešní době nezbytným prvkem mnoha odvětví, kde může nabývat různých významů v závislosti na kontextu. Simulační modely jsou navrhovány tak, aby zachytily klíčové prvky a souvislosti v reálném systému, které jsou relevantní pro daný výzkum nebo účel. Tímto způsobem mohou simulace pomoci analyzovat a porozumět složitým systémům, které by jinak byly obtížné studovat nebo analyzovat v reálném čase. Pokud se jedná o systémy, které již existují, simulace nám nabízí možnost změnit vstupní charakteristiky, nebo i změnit systém jako celek, a poté pozorovat, jak změny ovlivňují náš konečný výsledek.

Simulace se například používá v podnikové praxi, kde nalézá uplatnění ve výrobních procesech a logistických systémech. Výhody simulace vynikají především ve více komplexních systémech, kde se klasické analytické přístupy stávají méně použitelnými.

V současné době existuje mnoho simulačních systémů. Některé jsou specializované na určité systémy, jako jsou například simulace dopravních toků nebo simulace chemických reakcí, jiné nabízejí více rozsáhlejší použití. Jedním z těchto programů je SIMUL8, který nám nabízí velmi široké možnosti pro vytváření komplexních systémů. Společně s programem SIMPROCESS, slouží jako inspirace pro vytvoření doplňku SimulTrek pro MS Excel.

Doplňek SimulTrek je jednoduchý simulační program, který využívá prostředí MS Excel. Cílem práce je vytvořit doplněk, který seznámí uživatele se základními principy simulačních programů a se simulací diskretních událostí. Praktická část slouží zároveň jako návod na používání doplňku, a také popisuje jednotlivé principy použité při tvorbě programu.

Práce se skládá z pěti kapitol. V první kapitole se věnuje podrobnější teorii o simulačních programech a principech při vytváření simulačních modelů. Popisuje také jednotlivé aspekty diskretních simulací událostí a jejich funkce v modelu. Druhá kapitola je zaměřena na analýzu požadavků, tedy co je důležité zohlednit při vývoji nového simulačního programu. Součástí této kapitoly je podrobné vysvětlení jednotlivých funkcionalit v programu SimulTrek. Třetí kapitola se věnuje prostředím, ve kterých se v rámci doplňku bude uživatel pohybovat. Čtvrtá kapitola popisuje implementaci doplňku ve VBA a poté se zaměřuje na klíčové části kódu, kde rozvádí fungování těchto částí. Pátá, a poslední kapitola, obsahuje příklady použití doplňku SimulTrek, společně s vysvětlujícími informacemi o tvorbě daných modelů.

1. Teoretický úvod

Simulace představuje nepostradatelnou metodologii pro řešení mnoha reálných problémů v mnoha oblastech. Její význam spočívá v analýze chování jednotlivých komponent systémů, kde si klade otázku 'co kdyby', což napomáhá při návrhu a optimalizaci skutečných systémů (1).

Samotné simulační metody nejsou vždy závislé na použití počítačů, či moderní technologie; mohou být realizovány i prostřednictvím simulovaných cvičení. Například v roce 2010 zdravotní vychovatelé implementovali v jednom zdravotním zařízení nový program, který kladl důraz na umístění absolventů ve specializovaných oblastech. Součástí tohoto programu byly simulace krizových situací, kde si absolventi mohli zlepšit a ověřit své dovednosti a rozhodovací schopnosti(2). Znáмым simulačním programem je také simulace pilotování letadel, která kombinuje praktické cvičení s využitím počítačových simulací. Tyto simulátory se obvykle dělí do dvou hlavních kategorií: simulace pro komerční lety a pro vojenské účely. Během tréninku si piloti zkouší v simulátoru jednotlivé krizové situace, které mohou nastat během letu. To jim umožní lépe reagovat, pokud tyto situace nastanou při reálném letu. (3).

Zahrnutí simulace do různých oblastí, naznačuje její široké uplatnění a pro správné fungování simulačního programu je důležité, si správně určit jednotlivé aspekty programu (viz kapitola 2). Bakalářské práce se zaměřuje převážně na počítačové simulace. Tyto typy simulace se dají rozlišit na čtyři simulační modelové přístupy, které vycházejí z myšlenky napodobení chování systému.

- *Simulace Monte Carlo*, je numerické řešení pravděpodobnostních i deterministických úloh pomocí statistického experimentu(4). Při této metodě je pro experimentování sestavena nová pravděpodobnostní úloha, která má shodné řešení s původní úlohou. Řešení takto získané má pravděpodobnostní charakter. Jde o statistický odhad, jehož přesnost roste s počtem pokusů. Za zakladatele této metody je považován Stanislaw Ulam, který v roce 1949 společně s Nicholasem Metropolisem vydaly článek *The Monte Carlo Method* v časopise Journal of the American Statistical Association, který popisuje danou metodu a její aplikace. S vývojem počítačů a jejich rychlosti rostlo i použití této metody. V dnešní době výsledky získané touto metodou mohou být velmi přesné.
- *Simulace diskrétních událostí* modeluje systémy jako provázanou síť dynamických a statických objektů. Princip simulace je založen na modelování jednotlivých událostí a jejich dopad na stav systému v konkrétních časových okamžicích. Nesleduje se model jako celek, ale je pozorován pouze ve specifických okamžicích, kdy se události odehrávají. Součástí diskrétní simulace událostí je několik hlavních prvků(5):
 1. **Události** - Jedná se o konkrétní a často náhodné akce, jako jsou například příchody entity do systému, dokončení úkolu v systému, příchod entity do fronty atd.
 2. **Čas** - Může být lineární nebo nelineární a je vymezený nebo nevymezený. V simulaci se pracuje převážně s diskrétními časovými okamžiky.
 3. **Fronta událostí** - V této frontě jsou uspořádány události podle času, kdy se mají

stát. Program si postupně vybírá události z této fronty a zpracovává je v pořadí.

4. **Stav systému** - Systém má určitý stav, který se může měnit v reakci na události. Model systému udržuje informace o aktuálním stavu a aktualizuje je při každé události.

Diskrétní simulace událostí je často používána pro modelování a analýzu systému, jako jsou fronty, dopravní sítě, výrobní linky, počítačové sítě etc. Tato technika poskytuje detailní pohled na chování jednotlivých částí systému a umožňuje zkoumání různých scénářů a strategií. Je velmi důležitým nástrojem pro optimalizaci výkonu a rozhodování v různých oblastech. V programu SimulTrek je využíván tento přístup k simulaci a podrobněji bude rozveden v podkapitole 1.3.

- *Systémová dynamika* je přístup, který vychází z předpokladu, že prakticky veškeré události, procesy a jevy, lze chápat jako projevy chování v systému. Základem systémové dynamiky je přesvědčení, že systém není pouhým jednorázovým dějem, který by začal, probíhal a následně skončil, ale dějem, který neustále probíhá. Klíčovým prvkem je zpětná vazba, která zaručuje, že systém není jednosměrný a ani statický. Procesy a vlivy probíhající v jednotlivých částech systému ovlivňují jeho okolí, a to následně skrze zpětnou vazbu působí opět na tyto části (6).
- *Multiagentní systémy* obsahují autonomní agenty, kteří se chovají podle předem definovaných pravidel. Základními charakteristikami multiagentního systému jsou (7):
 1. Každý agent má neúplné informace nebo schopnosti pro řešení problému, a proto má omezený úhel pohledu na daný problém.
 2. Neexistuje globální kontrola nad systémem
 3. Data a informace jsou decentralizované
 4. Výpočet je asynchronní.

Samotný systém se dále skládá ze tří částí: a) z množiny agentů, jejich charakteristik a pravidel chování; b) z množiny vztahů a způsobu komunikace mezi agenty; c) ze systémového okolí, se kterým jsou agenti také v interakci (5). V samotném systému je možné pozorovat vznik heterogenity a sebeorganizace.

1.1 Simulační modelování a jeho význam

Simulační modelování je proces vytváření abstraktního modelu reálného systému. Modelem rozumíme systém, který reprodukuje zásadní chování svého originálu (i když tento originál v daném okamžiku nemusí existovat). To znamená, že model nemusí nutně jen zjednodušeně replikovat svůj originál, ale může se jednat i o soustavu diferenciálních rovnic, simulovaných na počítači.

Z výše uvedeného si odvodíme dva typy modelů:

- *Fyzikální modely*, které vycházejí z principu podobnosti nebo analogie.
- *Abstraktní modely*, využívají symbolických prostředků, jako je matematický popis, logický popis nebo programovací jazyky.

Při tvorbě modelu je také důležité vybrat ty vlastnosti, které jsou nezbytné, a nelze je opomenout. Například pokud malíř bude malovat pokoj, tak barvy vybírá na základě estetické stránky než na základě funkcionality. Oproti tomu při konstrukci atomové bomby se technici zaměří na funkcionalitu než estetickou stránku. Při tvorbě modelu je tedy důležité si vybrat ty vlastnosti, které jsou pro systém důležité na úkor těch, které nejsou nezbytné.(8)

S modelováním se pojí i **experimentace**. Experimentace s modely je důležitou součástí vývoje a optimalizace modelů, protože umožňuje identifikovat nejlepší možné modely pro konkrétní problém, nebo úkol. Pro správnou experimentaci s modelem, je potřeba dodržovat několik klíčových kroků (9):

- *Definice problému a cíle* - Nejprve je důležité přesně definovat problém, který má model řešit a stanovit si cíle, kterých má dosáhnout.
- *Nastavení simulace* - Jedná se o zhodnocení nastavení, které jsou v modelu k dispozici a jakým způsobem je možné je upravovat.
- *Zohlednění omezení* - Tímto se rozumí omezení, které jsou nám dána na sběr dat ze simulace.
- *Efektivní interpretace výsledků* - Velmi důležitá část experimentace pro správné porovnání modelů.

Dalším a posledním krokem v modelování je samotná **implementace**. Správně navržený a validovaný model, má schopnost predikovat budoucí chování systému a poskytovat doporučení pro optimalizaci a zlepšení výkonu. Samostatné modely poté mohou mít využití například v rizikovém managementu, kdy je možné za pomoci modelu identifikovat a analyzovat rizika spojené s daným systémem, a navrhnout strategie pro jejich minimalizaci nebo eliminaci. Simulační modely nám při implementaci pomáhají minimalizovat ztráty jak ekonomické, tak i na zdraví jedinců.

1.2 Simulační programování a jeho využití

Simulační programování označuje proces tvorby počítačových programů, nebo modelů, které slouží k simulaci reálných systémů či procesů. Ty mohou být implementovány v různých programovacích jazycích jako je VBA (Visual Basic for Application), MATLAB, Python a etc.

V dnešní době jsou počítačové simulace běžné téměř ve všech oblastech. Mezi nejpopulárnější metody patří Monte Carlo a multiagentní simulace. Nicméně popularita těchto metod neznamená, že jsou jejich konkrétní implementace vždy identické. V současné době zahrnují počítačové simulace širokou škálu praktik podporovaných počítačem, které se mohou lišit v různých oblastech disciplíny, v cílech výzkumu a v kontextech aplikace. V komplexních fyzikálních nebo chemických systémech lze simulace chápat jako vyprávění příběhu. Vědci, kteří se pomocí simulace snaží pochopit určitý proces, který se odehrává v reálném čase, vytvářejí dynamický model, který generuje příběh, který „vypráví“ průběh událostí v simulaci (10).

V humanitních vědách je simulace poměrně nová, ale přináší do této disciplíny novou perspektivu na náš „žijící svět“ (11). Své uplatnění si již ale našla v technických oborech. Inženýři a architekti ji využívají při návrhu a stavbě budov, a také při testování jednotlivých komponent automobilů. Díky tomu bylo vyvinuto mnoho vizualizačních technik zaměřených na zobrazování geometrických povrchů. Simulace se také stala nedílnou součástí dnešní populární kultury, kdy zábavní průmysl využívá počítačové simulace při tvorbě sdílených světů v počítačových hrách.

Uživatelské rozhraní je důležitou součástí simulačních programů. Většina simulačních programů používá metodu „click and drag“, kdy stačí kliknout na jednotlivé moduly a přetažením je integrovat do modelu.

1.3 Diskrétní simulace událostí

Jak již bylo popsáno v úvodu této kapitoly. Diskrétní simulace událostí se zaměřují na jednotlivé události v našem modelu. Tedy místo toho aby nás zajímal čas jako celek, se zaměřujeme na jednotlivé události v simulaci. Tyto události jsou tvořeny kombinací *entit* a *aktivit*.

Entity představují objekty, které prochází systémem a vykonávají určité činnosti. Jednotlivé entity mohou mít své vlastní charakteristiky a v průběhu interakcí se systémem se mohou měnit na jiné.

Aktivity jsou procesy, na kterých se entity podílejí. Tyto procesy mohou být rozděleny do tří základních kategorií:

- **GENERATE** (generovat, vytvořit) - Jedná se o proces sloužící pro generování entit do systému. Generace těchto entit vychází z předem nastavené distribuce, která je dána uživatelem. Některé programy umožňují nahrání předem definovaných intervalů mezi příchody entit. Tato funkce umožňuje získání přesnějších výsledků pro sledovaný systém.
- **DELAY** (zpoždění, prodleva) - Tento druh představuje zdržení entity po dobu určité aktivity. Trvání aktivity je konstantní nebo náhodné. Součástí těchto aktivit mohou být také *zdroje*. Ty mohou být přiděleny jedné nebo více aktivitám a slouží jako aktivující mechanismus. Každý zdroj disponuje kapacitou a množstvím. Kapacita zdroje omezuje maximální počet entit, které mohou být obslouženy pomocí tohoto zdroje současně, zatímco množství stanovuje limit počtu entit, jenž mohou být u aktivity obslouženy.
- **DISPOSE** (vyřídít, dohotovit) - Odstraňuje entity ze systému a slouží jako konečná hranice v modelu.

Pro lepší pochopení principu diskretní simulace událostí, je představen příklad malého obchodu s jednou pokladnou. V tomto systému identifikujeme tři základní aktivity: příchod zákazníka, obsluha u pokladny a následný odchod z obchodu. Klíčovými prvky tohoto modelu jsou entity, které v tomto případě představují jednotlivé zákazníky.

Po vytvoření takového modelu je možné spustit simulaci. Během průběhu simulace můžeme

identifikovat čtyři základní stavy. Prvním stavem jsou příchody zákazníků do obchodu. Zákazník po příchodu přejde k pokladně, kde mohou nastat dva scénáře: buď je pokladna volná, a tedy zákazník je obsloužen, nebo je pokladna obsazená, a zákazník se zařadí do fronty a čeká než na něj přijde řada. Posledním stavem je pak následující odchod z obchodu.

Po skončení simulace se vyhodnocují jednotlivé aktivity. Toto vyhodnocení záleží už na uživateli, který model vytvořil. Simulační programy velmi často nabízí obecný přehled jednotlivých aktivit a zdrojů. Zde se objevuje například vytížení zdrojů, čekací doba nebo průměrná doba entity v systému.

Režim fronty určuje způsob přechodu požadavků z fronty do obsluhy. Základní typy jsou (12):

- **FIFO** (first-in/ first-out) představuje situaci, kdy jednotlivé entity přecházejí z fronty do obsluhy v tom pořadí v jakém přišly. Někdy se pro tento režim fronty používá zkratka *FCFS* (first-come / first-served). Tento typ fronty se nejvíce vyskytuje v případech hromadné obsluhy (samoobsluha, banka, benzínové stanice atd.)
- **LIFO** (last-in / first-out) jedná se o opačný režim fronty. Entita, která přišla do fronty jako poslední, je obsloužena první. Tento typ fronty je možné ilustrovat pomocí výrobní linky, kdy jsou jednotlivé polotovary na sebe skládány a tedy je systém odebírá v opačném pořadí než přišly. Alternativním označením tohoto systému může být *LCFS* (last-com / first-served).
- **SIRO** (selection in random order) je náhodný výběr přechodu z fronty do obsluhy
- **PRI** určuje přechod z fronty do obsluhy podle zadaných priorit. V tomto režimu jsou tedy entity obsluhovány podle předem zadané priority. Pokud se ve frontě nachází entity se stejnou prioritou, jsou obsluhovány v předem zvoleném režimu (například FIFO).

Doplňek SimulTrek má pevně nastavený typ fronty na FIFO, a nepodporuje tvorbu vlastních entit. Důvodem je zaměření doplňku místo entit na zdroje, které jsou v doplňku rozšířeny o speciální vlastnosti (viz podkapitola 2.2.2).

1.4 VBA (Visual Basic for Applications) jako prostředí pro tvorbu simulačních programů

VBA je skriptovací jazyk založený na Visual Basic, který je integrovaný do produktů Microsoft Office, jako je Excel, Word, PowerPoint etc. Tento jazyk umožňuje uživatelům automatizovat úkoly, a vytvářet vlastní funkce, nebo makra přímo v těchto aplikacích. Vzhledem k jeho snadné dostupnosti a širokému využití, se VBA často používá pro analýzu dat nebo i vytváření automatizovaných reportů. Kvůli těmto důvodům, může být VBA, i vhodným simulačním nástrojem. Klíčové aspekty VBA jako prostředí pro tvorbu simulačních programů zahrnují:

- **Integrace s Microsoft Excel** - Samotný program Microsoft Excel (zkráceně MS Excel) je široce používaným nástrojem pro ukládání nebo analýzu dat. Program má velkou škálu přednastavených funkcí a díky nim nemá velké bariéry vstupu pro nové

uživatele. Pro vytváření jednoduchých simulačních modelů si je možné vystačit jen s těmito funkcemi.

- **Flexibilita a rozšířenost** - Pomocí VBA je možné vytvářet funkce a procesy vytvořené přímo pro potřeby uživatele, což poskytuje velkou míru flexibility pro tvorbu simulačních programů, a jejich přizpůsobení pro konkrétní zadání.
- **Snadná distribuce a sdílení** - Programy vytvořené ve VBA mohou být velmi snadno sdíleny s ostatními uživateli. Můžou se sdílet soubory MS Excel s předem nastavenými makry, kdy po otevření souboru se mohou rovnou využívat. Dalším způsobem sdílení těchto programů je pomocí Add-Ins. Oproti předchozí metodě nemusíme otvírat předem specifický soubor MS Excel, ale stačí si v nastavení přidat program jako Add-In. Následně je možné si ho otevřít a používat v jakémkoliv souboru MS Excel.
- **Dostupnost dokumentace a zdrojů** - Samotný program VBA existuje již mnoho let, a tedy existuje mnoho on-line zdrojů, tutoriálů a komunit, které poskytují podporu a informace o používání VBA. To velmi usnadňuje práci s programem.

Jak si tedy můžeme všimnout VBA je silným jazykem pro vytváření simulačních programů, protože obsahuje velké množství užitečných nástrojů, které pomáhají při vývoji těchto programů. Některé funkcionality ve VBA, ale chybí, nebo jsou zbytečně zmatečné. Jedná se například o velmi zdlouhavé odhalování chyb, nebo překlepů v programu. Tyto nedostatky mohou být opraveny za pomoci integrace externích knihoven a API. Tyto externí knihovny se v doplňku SimulTrek nepoužívají.

2. Analýza požadavků

Jak bylo zmíněno v předchozí kapitole, simulační programy jsou velice universálním nástrojem pro různé disciplíny. Z toho důvodu je při vývoji simulačního programu důležité si nastavit správné cíle a požadavky, abych bylo zaručené správné chování vytvořeného programu.

Důležitým segmentem jsou i uživatelé. Ne každý program je určený pro všechny uživatele. Některé programy jsou více specializované pro určitou část uživatelů, kteří mají již nějaké teoretické znalosti v rámci oboru. Příkladem můžou být již zmíněné letové simulátory, které obsahují velké množství přepínačů a tlačítek s předem nastavenými funkcemi. Tedy uživatel, který nemá znalost jakou funkci tyto přepínače mají, nebude moci simulátor použít pro správné simulování pilotování letadla. Některé programy jsou ale více obecné, a snaží se uživatelům usnadnit jejich používání. Využívají ikon, nebo obrázků, pro jednotlivé entity nebo aktivity v simulaci. Dovolují 3D/2D nastavení prostředí, pro lepší orientaci v modelu. To vede k menší bariéře vstupu pro nové uživatele, ale také k tomu, že tyto programy poté mohou ztrácet efektivitu, nad více specializovanými simulačními programy, pro určitý typ problémů.

2.1 Identifikace požadavků na simulační program

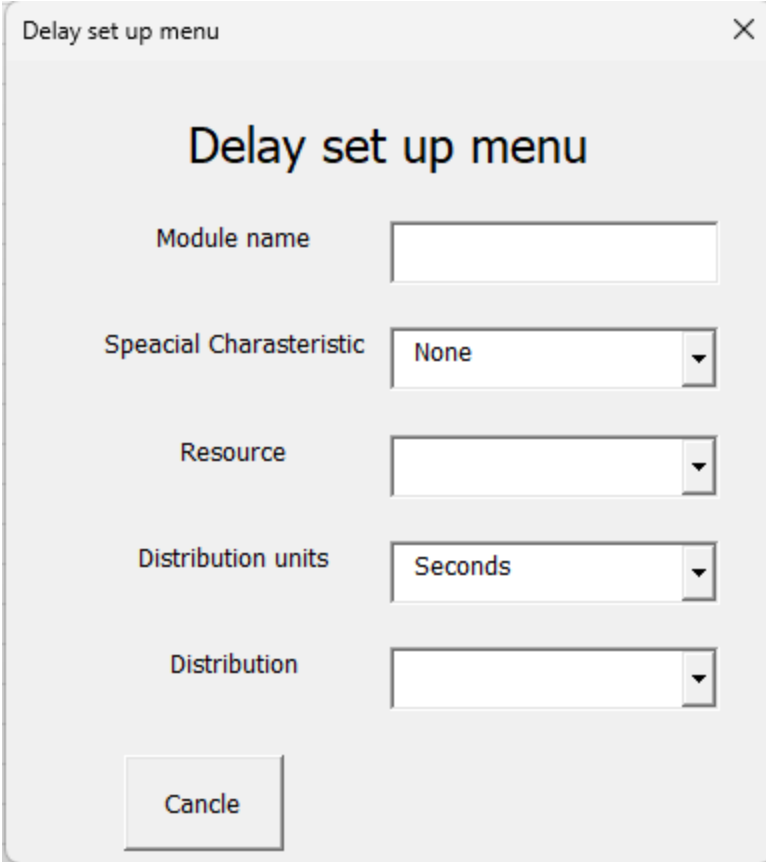
Existuje mnoho simulačních programů, nabízejících velké množství funkcí, pro zpracovávání simulačních modelů. Velkým faktorem je ale samozřejmě cena daných programů. I když některé programy mají zkušební verzi, která nabízí jen omezené možnosti pro tvorbu simulačních modelů a jejich použití je občas velmi konkrétní. Jelikož některé funkcionality nejsou uživateli přístupné, může nastat situace, kdy se nezkušený uživatel ztratí ve velkém množství funkcí a položek, i když některé nemůže vůbec použít.

Tyto nedostatky mě vedly k vytvoření programu čistě pro uživatele, kteří nemají se simulačními programy velké zkušenosti a hledají něco co je přivítá do světa simulací. Doplněk SimulTrek je open-source což znamená, že veškeré fungování programu je volně dostupné pro všechny uživatele. Tato vlastnost doplňku SimulTrek chce otevřít dveře možným úpravám nebo i studování jak doplněk SimulTrek funguje pro uživatele, kteří mají o tyto vědomosti zájem. Celý doplněk SimulTrek je dále napsán v anglickém jazyce a to z několika důvodů. Jedním z důvodů je dostupnost pro větší množství uživatelů, a druhým je problém špatného zobrazování českých slov s čárkami a háčky v rámci vyskakovacích oken. Součástí doplňku SimulTrek je i zjednodušený průvodce jednotlivých funkcí, který je také napsaný v anglickém jazyce. Průvodce můžeme najít v příloze A.

Uživatelské rozhraní se skládá z jedenácti tlačítek, jímž jsou přiřazeny různé funkcionality, což představuje minimalistický design uživatelského rozhraní. Doplněk SimulTrek využívá také UserForm, což je součást programu VBA, umožňující vytvářet a přizpůsobovat vyskakovací

okna pro různé účely (ukázko okna na obrázku 2.1). Tato okna jsou dynamická, a přizpůsobují se informacím, které uživatel potřebuje zadat. Například při vytváření Delay aktivity se zobrazí okno, ve kterém uživatel musí vyplnit určité informace. Po doplnění distribuce se okno přeformátuje, a zobrazí se nové pole pro doplnění hodnoty parametru. Jelikož se počet parametrů liší pro různé distribuce, přidává se do okna i různý počet těchto polí.

Rychlost simulačního programu představuje další klíčovou vlastnost. Delší doba trvání simulace v simulačním programu znamená vyšší časovou náročnost při provádění různých experimentů s modelem. Zde se však naráží na určité nevýhody jazyka VBA. Vzhledem k jeho univerzálnosti, může celková rychlost simulačního procesu utrpět, kvůli nedostatečné optimalizaci. Doplněk SimulTrek je tedy navržen pro menší a středně komplexní modely, a není určen jako nástroj pro simulace s velkým počtem aktivit. Jeho hlavním účelem je poskytnout prostředek pro základní pochopení principů simulace a pro seznámení se s různými objekty, které mohou v simulaci existovat. Jednoduchost doplňku SimulTrek navíc přispívá k nižší chybovosti při tvorbě modelů. V rámci doplňku jsou rovněž implementovány různé kontroly, které zajišťují správné fungování, a upozorňují a vzdělávají uživatele o správném používání doplňku SimulTrek.



Delay set up menu

Delay set up menu

Module name

Speacial Charasteristic

Resource

Distribution units

Distribution

Cancle

Obrázek 2.1: Menu pro nastavení Delay akce.

2.2 Specifikace funkcionalit programu SimulTrek

Samotný doplněk SimulTrek je zaměřený na podnikové procesy, jako jsou systémy hromadné obsluhy. Obsahuje základní vlastnosti pro tvorbu těchto modelů s více rozvinutými vlastnostmi pro zdroje. Součástí doplňku SimulTrek je i vlastnost *Forced Clear*, která zajišťuje, že entity, které se nachází v systému po uplynutí simulační doby, jsou nuceny systém opustit (pokud je to pro ně možné). Vlastnost v této situaci prodlužuje dobu simulace, dokud entity systém neopustí.

Další součástí doplňku SimulTrek jsou animace transitu entit mezi aktivitami. Tato vlastnost umožňuje uživateli pozorovat a kontrolovat jednotlivé přechody, které entity vykonávají. Animace velice zpomalují rychlost simulace, z toho důvodu je možné animace zastavit v průběhu simulace pomocí tlačítka **Stop**. To povede k zastavení animací, ale simulace bude nadále pokračovat bez nich.

2.2.1 Distribuční typy

Mezi základní funkce doplňku SimulTrek patří generování náhodných čísel pomocí šesti základních distribucí. Tyto distribuce generují náhodné číslo pomocí předem daných parametrů, které je následně zpracováno pro určení času dané aktivity. Těmito distribucemi jsou (5):

- *Exponenciální distribuce* je pravděpodobnostní distribuce, ta se často používá k popisu doby mezi událostmi procesu. Je typicky charakterizována jedním parametrem, často označovaným jako λ (lambda), který je v případě modelování příchodů požadavků interpretován jako intenzita příchodů. Pravděpodobnostní hustota funkce exponenciální distribuce je dána vztahem:

$$f(x|\lambda) = \lambda e^{-\lambda x}$$

kde $x > 0$ je náhodná veličina, $\lambda > 0$ je parametr distribuce a e je Eulerovo číslo. Tato funkce představuje pravděpodobnost, že náhodná veličina X bude mít hodnotu mezi $x + dx$, pokud je X exponenciálně distribuována s parametrem λ . Samotná generace této distribuce, je poté v programu VBA vypočítána pomocí vzorce:

$$x = (-1/\lambda)\ln(1 - r)$$

kde r je náhodné číslo vygenerované pomocí VBA.

- *Rovnoměrná distribuce* je základem pro generování dalších náhodných čísel při $R(0,1)$, ale dá se také využít i pro zachycení délky trvání činnosti. Pro spojitou rovnoměrnou distribuci definovanou na intervalu $[a,b]$, je pravděpodobnostní hustota funkce konstantní uvnitř tohoto intervalu a nulová mimo něj. Matematicky lze tuto distribuci vyjádřit jako:

$$f(x|a,b) = \frac{1}{b - a}$$

kde $f(x|a,b)$ je pravděpodobnostní hustota funkce, x je hodnota náhodné veličiny a a, b jsou koncové body intervalu, přičemž $a < b$. Generovat veličinu rovnoměrné distribuce můžeme pomocí vzorce:

$$x = a + (b - a)r$$

kde r je náhodné číslo vygenerované pomocí VBA.

- *Normální distribuce*, často nazývaná jako Gaussova distribuce, je jedním z nejdůležitějších pravděpodobnostních rozdělení ve statistice. Často se používá pro zachycení chyby při fyzikálních a ekonomických pozorování. Může být použita i pro generování dob trvání činnosti. Je ale potřeba volit střední hodnotu a rozptyl tak, aby byla minimalizována možnost generování záporného čísla. Pokud je vygenerována záporná hodnota, tak doplněk SimulTrek ji interpretuje jako nulovou.

Je charakterizována svou symetrií kolem střední hodnoty μ a zvonovitým tvarem hustoty pravděpodobnosti, kde maximum hustoty pravděpodobnosti je právě v bodě μ . Rozptyl σ^2 udává, jak moc jsou hodnoty proměně rozptýleny od střední hodnoty. Čím máme vyšší rozptyl, tím širší je rozpětí hodnot. Její hustota pravděpodobnosti je dána funkcí:

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

kde x je náhodná proměnná, μ je střední hodnota (průměr) distribuce, σ^2 je rozptyl (kvadratický rozptyl) distribuce, π je matematická konstanta π a e je Eulerovo číslo.

Pro generování těchto veličin použijeme algoritmus:

1. Generuj dvě náhodné veličiny r_1 a r_2 ;
 2. $V_1 = 2r_1 - 1$, $V_2 = 2r_2 - 1$, $W = V_1^2 + V_2^2$;
 3. Jestliže $W \geq 1$ opakuj krok 1, jinak $Y = ((2 - \ln W)/W)^{1/2}$, $z_1 = V_1 Y \sigma + \mu$, $z_2 = V_2 Y \sigma + \mu$;
 4. Konec, z_1 a z_2 jsou dvě hodnoty rozdělení $N(\mu, \sigma^2)$.
- *Geometrická distribuce* se používá k modelování času do události „prvního úspěchu“, kde každý pokus má dvě možnosti: úspěch nebo neúspěch. Tedy pomocí této distribuce je možné zjistit počet neúspěšných pokusů, které se vyskytly před prvním úspěchem. Tato distribuce má jediný parametr p , který představuje pravděpodobnost úspěchu v jednom pokusu. Pravděpodobnostní funkce je dána vzorcem:

$$P(X = k) = (1 - p)^{k-1}p, \text{ pro } k = 0, 1, 2, \dots \text{ a } 0 < p < 1$$

kde X je náhodná veličina představující počet neúspěšných pokusů před prvním úspěchem, p je pravděpodobnost úspěchu v jednom pokusu a k je pořadí neúspěšného pokusu. Mezi základní vlastnosti geometrické distribuce se zahrnuje nezávislost (každý pokus je nezávislý na ostatních), Stálá pravděpodobnost (pravděpodobnost úspěchu p zůstává stejná pro všechny pokusy) a nekonečné rozpětí (distribuce má nekonečné rozpětí, protože počet neúspěšných pokusů může dosáhnout nekonečna). Geometrickou distribuci budeme generovat pomocí algoritmu:

1. $k = 0$;
2. Generuj náhodné číslo r ;
3. Jestliže $r < p$, jdi na krok 5;

4. $k = k + 1$, jdi na krok 2;
 5. Konec, v k je hodnota geometrické distribuce.
- *Binomická distribuce* popisuje rozdělení počtu nastoupení příznivého jevu v n nezávislých realizacích náhodného pokusu. Má dva parametry n (počet nezávislých pokusů) a p (pravděpodobnost úspěchu v jednom pokusu). Pravděpodobnostní funkce binomické distribuce je dána vzorcem:

$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}, \text{ pro } k = 0, 1, 2, \dots \text{ a } 0 < p < 1$$

kde X je náhodná veličina představující počet úspěchů v n pokusech, k je počet úspěchů, p je pravděpodobnost úspěchu, a $\binom{n}{k}$ je binomický koeficient, který vyjadřuje počet možných kombinací k úspěchů v n pokusech. Dále platí že n je přirozené číslo a $0 < p < 1$. Binomickou distribuci generujeme pomocí algoritmu:

1. $x = 0$;
 2. Pro $i = 1, 2, \dots, n$ opakuj kroky 3 a 4;
 3. Generuj náhodné číslo r ;
 4. Jestliže $r \leq p$, pak $k = k + 1$;
 5. Konec, v k je hodnota binomické distribuce.
- *Poissonova distribuce* se v simulaci používá pro generování počtu příchodů entit do systému, počtu vadných výrobků, počtu vad na jeden výrobek nebo počtu přerušení provozu za danou jednotku. Má jediný parametr λ , který tuto distribuci spojuje s exponenciální. Pravděpodobnostní funkce je dána vzorcem:

$$P(X = k) = \frac{e^{-\lambda} \lambda^k}{k!}$$

kde X je náhodná veličina představující počet událostí, k je počet událostí v daném intervalu, λ je průměrný počet událostí v tomto intervalu a e je Eulerovo číslo. Algoritmus pro generování této distribuce je:

1. $x = 0$, $A = e^{-\lambda}$, $B = 1$;
2. Generuj náhodné číslo r ;
3. $B = Br$;
4. Když $B \geq A$, pak $k = k + 1$, jdi na krok 3;
5. Konec, k je hodnota Poissonovy distribuce.

Všechny funkce pro generování hodnot pomocí VBA jsou k dispozici v příloze B.

Distribuce se rozdělují na dva základní typy, a to na **spojitou** a **diskrétní** distribuci, které se liší v tom, jaké hodnoty může náhodná veličina nabývat. Diskrétní distribuce popisuje pravděpodobnost jednotlivých diskrétních hodnot, které jsou oddělené a konkrétní. V našem případě se jedná o geometrickou distribuci, binomickou distribuci a Poissonovu distribuci. Spojitá distribuce na druhou stranu popisuje pravděpodobnosti pro spojitý rozsah hodnot, kde pravděpodobnostní hustota funkce je spojitá. Zbylé tři distribuce, tedy exponenciální, rovnoměrná a normální distribuce, jsou tohoto typu.

Existují i distribuce, které mohou kombinovat prvky obou typů. Například smíšená distribuce obsahuje jak hodnoty diskrétní, tak spojitě. Příkladem může být distribuce, popisující časové

meze doručení zásilek (počet dnů potřebných k doručení je diskrétní proměnná, zatímco skutečné časy doručení jsou spojité proměnné).

2.2.2 Zdroje

Zdroje jsou definované jako objekt, který může být omezený, využitý nebo obsazený entitami v simulovaném modelu. Samotné zdroje mohou být ale cokoliv co je potřeba při tvorbě modelu. Příkladem mohou být prodavačky na překážkách nebo počty krabic připravené na balení produktů z výroby. V některých simulačních programech je možnost společně se zdroji nastavit náklady spojené s jejich užíváním. Tato vlastnost v doplňku SimulTrek sice není k dispozici v základním nastavení, ale jelikož jsou veškerá data ze simulace ukládána, může si uživatel tyto informace jednoduše zjistit.

Každý zdroj má určité vlastnosti v rámci své aktivity. Doplňěk SimulTrek nabízí možnost přiřadit jen jeden zdroj do jedné aktivity, a to jen pro aktivity Delay. Ve více komplexnějších programech je ale možnost zdroje kombinovat a přiřazovat jiným druhům aktivit. Zdroje v doplňku SimulTrek mají vlastnosti vždy specifické, v rámci aktivity, ke které jsou přiřazeny. Těmito vlastnostmi jsou:

- *Capacity* (kapacita zdroje) říká kolik entit může v být obsluhovány zdrojem ve stejnou chvíli bez tvorby front. Může nabývat hodnoty 1 a větší.
- *Count* (počet jednotek): udává jestli se má zdroj omezenou spotřebu ve vybrané aktivitě. Ve výchozím nastavení, je pro všechny nově vytvořené zdroje, dána hodnota "Inf", která nám označuje zdroj, který není omezený. Tuto hodnotu můžeme přepsat kladným celým číslem pro omezení zdroje v aktivitě. Jak se zdroj spotřebovává poté záleží na typu jednotky.
- *Units* (typ jednotky): určuje jakým způsobem se jednotky spotřebovávají v rámci aktivity. Tento typ jednotky je rozdělen na časové a početní. Pro početní máme jen typ „count“. Ten vyjadřuje, že pro každou entitu, která se zpracuje v aktivitě je zpracována jedna jednotka zdroje. Časový typ dělíme na hodiny, minuty a vteřiny. Oznamuje, že zdroj bude fungovat v rámci aktivity určitou dobu, předtím než se spotřebuje, a to bez ohledu na to, kolik entit aktivitou projde.
- *Order* (pořadí aktivity): říká v jakém pořadí daná aktivita bude obsloužena zdrojem. Tato vlastnost je specifická pro zdroje, napojené na dvě nebo více aktivit.

V rámci doplňku SimulTrek existují čtyři typy zdrojů, které můžeme přiřadit k aktivitám. Jedná se o:

- *Fixed* (fixní) typ může být přiřazen jen jedné aktivitě. Jeho využití spočívá v možnosti nastavení omezení a kapacit pro danou aktivitu. Pokud se daný zdroj vyčerpá, aktivita se zavře, a již nebude obsluhovat další entity.
- *Flow* (plovoucí) typ se pohybuje mezi dvěma, či více aktivitami. Na začátku simulace se přiřadí do první aktivity v pořadí, a jakmile se vypotřebuje, tak se přemístí do další aktivity, pokud existuje. V situaci, kdy je poslední aktivita v systému vypotřebována,

odejde tento zdroj ze systému.

- *Cycle* (cyklický) typ funguje na podobném principu jako plovoucí. Jediným rozdílem je jeho vlastnost opakování. Tedy pokud se zdroj nachází v poslední aktivitě v pořadí, po spotřebování této aktivity se přemístí opět na začátek, a začne cyklus znova.
- *Priority* (prioritní) je speciální typ, který přeskakuje mezi aktivitami na základě příchoď entit. Tedy tento typ zdroje neustále kontroluje jaká aktiva, má přednost pro zpracování. Například budeme mít tři aktivity A1, A2 a A3. Zdroj bude momentálně aktivní v A2, kde je i fronta a ostatní jsou prázdné. Nově příchozí entita přijde do aktivity A1. Jelikož se jedná o aktivitu s větší prioritou, přesune se zdroj z A2 do A1 (poté co do obslouží entitu již rozpracovanou v A2). Jakmile obslouží entitu v A1 vrátí se zpět do A2. Pokud ale nová entita přijde do A3 nic se nezmění, jelikož tato aktivita není prioritní oproti A2.

Pokud nastane situace, kdy u prioritních aktivit byly vypočteny jednotky, tak tato aktivita nebude již preferována, i kdyby v ní byly nějaké entity.

2.2.3 Aktivita Delay a její specializace

Každá aktivita Delay má dva stavy ve kterých se může nacházet. Buď je **aktivní** nebo **zavřená**. Aktivní jsou všechny aktivity, ve kterých je nevyčerpaný zdroj, nebo nejsou přiřazené ke zdroji. Tyto aktivity zpracovávají entity, přicházející v rámci simulace. Zavřené simulace naopak nemohou příchozí entity zpracovat. Tento stav může nastat jak v případě vyčerpání zdroje, tak i v případě, že přiřazený zdroj sice není zrovna v pořadí.

Doplňek SimulTrek nám nabízí dva způsoby pomocí kterých můžeme posunovat entity dále modelem, i když nebyly zpracovány aktivitou. Jedná se o *Go Throught* a *Go Throught (Empty)*. První tento způsob pouští všechny entity aktivitou, bez ohledu na to, jestli je daná aktivita vyčerpaná. Naopak druhý způsob pouští entity aktivitou jen v případě vyčerpání zdroje u dané aktivity. Tyto způsoby jsou součástí specializovaných vlastností aktivity Delay. Třetí a poslední specializací, je *No Queue*. Tento způsob zajišťuje to, že pro danou aktivitu není možnost tvorby front. Tedy pokud se nějaká entita rozhodne jít do této aktivity, může tak učinit jen pokud je v aktivitě volné místo. Pokud není, je nucena zůstat ve své současné aktivitě, a počkat do doby než se místo uvolní.

2.3 Stanovení cílového uživatelského segmentu

Stanovení cílového uživatelského segmentu je klíčové, pro úspěšné šíření a přijetí doplňku SimulTrek. Vzhledem k jeho zaměření na simulaci diskrétních událostí, a jeho jednoduchosti, jsou cílovým uživatelským segmentem především studenti a začátečníci v oblasti simulací, kteří se chtějí seznámit s tímto specifickým typem simulací, a prakticky je vyzkoušet. Doplňek SimulTrek je vhodný pro studenty univerzitních oborů jako informatika, management a ekonomie, hledající nástroj, který je intuitivní, snadno použitelný a zároveň dostatečně výu-

kový v oblasti simulace diskretních událostí. Díky integraci s MS Excel je program přístupný i pro uživatele se základní znalostí práce s tabulkovými procesory. Doplněk SimulTrek nemusí být užitečný jen pro studenty, ale mohou ho využívat i učitelé, nebo výzkumníci zaměřující se na simulaci diskretních událostí, kteří potřebují demonstrační nástroj pro výuku nebo experimentaci.

Aby program splňoval potřeby uživatelského segmentu, byl navržen s ohledem na specifické požadavky studentů a začátečníků. Následující cíle byly zohledněny při vývoji programu:

- **Jednoduchost použití** - Doplněk SimulTrek nabízí intuitivní uživatelské rozhraní, které umožňuje snadné nastavení parametrů simulace a spuštění simulace bez složité konfigurace.
- **Výukový charakter** - Návrh doplňku SimulTrek a bakalářská práce je zpracována tak, aby umožnila uživatelům porozumět základním principům simulace a procesům spojeným s nimi.
- **Interaktivita** - Součástí doplňku SimulTrek jsou jednoduché animace přechodu entit mezi aktivitami, které pomáhají uživatelům kontrolovat průběh simulace.
- **Spolehlivost a stabilita** - Doplněk SimulTrek je spolehlivý a stabilní a uživatelé mohou spoléhat na přesné, a konzistentní výsledky.

3. Návrh a architektura programu

Doplněk SimulTrek je kompletně implementován v programovacím jazyce VBA, bez použití externích knihoven a API. Při vývoji doplňku byl využit pouze nástroj *Custom UI Editor for Microsoft Office*. Tento nástroj slouží k úpravě a práci s pásem karet v MS Excel. Pomocí tohoto nástroje, byl vytvořen pás karet obsahující veškerá potřebná tlačítka, pro přehledné uživatelské rozhraní. Součástí tohoto procesu bylo také možné vložení vlastních ikon pro tlačítka.

Doplněk SimulTrek dále využívá listy pro tvorbu modelů a ukládání dat získaných během simulace. Během simulace jsou vytvořeny dva listy: *Data* a *Summary*. Na listu *Data*, jsou uloženy podrobné informace o jednotlivých entitách, jako jsou časy jejich příchodů, odchodů, nebo délka pobytu ve frontě. Všechny časové údaje jsou vyjádřeny ve formátu *den:hodiny:minuty:vteřiny*, například hodnota 1:00:15:15 značí, že událost nastala druhého dne (první den má index nula) v 15 minut a 15 sekund. Tento formát není přímo interpretabilní jako číslo, avšak pro převod na číslo lze využít funkci **TimeValueCreation**, která hodnotu převede na počet sekund ($TimeValueCreation(1 : 00 : 15 : 15) = 87315s$). Na listu *Summary* jsou pak zaznamenány obecné informace o aktivitách. Více podrobností o obsahu těchto listů je uvedeno v podkapitole 3.2.

3.1 Popis architektury simulačního programu SimulTrek

Doplněk SimulTrek využívá prostor listů v programu MS Excel jako základní prostředek pro tvorbu modelu. Uživatel zde může přidávat jednotlivé aktivity a cesty, přičemž veškeré informace zadávané během tvorby modelu jsou ukládány do specializovaného listu pro následné využití při simulaci.

Pro vytvoření aktivit je využíván objekt UserForm, který umožňuje tvorbu uživatelských rozhraní v MS Excel. UserForm je okno, které obsahuje různé ovládací prvky, jako jsou textová pole nebo seznamy, a umožňuje interakci uživatele s aplikací. Díky tomu je možné vytvářet uživatelsky přívětivé formuláře a dialogová okna, která postupně zobrazují potřebná pole k vyplnění. Pro tyto pole je možné vytvářet speciální procesy, které kontrolují vložené informace. Jednou z těchto kontrol je kontrola hodnoty u parametrů. Ta kontroluje, že uživatelem zadaná hodnota je numerická a větší nebo rovna nule.

Po zadání informací a uzavření okna (tlačítko OK), se informace uloží do specializovaného listu, a v otevřeném listu se vytvoří aktivita. Aktivita se vždy vytvoří v buňce, kterou má uživatel v listu označenou a je zobrazena jako obdélník se zakroucenými rohy. Každá aktiva má přiřazenou baru podle typu: zelenou pro Generate, modrou pro Delay a červenou pro Departure. V rámci modelu je možné jednotlivé aktivity přesouvat. Jelikož ale objekty, které reprezentují aktivity, mají přiřazené vlastní makra, je nutné nejdříve aktivitu roz-kliknout

pomocí pravého tlačítka. Tím se aktivita označí a poté je možné jí přesunout.

Pro propojení aktivit cestou je potřeba využít tlačítko Connection, které se nachází v pásu karet. Tento způsob není moc intuitivní, jelikož většina simulačních programů nabízí možnost spojit aktivity v modelu, ale tento přístup pro propojení aktivit je pro doplněk SimulTrek nutný. Důvodem je velice obtížné sledování změn, které uživatel dělá v rámci sešitu samotného s vestavěnými funkcemi MS Excel.

V rámci tvoření cest mezi aktivitami, je možnost nastavit délku trvání přechodů mezi nimi. Pokud tyto informace nejsou nastaveny, doplněk SimulTrek realizuje přesuny okamžitě. Jestli existuje více jak jedna cesta z aktivity, je možnost nastavit, s jakou pravděpodobností si entita tuto cestu vybere. Pokud se stane, že celkový součet pravděpodobností se nerovná jedné, tak se v rámci simulace tyto pravděpodobnosti přepočítají. Toto přepočítání zajišťuje vzorec:

$$P(x) = \frac{k_i}{\sum_{i=1}^n k_i}$$

kde k_i jsou nastavené pravděpodobnosti a n je počet cest, které vedou z aktivity.

Předtím než se vytvoří cesta mezi dvěma aktivitami, je potřeba najít podle jakých bodů budou spojeny. Objekty, které reprezentují aktivity mají čtyři body, které se nachází uprostřed každé hrany objektu. Je tedy potřeba dohledat nejkratší možnou cestu mezi body aktivit. Tato cesta se hledá pomocí rovin omezené funkcemi. Celkově má každá aktivita jednu rovinu pro každý její bod. Vybraný bod je ten, u kterého rovina obsahuje střed druhé aktivity. Následující kód zobrazuje tento proces:

```
Function QuarterManager(FromMod, ToMod)
Dim FromS As Shape
Dim ToS As Shape
Set FromS = ActiveSheet.Shapes(FromMod)
Set ToS = ActiveSheet.Shapes(ToMod)
'From moduls parameters
l = FromS.Left
t = FromS.Top
h = FromS.Height
w = FromS.Width
'To module coordinates
x = ToS.Top + (ToS.Height) / 2
y = ToS.Left + (ToS.Width) / 2
fx = x + (l - t)
gx = -x + (t + h + l)
jx = x + (l + w - (t + h))
px = -x + (t + l + w)
If jx >= y And y > gx And t + h < x Then QuarterManager = 3
If fx > y And y <= gx And l > y Then QuarterManager = 2
If fx <= y And y < px And t > x Then QuarterManager = 1
If px <= y And y > jx And l + w < y Then QuarterManager = 4
End Function
```

Bod propojení pro aktivitu do které cesta směřuje, je poté určen jako opačný bodu ze kterého cesta směřuje, tedy pokud je vybraný bod 4, tak bod příchodu do další aktivity je 2.

Samotné zdroje nejsou zobrazeny v modelu. Zobrazené jsou ale cesty zdrojů, které existují mezi aktivitami. Tyto zdrojové cesty, oproti cestám mezi aktivitami, jsou vytvořeny automaticky po zadání zdroje do více jak jedné aktivity. Mají přidělenou oranžovou barvu pro lepší rozeznání. Cesty zdrojů oproti cestám aktivit mají předem nastavenou délku trvání, a to jednu vteřinu. To je minimální doba, která může být pro cestu zdroje nastavena. Pokud je potřeba cestu změnit, může se tak udělat pomocí tlačítka *Adjust*. Toto tlačítko nám umožní zobrazení všech aktivit, ke kterým je zdroj přiřazen. Zde také je možné zdroj přiřadit i k dalším aktivitám. Samotné zdroje se vytváří přes tlačítko *Add* v pásu karet.

Pokud si uživatel přeje upravit jednotlivé aktivity nebo cesty, může tak učinit rovnou v rámci modelu. Každý objekt který je v modelu zobrazen (až na cesty pro zdroje), se dá otevřít, a upravit. Zde se také zobrazí možnost pro **odstranění** aktivity z modelu. To je možné jen pomocí této metody, tedy pokud uživatel odstraní aktivity pomocí akce „vyjmout“. Aktivity se sice z modelu vyjmou, ale informace o aktivitě budou stále uloženy v systému.

Každý objekt v rámci modelu si také uchovává svoje interní identifikační číslo. S jeho pomocí následně komunikuje se zbytkem modelu. Tedy samotné jména aktivit, které se uživatel rozhodne použít, nemají vliv na procesy simulace. Z toho důvodu uživatel může pojmenovat více aktivit stejným jménem. Celkově se v modelu nachází pět skupin identifikačních čísel, které jsou od sebe rozděleny. Šestá a poslední skupina se poté váže na entity, které se generují v rámci simulace. Tyto identifikační čísla budou blíže popsána v podkapitole 4.1.

3.2 Návrh uživatelského rozhraní

Doplňek SimulTrek má dvě hlavní prostřední a jedno vedlejší. Všechny tyto prostředí jsou součástí listů v MS Excel. Mezi hlavní prostředí se řadí tvorba modelu a poté zpracování dat získané ze simulace. Model je možné vytvořit v prázdném, ale i popsaném listu. Tento list tedy slouží jako prostředí pro tvorbu modelu, kde se přidávají jednotlivé aktivity a cesty. V jednom sešitě MS Excel může existovat jen jeden takový list. Je tedy nutné, si pro tvorbu více modelů, otevřít více sešitů. Prostředí pro zpracovávání dat se rozděluje na dva listy a to Data a Summary. Tyto listy jsou vytvořené pomocí programu a uživatel si je může zobrazit po skončení simulace. Jak bylo zmíněno v úvodu kapitoly, list **Data** obsahuje veškeré informace o jednotlivých entitách, které se vytvořily v rámci simulace. Tyto informace, jsou rozdílné pro každý typ aktivity. Pro aktivity typu Generate a Departure se zaznamenává jen jediná hodnota a to čas příchodu do modelu, a odchodu z modelu. Pro aktivitu Delay se zaznamenává více informací a to:

- **Arrive** (čas příchodu) znázorňuje čas, kdy entita přišla k příslušné aktivitě.
- **Queue** (fronta) je znázorněním počtu entit ve frontě při příchodu. Například pokud je tato hodnota 4, znamená to, že v době příchodu entity existují 4 jiné, které čekají

ve frontě. Tedy pokud je tato hodnota nulová, nemusí vždy znamenat, že aktivita je prázdná. Informuje uživatele o tom, že v době příchodu entity do aktivity neexistovala žádná fronta, do které by byl možno se zařadit. Pokud je tedy aktivita prázdná, entita se přesune na zpracování. Naopak pokud je aktivita plná vytvoří entita novou frontu pro aktivitu.

- **Waiting** (čekací doba) znázorňuje čas, kdy se entita začala zpracovávat pomocí aktivity. Pomocí této hodnoty a hodnoty v **Arrive** je tedy možné zjistit, jak dlouho jednotlivé entity čekaly předtím, než byly zpracovány. Mohou nastat dvě situace a to $Arrive = Waiting$ a $Arrive < Waiting$. Pokud nastala první situace, znamená to, že entita nemusela čekat předtím než byla zpracována. Naopak druhá situace uvádí, že entita musela čekat nějakou dobu, a to přesně $Waiting - Arrive$ než mohla být zpracována. Tato čekací doba může být vyvolána například frontou nebo zavřenou aktivitou (viz podkapitola 2.2.3).
- **Done** (čas dokončení) je čas, kdy entita byla zpracována pomocí aktivity a je připravena přejít do jiné. Pomocí tohoto času a času zaznamenaném v **Waiting** můžeme zjistit, jak dlouho trvalo zpracování entity.
- **Service** (servis) nabývá hodnot "Yes" a "No". Hodnoty nám říkají, jestli entita byla zpracována nebo ne. Entita nemusí být zpracována, pokud je nastavena speciální podmínka v aktivitě (viz podkapitola 2.2.3). Tyto informace je možno využít pro lepší alokaci zdrojů a jejich počtu.

Předešlá rozdělení jsou vytvořena pro všechny aktivity, které se v modelu nachází.

List **Summary** obsahuje obecné informace o jednotlivých aktivitách. V rámci listu jsou tři tabulky informací pro každý typ aktivity.

Obecné informace pro aktivitu Generate:

- **Count** (počet) uvádí počet entit, které byly vytvořeny pomocí této aktivity.
- **Average Arrive Interval** (průměrný interval mezi příchody) představuje průměrný čas mezi příchody po sobě následujících entit. Tento čas se vypočítá pomocí vzorce:

$$x = \frac{1}{n} \sum_i^n r_i - r_{i-1}, \text{ pro } i = 1, 2, \dots, n$$

kde r_i představuje jednotlivé časy příchodu a n počet entit, které aktivita vygenerovala. Zároveň platí, že pokud $i = 1$, tak $r_{i-1} = S$, kde S , je čas zahájení simulace.

Obecné informace pro aktivitu Delay:

- **Max Queue** (maximální fronta) označuje nejvyšší počet entit ve frontě ve stejnou chvíli.
- **Average Queue** (průměrná fronta) nám říká kolik entit bylo průměrně ve frontě u aktivity. Tento průměr můžeme vypočítat pomocí vzorce:

$$x = \frac{1}{n} \sum_i^n y_i, \text{ pro } i = 1, 2, \dots, n$$

kde y_i představuje počet entit ve frontě v době příchodu a n počet entit, které přišli k této aktivitě.

- **Average Waiting Time** (průměrný čas ztrávený čekáním) znázorňuje průměrný čas, který entity strávily čekáním na zpracování. Tento čas je vyjádřený pomocí vzorce:

$$x = \frac{1}{n} \sum_i^n w_i - a_i, \text{ pro } i = 1, 2, \dots, n$$

kde w_i je čas, kdy entita začala být aktivitou zpracována (Waiting), a_i je čas, kdy entita dorazila k aktivitě (Arrive) a n počet entit, které byly zpracovány nebo momentálně se zpracovávají aktivitou.

- **Average Service Time** (průměrný čas zpracování) je délka trvání zpracování jednotlivých entit. Vyjádříme pomocí:

$$x = \frac{1}{n} \sum_i^n d_i - w_i, \text{ pro } i = 1, 2, \dots, n$$

kde w_i opět představuje, čas kdy entita začala být zpracovávána (Waiting), d_i je čas, kdy tento proces skončil (Done) a n počet entit zpracovaných aktivitou.

- **Precentage of Serviced** (entit obslouženo v procentech) vyjadřuje kolik procent entit, byla aktivita schopná zpracovat. Výpočet této hodnoty dosáhneme pomocí vzorce:

$$x = \frac{1}{n} \sum_i^n s_i \text{ pro } i = 1, 2, \dots, n, \text{ kdy } s_i = \begin{cases} 1 & \text{jestliže je hodnota Yes} \\ 0 & \text{jinak} \end{cases}$$

kde s_i je binární číslo, které reprezentuje jestli entita byla zpracována a n počet entit, které aktivitou prošly.

- **Precentage of modul usage** (využití aktivity v procentech) říká, jak moc byla aktivita využita v rámci simulace. Tím je myšleno časové vytížení, tedy pokud je hodnota například 90%, znamená to, že aktivita byla v provozu 90% celkové doby trvání simulace. Tato hodnota je vypočítaná pomocí:

$$x = 1 - \frac{1}{E - S} \sum_i^{n+1} w_i - d_{i-1} \text{ pro } i = 1, 2, \dots, n$$

kde E je nastavený čas konce simulace, S je čas zahájení simulace, n je počet zpracovaných entit, w_i je čas, kdy je entita převzata na zpracování (Waiting) a d_{i-1} je čas, kdy předešlá entita v aktivitě byla zpracována (Done).

- **Count** (počet) určuje počet entit, které dorazily do aktivity.

Obecné informace pro aktivitu Departure:

- **Count** (počet) určuje počet entit, které odešly ze systému přes danou aktivitu.
- **Average Departure Interval** (průměrný interval mezi odchody) určuje průměrný interval mezi jednotlivými odchody entit. Tato hodnota je vypočítaná pomocí vzorce:

$$x = \frac{1}{n} \sum_i^n p_i - p_{i-1}, \text{ pro } i = 2, 3, \dots, n$$

kde p_i jsou časy po sobě odcházejících entit a n je celkový počet entit, které odešly přes aktivitu.

- **Average Time in Simulation** (průměrný čas ztrávený v modelu) je průměrný čas, který entity strávily v modelu. Jedná se o čas, který začal příchodem do modelu a poté jejich odchodem. Je vypočítán pomocí vzorce:

$$x = \frac{1}{n} \sum_i^n p_i - r_i$$

kde r_i , jsou časy příchodů jednotlivých entit do modelu, p_i jsou časy odchodů entit ze systému a n je počet entit, které odešly z modelu.

Jak bylo zmíněno v úvodu této podkapitoly, doplněk SimulTrek má třetí vedlejší prostředí. Jedná se o list s názvem **Storage**, který je pro uživatele doplněk schovaný. V tomto listu se uchovávají veškeré informace o modelu, aktivitách, a pomáhá s průběhem simulace. List a jeho funkce v průběhu simulace jsou rozvedeny v podkapitole 4.1.

4. Implementace

SimulTrek je simulační program vytvořený jako doplněk pro MS Excel ve verzi 2403. Pro správné přidání doplňku SimulTrek postupujte podle následujícího návodu:

1. Doplněk SimulTrek si stáhněte do svého počítače;
2. Otevřete si prázdný soubor MS Excel;
3. V horní liště si otevřete File (Soubor);
4. Po otevření, v levém dolním rohu se nachází Options (Možnosti). Ty si rozbalíme. Pokud tam tuto možnost nemáte a místo toho tam je More (Další). Je potřeba nejdříve kliknout na tuto možnost. To otevře malý seznam, kde se budou nacházet Options (Možnosti);
5. Otevře se okno s možnostmi a zde si otevřeme Add-ins (Doplňky), které by měly být druhé odspoda;
6. V dolní části bude tlačítko Go... (Přejít...), na které klikneme;
7. Otevře se nám nové vyskakovací okno, kde můžeme spravovat naše doplňky. Může se stát, že v seznamu doplňků již bude doplněk SimulTrek, ale pokud se zde nenachází klikneme na tlačítko Browse... (Procházet..);
8. Najdeme doplněk SimulTrek v našem počítači a poté dáme OK.
9. Nakonec v seznamu doplňků zaškrtneme SimulTrek a program by měl být implementován do MS Excel.

Jakmile je doplněk SimulTrek přidán do MS Excel, v horní liště nalezneme novou záložku s názvem SimulTrek, kde se nachází veškeré funkcionality doplňku. Přidaný doplněk se stává součástí souborů MS Excel, a každý následně otevřený sešit bude obsahovat záložku SimulTrek.

Některé programy MS Excel nedovolují, kvůli zabezpečení nedovolují, používat makra, která jsou součástí externích doplňků. Pokud tato situace nastane, je potřeba jít znovu do nastavení a v záložce Trust Center (Centrum zabezpečení) povolit externí programy. Jedním způsobem je v této záložce jít do Trusted Locations (Důvěryhodná umístění) a přidat do seznamu soubor, ve kterém se stáhnutý doplněk SimulTrek nachází.

4.1 Popis klíčových částí kódu

Kód doplňku SimulTrek je rozdělený do dvou skupin. První skupina je zaměřena na tvoření aktivit, cest a zdrojů, druhá na průběh simulace. Obě tyto části vstupují do listu Storage, kde se nachází všechny informace o modelu a simulaci. V listu se nachází osm tabulek s informacemi, a každá tabulka obsahuje informace o určité části modelu. Tyto tabulky jsou rozdělené na:

1. **Aktivita** - Tabulka obsahuje veškeré informace týkající se aktivit, které jsou vytvořené

v modelu. Součástí této tabulky je i stav, ve kterém se aktivita nachází v průběhu simulace. Identifikační hodnoty používané v této tabulce se rozlišuje podle typu aktivity na **Arr_#**, **Del_#** a **Dep_#**, kde **#** je pořadové číslo pro danou aktivitu. Jelikož každý typ aktivity má v simulaci jiné funkce, pomáhá nám toto rozdělení pro určení v jaké aktivitě se nacházíme při výběru událostí.

2. **Zdroje** - Jelikož některé informace o zdrojích jsou specifické pro aktivity, ke kterým jsou přiděleny, jako je například kapacita, počet jednotek a pořadí, jsou tyto informace součástí tabulky s aktivitami. Oproti tomu obsahuje tato tabulka obecné informace o zdroji, jeho identifikační hodnotu, název, typ a řadu s pořadím. Identifikační hodnota pro tabulku je **Resource_#**, kde **#** je pořadové číslo zdroje. V řadě s pořadím se nachází čísla řádků pro jednotlivé aktivity, podle pořadí, které mají určené. Tato informace se během simulace aktualizuje. První na řadě je vždy ta aktivita, které je aktivní. Jediným rozdílem jsou zdroje typu priority, kde se tato informace neaktualizuje v průběhu simulace, ale na základě událostí, které v simulaci probíhají. Nakonec v tabulce existuje sloupec s počty, které se mění v průběhu simulace. Tyto počty slouží jako kontrola omezení. Máme-li tedy aktivitu omezenou počtem jednotek a jsou-li všechny vypotřebovány, zdroj se přesouvá do další aktivity a současná aktivita se zavře.
3. **Prioritní zdroje** - Jedná se o velmi malou tabulku, která obsahuje převážně informace o aktivitách s prioritním zdrojem. Jelikož tento typ zdroje přechází mezi aktivitami různě, musíme vést omezení (pokud existují) pro každou aktivitu najednou. Jejich identifikační hodnota je sloučením identifikační hodnoty aktivity a zdroje.
4. **Cesty zdrojů** - Zde jsou uvedeny cesty pro zdroje, které existují mezi aktivitami. Identifikační hodnota pro cesty zdrojů je **ResConn_#**, kde **#** představuje pořadové číslo cesty pro zdroj.
5. **Cesty pro entity** - Tabulka obsahuje informace o cestách mezi aktivitami pro entity. Jejich identifikační číslo je **Conn_#**, kde **#** představuje pořadové číslo cesty.
6. **Obecné informace** - Tabulka, kde se uchovávají pořadové čísla jednotlivých aktivit, cest, zdrojů a entit. Dále je zde zaznamenán čas zahájení a konce simulace.
7. **Simulační přehled** - Ten se vytváří na začátku simulace a obsahuje všechny možnosti, do kterých mohou entity dorazit. V tabulce se uchovávají časy entit, jejich stav, a kde se nachází v modelu. Pokud existují zdroje, které jsou omezeny časovou jednotkou jsou obsaženy také v této tabulce a společně s časem entit, se odtud vybírá čas události, která bude zpracována.
8. **Fronty** - Tabulka front je speciální v tom, že není omezená na počet sloupců. Sloupce v tabulce se přidávají a odebírají podle situace v simulaci. Seznam front, které existují v daném okamžiku se zaznamenává do samostatného sloupce.

4.1.1 Vytváření modelu pomocí skriptu

Jak bylo popsáno v podkapitole 3.1, pro tvorbu jednotlivých aktivit a cest se používá objekt UserForm. Objekt UserForm tedy umožňuje tvorbu vlastních vyskakovacích oken s různým typem polí. Tyto pole mohou obsahovat vlastní skripta, která jsou určena jen pro ně. Doplněk

SimulTrek využívá tuto vlastnost, pro přidávání a odebírání polí, podle uživatelem vybrané možnosti. Skripta rozdělíme na tři hlavní části: inicializace okna, úpravy a kontroly polí a uložení informací.

Inicializace okna se stará o jeho vytvoření a nastavení. Součástí tohoto skriptu je nastavení velikosti písma, pro jednotlivé pole, a přidávání položek do seznamů, které se v objektu nachází. Jelikož se většina těchto nastavení opakuje, jsou vytvořeny jako samostatné procesy, a objekty je jen volají. Procesy, které se volají, mohou být například již zmíněné nastavení velikosti písma, nebo přidání typů distribuce do seznamu. Jelikož se používají pro všechny inicializace UserForm, mohou být zkušenějším uživatelem VBA jednoduše změněny, nebo upraveny.

Úpravy a kontroly polí fungují pomocí vybudované funkce „OnChange“. Díky této funkci VBA zaznamená jakoukoliv změnu, kterou uživatel v prostředí okna udělá, a poté na to reaguje. Například uživatel vybere exponenciální distribuci, která má jeden parametr. Skript v UserForm tuto změnu zaznamená a vytvoří nové pole, kam se bude moct doplnit informace pro daný parametr. Pokud se ale uživatel rozhodne tuto distribuci změnit na normální, která má parametry dva, bude tato změna opět zaznamenána. Odstraní se staré pole a místo něj se vytvoří dvě nové. Jelikož neustále tedy přidáváme, a odebíráme pole, je potřeba i měnit pozice a velikost okna. To je dokázáno pomocí jasně definovaných pravidel pro strukturu polí v objektu. Každé okno má v našem případě pole, které obsahuje název s velkým písmenem. Od tohoto pole pak začíná další řádek polí po 40 pixelech. Pomocí této informace, se tedy přidává a odebírá 40 pixelů od pozice polí, a tím se i upravuje velikost okna.

V jednotlivých oknech také existují různé kontroly specifické pro informace, které se vyplňují. Tyto kontroly se dělají poté, co uživatel potvrdil vytvoření aktivity nebo cesty. Mezi kontroly se řadí jednoduché kontroly čísel (například že kapacita u Delay aktivitě je větší nebo rovna jedné), po složitější (například kontroly pravděpodobnosti u cest). Tyto kontroly upozorňují uživatele na špatně vyplněné informace, vedoucí k chybám v simulaci.

Ukládání informací se provádí poté co se zkontrolovaly všechny hodnoty. VBA převezme informace vyplněné uživatelem a zaznamená je do určených tabulek. Pokud se jedná o úpravy již vytvořené aktivity, nebo cesty, přepíšou se všechny informace, které máme k dispozici. Tím je zaručeno, že pokud uživatel udělal nějakou změnu, bude tato změna přenesena do tabulek.

4.1.2 Průběh simulace

Simulace v doplňku SimulTrek funguje na principu diskrétních událostí. Tyto události jsou ukládány do seznamu a VBA si z nich vybírá právě tu událost, která má minimální čas, tedy stane se nejdříve. Předtím, než se dojde k samotnému průběhu simulace, je potřeba připravit seznam událostí společně s dalšími procesy. To probíhá v **přípravné fázi**. V této fázi je nejdříve potřeba uživatelem doplnit na jak dlouho bude simulace probíhat, a také jestli si přeje zapnout možnosti **Forced Clear** (podkapitola 2.2) a **animace**. Jakmile jsou tyto informace k dispozici, probíhají kontroly modelu. Jeden z těchto procesů kontroluje pravděpodobnosti

cest z aktivit. Pokud narazí na nějaké, u kterých je součet odlišný od jedničky, přepočítá je a upraví. Po kontrolách skript začne vytvářet seznam pro události. Tento seznam obsahuje veškeré možnosti, jaké události mohou nastat. Jedná se nejen o jednotlivé aktivity a jejich kapacity, ale také o časová omezení zdrojů. Posledním krokem v přípravné části je vytvoření list Data, do kterého se vyplní sloupce pro doplňování informací v průběhu simulace.

Po skončení přípravné fáze začíná **fáze otevření simulace**. Zde se vyhledají všechny aktivity Generate, a pomocí jejich nastavení, jsou vygenerované první entity, které budou vstupovat do systému.

Následuje **fáze simulace**. Jelikož je tato fáze největší a obsahuje nejvíce kroků, bude rozdělena na čtyři procesy. Ty budou rozdělovat podle toho, v jaké aktivitě se událost bude nacházet. Nejdříve se zaměříme na způsob se vybírání události. Jak již bylo zmíněno v listu Storage, je seznam, ve kterém jsou všechny možnosti aktivit, u kterých může událost vzniknout. U těchto aktivit je uvedený čas, identifikační číslo entity, její status a speciální vlastnost (status entity a speciální vlastnost se používají jen v některých případech). Časy událostí jsou vedeny ve vteřinách, a nově generované události jsou zaokrouhleny na celé čísla. Z těchto časů skript vybírá ten **nejmenší**, a stanoví ho jako *čas simulace*. Probíhá kontrola jestli nově vybraný čas není větší, než čas konce simulace. Pokud ano, a uživatelem není zapnutá možnost Forced Clear, simulace se ukončí. Jestliže je Forced Clear zapnutý, simulace pokračuje dále, a zastaví se jen v případě, že v systému nezbudou žádné entity vykonávající akce. K takto zvolenému času bude náležet i aktivita, podle které si určíme další proces:

- **Generate** - Jedná se o entitu, které přišla nově do systému. Proces začne tím, že vybere cestu. Při vybírání cesty existují dvě možnosti:
 - Existuje cesta jen jedna, tedy entita si vybere tuto cestu
 - Existuje dvě nebo více cest. Každá tato cesta má předem nastavenou pravděpodobnost. Pro vybrání cesty v tomto případě použijeme algoritmus:
 1. $P = 0$;
 2. Generujeme náhodné číslo r ;
 3. $P = P + p_i$;
 4. Když $P < r$, pak $i = i + 1$ a jdi na krok 3;
 5. konec, i je indexové číslo vybrané cesty.

kde P je kumulativní pravděpodobnost a p_i je seznam pravděpodobností možných cest.

Poté co je vybraná cesta, generuje se čas přechodu mezi aktivitami, který se poté přidává k době příchodu. U následující aktivity je potřeba zkontrolovat, jestli se entita nebude muset zařadit do fronty. Tato kontrola se přeskakuje, pokud je u aktivity přidělena speciální vlastnost. Pokud tato situace nastane, entita projde aktivitou, a v listu Data se bude zaznamenáno, že nebyla zpracována.

Kontrola existence fronty bude provedena nejdříve rychlým způsobem. Ten se podívá na seznam front, které v daném čase existují. Jestliže je následující aktivita v seznamu, entita se rovnou přidá do příslušné fronty. Pokud ale aktivita není v seznamu front, přechází se na delší kontrolu. Ta spočívá v procházení kapacit aktivit a kontrole, jsou-li

všechny obsazené. Pokud je nějaká volná, entita přejde do této aktivity, a její stav se označí jako *Waiting*. Jestli jsou všechny kapacity pro aktivitu plné, vytvoří se fronta a aktivita se zapíše do seznamu front.

Jakmile entita přešla k jiné aktivitě probíhá kontrola času, není-li čas simulace větší než konec simulace. Pokud nevygeneruje aktivita novou entitu, která vstoupí do systému. Posledním krokem v tomto procesu je kontrola zdroje typu **Priority**, pokud existuje. Ta začne jen v případě, že entita přišla do aktivity, obsahující zdroj tohoto typu. Proces se následně podívá na všechny aktivity, součástí tohoto zdroje, a zkontroluje, že je aktivní zdroj s nejvyšší prioritou. Bude-li potřeba změnit stav, zdroj se přesune. Doba potřebná k přesunutí zdroje bude přičtena k času simulace. Tuto hodnotu přidělíme jako čas zahájení zpracování. Může nastat situace, kdy zdroj, který se má přemístit, zpracovává jinou entitu. V této situaci místo času simulace, se vybere čas skončení zpracování, a společně s délkou cesty se určí jako čas zahájení. Například budeme mít dvě aktivity, které sdílí jeden zdroj typu priority. Tyto aktivity si nazveme A1 a A2. V čase 1000 A1 je prázdná a v A2 je entita a fronta. Tedy zdroj se nachází v A2 a zpracovává entitu až do času 1020. Nová entita přišla v čase 1010 do systému a přešla k A1. Jelikož A1 má větší prioritu než A2 zdroj se přesune do A1 a přesun trval 5. Začátek zpracování entity v A1 tedy bude 1025. Důvodem je: Zdroj dokončil zpracování v A2 (1020); Zdroj přechází do A1 (+5); Zahájení zpracování v A1 (1025).

Tímto tento proces končí a začíná výběr nové události.

- **Delay** - Proces je rozdělen podle stavu entity a to na *Waiting* nebo *Done*. Pokud je stav entity *Waiting*, znamená to, že entita ještě nebyla zpracována a čeká na zpracování, naopak stav *Done* říká, že entita byla již zpracována a bude z aktivity odcházet.
 - **Stav entity: Waiting** - Aktivita začne entitu zpracovávat. V rámci zpracování se kontroluje i **omezení u zdroje** pokud existuje. Tato kontrola probíhá jen v situaci, kdy zdroj zpracovává poslední jednotku omezení, a tedy bude se muset přesunout, a nebo odejít ze systému. Pokud je přidělený zdroj typu **Fixed** nebo **Priority**, aktivita se zavře a zdroj odejde ze systému. Pro zdroje typu **Flow**, také probíhá kontrola, jestli se nejedná o poslední aktivitu v pořadí. Jestli ne, přesune se zdroj na další aktivitu v pořadí. Pokud ano, odejde zdroj ze systému. V obou případech se aktivita zavře. Po typ **Cycle**, se zdroj přesune na následující aktivitu a současná aktivita se zavře a přesune na konec pořadí. Jestli se zdroj přesunul na novou aktivitu, její stav se změní na aktivní a aktualizují se omezení. Pokud je nastaveno omezení časové, vytvoří se nová událost v tabulce simulace. Poté, co máme zpracované zdroje, začneme generovat délku obsluhy. Takto vygenerovaný čas bude přiřazen k entitě jako její nová událost, a její stav se přepíše na *Done*.
 - **Stav entity: Done** - Entita odchází z aktivity a vybírá si novou cestu. Tento proces funguje stejně jako u procesu *Generate*. Poté co entita odešla k nové aktivitě, je potřeba zkontrolovat existence fronty. Jestli fronta pro současnou aktivitu existuje, přesune se entita z fronty do aktivity a začne se zpracovávat. Pokud tato entita byla poslední ve frontě, tak fronta bude odstraněna ze seznamu. Nakonec se udělá opět kontrola zdroje typu **Priority**. Kontrola probíhá podobně

jako v procesu Generate, jen s tím rozdílem, že se nekontroluje jen následující aktivita, ale i současná. Tedy pokud v aktivitě nebyla fronta, zdroj je uvolněn a může se přesunout k aktivitám, které mají nižší prioritu.

Tímto tento proces končí a začíná výběr nové události.

- **Deputre** - Je velmi jednoduchým procesem, spočívající jen v tom, že odstraní entitu ze systému. V tomto procesu probíhá, podobně jako u procesu Delay, kontrola existence fronty.

Poté co proběhne tato kontrola se tento proces ukončí a začíná výběr nové události.

- **Resource** - Jedná se o speciální proces, který neposouvá entity, ale jen zdroje. Nastává jen ve chvíli, kdy se vypořebuje zdroj, který má nastavené časové omezení. V rámci tohoto procesu se zdroj přesouvá stejně jak je to popsáno v procesu Delay. Jakmile je zdroj přesunut, je událost odstraněna z tabulky.

Poslední nastupuje **fáze vyhodnocovací**. V této fázi se vytvoří list Summary, do kterého se začnou vkládat informace z listu Data. Tyto informace se počítají podle vzorců probraných v podkapitole 3.2. Poté co jsou doplněné všechny informace se celý proces ukončí.

4.2 Testování a ladění programu SimulTrek

Doplňek SimulTrek je otestovaný na několika příkladech, kde se testovala jeho funkcionality a rychlost. Při testování různých modelů se přišlo na malý počet chyb, které občas bránily průběhu simulace. Jednalo se jak o chybně zapsaný kód, tak o chyby v logice některých procesů. Tyto chyby byly následně opraveny, aby bylo zaručeno, že doplněk splňuje svoji funkci

Dále se testovala rychlost doplňku, která byla velmi odlišná u různých modelů. Jedním důvodem těchto rozdílů bylo zpracování front. Pokud je vytvořený model, ve kterém se budou vytvářet velké fronty o desítkách, entit velice to sníží rychlost doplňku. Doplněk je schopný zpracovat přibližně 300 entit za minutu.

5. Příklady využití

V této kapitole jsou představené možné použití doplňku SimulTrek. Doplňěk SimulTrek byl primárně vytvořen pro modely hromadné obsluhy, ale dá se využít i na jiné typy simulace.

5.1 Modely hromadné obsluhy

Jedná se o systémy, ve kterých dochází k obsluze entity. Entity do systému přicházejí, jsou obslouženy a následně systém opustí. Sítě obslužných linek můžeme rozdělit do tří základních možností (12): *jedna obslužná linka*, *paralelně uspořádané obslužné linky* a *sériově uspořádané obslužné linky*. **Klasifikace** těchto modelů je vyjádřena posloupností šesti symbolů a to:

$$\mathbf{A/B/C/D/E/F}$$

kde: **A** charakterizuje typ pravděpodobnostního rozdělení, popisující intervaly mezi příchody entit do systému. Pro exponenciální distribuce je používán symbol **M**, pro konstantní intervaly symbol **D** a pro normální distribuci **G**. Jedná se o spojitě distribuční rozdělení. V rámci modelu je možné, použít i diskrétní distribuce. Jen je potřeba jejich použití dobře vysvětlit.

B charakterizuje typ pravděpodobnostního rozdělení, které popisuje trvání zpracování. Používají se stejné symboly jako u příchodů.

C je počet paralelně uspořádaných linek.

D je číslo udávající kapacitu aktivity.

E udává počet jednotek u zdroje v aktivitě.

F je režim fronty. V případě doplňku SimulTrek se tedy jedná o *FIFO*.

Pomocí těchto informací jsou připraveny následující příklady. Pro každou možnost je uveden jeden příklad simulačního modelu v doplňku SimulTrek.

Příklad 1: Jedna obslužná linka

Model je zapsaný jako **M/M/1/1/∞/FIFO**, který představuje obchod s jednou pokladnou. Po vytvoření modelu v SimulTrek (obrázek 5.1), se v systému nachází tři aktivity a dvě cesty. Cesty v tomto případě budou brány jako okamžité.

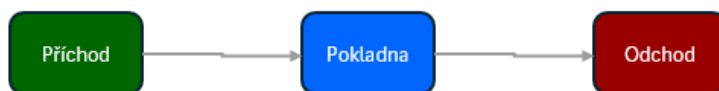
V modelu se nachází jedna aktivita Generate (Příchod), která má exponenciální distribuci ($\lambda = 2$ minuty), jedna aktivita Delay (Pokladna), která má také exponenciální distribuci ($\lambda = 2$ minuty) a jedna aktivita Departure (Odchod).

Z modelu jde tedy poznat, že **intenzita příchodu** (λ) je stejná jako **intenzita obsluhy** (μ). Tedy takto vytvořený model není možné počítat matematicky, jelikož nesplňuje podmínku $\lambda/\mu < 1$. Pomocí simulace, tedy je možné pozorovat i tyto typy modelů.

Při spuštění simulace nastavíme dobu trvání na jednu hodinu a je aktivována i podmínky *Forced Clear*.

Výsledky simulace:

Celkem modelem prošlo 129 zákazníků, kteří strávily v modelu průměrně 3 minuty a 19 vteřin. Pokladna byla využita na 93,92%, kdy průměrně u ní byla fronta o 4,93 entit. Tato fronta dosáhla maximální délky 15 zákazníků a průměrně zákazníci čekaly 2 minuty a 49 vteřin, předtím než byly zpracovány.



Obrázek 5.1: Model jedné obslužné linky v SimulTrek.

Příklad 2: Paralelně uspořádané obslužné linky

Klasifikace modelu je **M/M/3/1/∞/FIFO**. V modelu se nachází pět aktivit a šest cest (obrázek 5.2). Cesty jsou nastavené na okamžitý přenos entit.

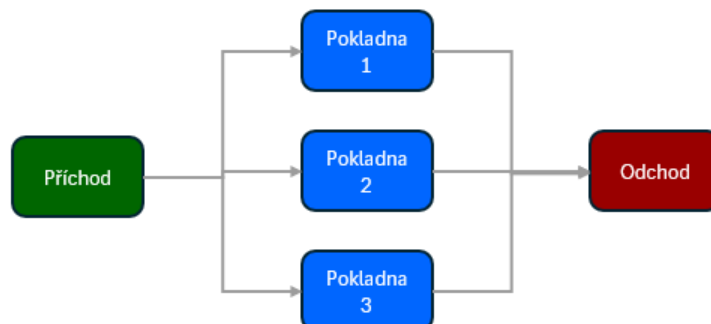
Všechny aktivity, kromě Departure (Odchod) používají exponenciální distribuci. Pro aktivitu Genereta je $\lambda = 4$, a pro Delay kativity je u všech $\lambda = 1$. Pravděpodobnost pro vybrání cesty do jednotlivých Delay aktivit je u všech 1/3. Simulace je spuštěná jednu hodinu a podmínka *Forced Clear* je aktivována.

Výsledky simulace:

V do modelu přišlo 274 entit a všechny byly obslouženy. Simulace začla v 1:00:00 a skončila v 2:44:11. To znamená, že simulace musela běžet o přibližně 44 minut déle, než odešly všechny entity ze systému. Průměrný čas, který entita strávila v modelu je 16 minut a 32 sekund. Jednotlivé poklady v modelu měly následující statistiky:

- **Pokladna 1** - průměrná fronta 21,17 entit; maximální fronta 40; průměrný čas strávený čekáním na zpracování byl 20 minut a 47 sekund; aktivita byla využita na 99,61%; bylo zpracováno 95 entit.
- **Pokladna 2** - průměrná fronta 7,39 entit; maximální fronta 13; průměrný čas strávený čekáním na zpracování byl 6 minut a 5 vteřin; aktivita byla využita na 97,69%; bylo zpracováno 80 entit.
- **Pokladna 3** - průměrná fronta 17,85 entit; maximální fronta 37; průměrný čas strávený čekáním na zpracování byl 18 minut a 16 sekund; aktivita byla využita na 97,06%; bylo zpracováno 99 entit.

Výsledky nám poukázali na zajímavý úkaz. Pokladna 2, i když byla nastavené úplně stejně jako ostatní pokladny, má o polovinu menší frontu, než ostatní. S touto informací tedy uživatel může dále experimentovat při úpravách v modelu.



Obrázek 5.2: Model paralelně uspořádané linky v SimulTrek.

Příklad 3: Sériově uspořádané obslužní linky Pro tento příklad již nebude uvedena klasifikace. Bude zde představen jedna speciální vlastnost Delay aktivit a to *No Queue*. Model se skládá ze čtyř aktivit a tří cest (obrázek 5.3) a zobrazuje nám Drive Thru u například KFC. Proces je následovný:

1. Zákazník přijede k pokladně a zaplatí;
2. Pokud je u výdejního okna volná přejede k němu, jinak čeká;
3. U výdejního okna dostane objednané jídlo;
4. Odjede ze systému.

V uvedeném příkladě tedy u výdejního okna nemůže existovat fronta (vzdálenost mezi okny je malá a auto se tam nevejde, aby uvolnilo pokladnu). V simulačním modelu je tato vlastnost označena jako *No Queue*.

Doplňek SimulTrek nenabízí možnost omezovat velikosti front. Pokud ale uživatel potřebuje mít tuto funkci, může ji nahradit kombinací existujících funkcí. Například vytvořením aktivity Delay, s rovnoměrnou distribucí, kdy minimální i maximální hodnota je 0, a speciální vlastností *No Queue*, se vytvoří další umělé místo ve frontě pro následující aktivitu. Pro uvedený příklad nám tuto situaci znázorňuje obrázek 5.4. Aktivita "Volné místo" slouží tedy jako fronta o velikosti 1 pro aktivitu "Výdej".

Vybraná distribuce pro všechny aktivity je exponenciální s parametrem $\lambda = 2$.

Výsledky simulace:

Do Drive Thru přijelo celkem 118 aut a za hodinu bylo obslouženo 93 z nich (Forced Clear není aktivní). Ty co byly obslouženy strávily v systému průměrně 7 minut a 34 sekund. Průměrná fronta u pokladny byla 3,96 aut s maximální délkou 11 aut. Aktivita pokladna byla využita na 71,17% a výdejní okno na 64,5%. Toto využití říká, kolik času trávila aktivita

zpracováváním aut. Ty auto co byly nuceny čekat u aktivity, ji tedy i blokovaly a aktivita se v ten čas nevyužívala.



Obrázek 5.3: Model sériově uspořádané linky v SimulTrek.



Obrázek 5.4: Model sériově uspořádané linky v SimulTrek s omezenou fortanou.

5.2 Modely řízení zásob

Doplňěk SimulTrek není ideálním doplňěk pro jednoduché vytváření těchto modelů. Důvodem je rozdílný přístup ke zdrojům. V modelech řízení zásob se tyto zdroje mohou spotřebovávat flexibilně nebo se doplňovat v průběhu systému. Tedy více jak jeden typ zdroje může být součástí jedné aktivity. V doplňku SimulTrek sice tyto vlastnosti pro zdroje nejsou, ale mohou být nahrazeny procesy, které k dispozici jsou.

Příkladem takového modelu je třeba stánek s limonádou. Tento stánek je otevřený je jeden den a to například v den kdy se jde pochod Praha-Prčice. Provozovatel stánku tedy bude prodávat studenou limonádu lidem, kteří se účastní tohoto pochodu. Jelikož chce, aby limonáda zůstala studená, připraví 15 litrů limonády ve stánku a zbytek uchová v ledničce. Provozovatel se domluví s kolegou, že mu dodá vždy 15 litrů limonády po hodině, aby si tím mohl doplnit zásoby. Tím tedy chce zajistit, že jeho limonáda bude vždy studená.

Simulační model tohoto systému se bude skládat ze dvou částí, každá o třech aktivitách. První část reprezentuje příchody účastníků ke stánku, obsluhu těchto zákazníků a následně jejich odchod. Druhá část se zaměřuje na zásobování. Jelikož doplňěk SimulTrek nenabízí možnost pro doplnění zásob do jiné aktivity, je potřeba tento proces nahradit něčím jiným. K náhradě využijeme zdroj typu Cycle, který bude reprezentovat dodávání zásob. Délka simulace je 8 hodin a podmínka Forced Clear je zapnutá.

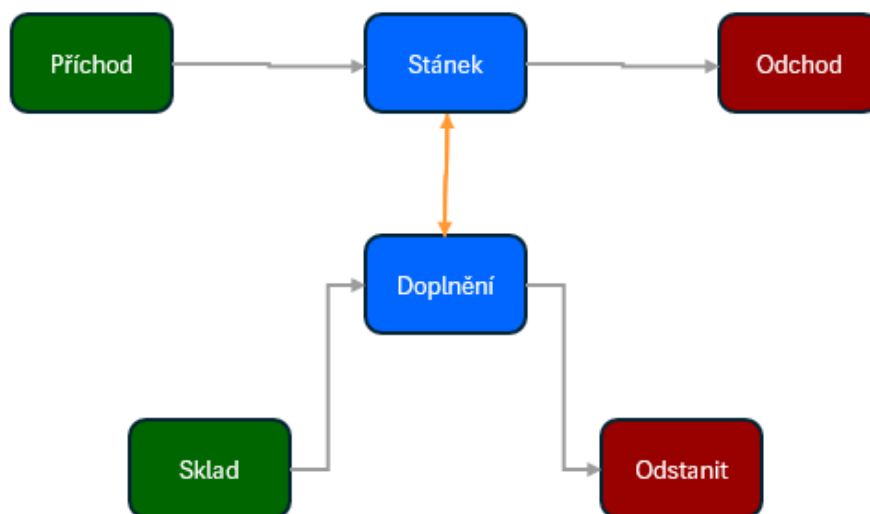
Jednotlivé aktivity mají následující parametry:

- **Příchod** - Generate aktivita s normální distribucí pro $\mu = 100$ a $\sigma^2 = 10$ sekund.
- **Stánek** - Delay aktivita exponenciální distribucí pro $\lambda = 2$ minut. V této aktivitě se nachází zdroj **Provozovatel**, který má jednu kapacitu a je omezen 30 jednotkami (v našem případě se jedná o 30 püllitrů, které může prodat). Tato aktivita má i speciální vlastnost Go Throught.

- **Odchod** - Departure aktivita.
- **Sklad** - Generate aktivita s rovnoměrnou distribucí pro $a = 2$ a $b = 2$, tedy entity se generují každé dvě minuty.
- **Doplnění** - Delay aktivita s rovnoměrnou distribucí pro $a = 0$ a $b = 0$ (entity jsou zpracovány ihned). Aktivita obsahuje i zdroj **Provozovatel** se stejnými parametry jako v aktivitě **Stánek**.
- **Odstanit** - Departure aktivita.

Propojení těchto aktivit je zobrazené na obrázku 5.5.

Tento model funguje následovně. v První části se generují zákazníci, kteří si kupují půllitru limonády. Jakmile se vyčerpá naše zásoba ve stánku (15 litrů), provozovatel se přesune do skladu, kde se postupně hromadí entity ze skladu. Tyto entity jsou generovány každé dvě minuty (podle zadání za hodinu jich musí být 30 (15×2), tedy $30/60 = 2$ minuty). Pokud nastane situace, že provozovatel prodá limonády rychle, tak je poté nucen čekat v aktivitě **Doplnění**, než dorazí zbylé. V realitě toto představuje situaci, kdy provozovatel čeká na kolegu než dorazí s další dodávkou. Naopak u situace, kdy se limonády prodávají pomalu, se entity ze skladu hromadí. V speciální vlastnost aktivy **Stánek**, říká že pokud jsme v situaci, kdy je neaktivní (limonáda není k dispozici) zákazník rovnou odejde, protože nechce čekat.



Obrázek 5.5: Model řízení zásob pro jeden stánek v doplňku SimulTrek.

Výsledky simulace:

Stánek navštívilo 289 zákazníků z nich byl provozovatel schopen obsloužit 83.99%. Maximální fronta u aktivy **Doplnění** byla 23, tedy za dobu simulace nenastala situace, kdy limonáda ze skladu dorazila do stánku předtím, než byla spotřebována předchozí dodávka. V průměru strávili zákazníci v u stánku 25 vteřin.

5.3 Simulace křižovatky

Následující příklad je vytvořený za účelem představení více komplexního příkladu. Modely křižovatek jsou velmi studovaným problémem a existují i programy, které se specializují jen na simulaci určitých typů. Doplněk SimulTrek nabízí jednoduchou variantu toho problému, kdy příjezdy aut jsou generována podle předem nastavených pravidel a jsou tyto parametry stejné v průběhu simulace. Pro simulování křižovatek se v praxi moc nepoužívá simulace diskrétních událostí, ale spíše systémová dynamika.

Pro uvedený příklad je simulovaná křižovatka ve tvaru „plus“ a simulují se příjezdy aut zespoda a ze shora do této křižovatky. Samotný model se skládá z: 2 Generate aktivit, 14 Delay aktivit a 4 Departure aktivit. Součástí modelu také čtyři zdroje. Modelem je vizualizován na obrázku 5.6.

Jelikož se jedná o takto komplexní model místo popsání jednotlivých parametrů bude popsána logika v rámci modelu.

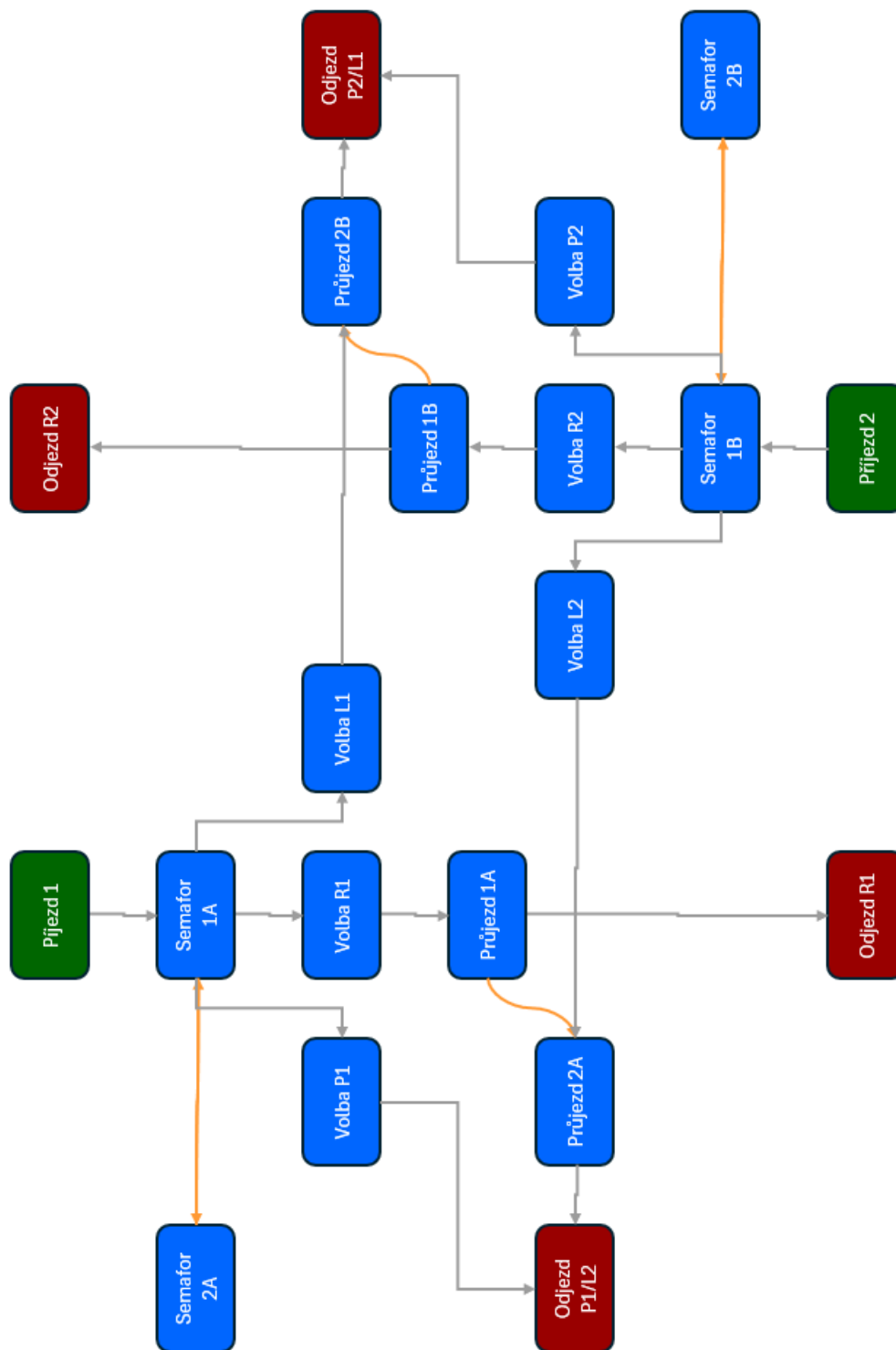
Jak již bylo zmíněno model obsahuje čtyři zdroje, z toho dva jsou identické a jedná se o zdroje, které jsou součástí aktivit „Semafor XX“. Tento zdroj má typ Cycle reprezentuje zelenou na semaforu a trvá 50 vteřin než se přesune. Samotné přesunutí poté trvá 10 vteřin. Proces reprezentuje semafor na křižovatce, které mají konstantě nastavené měnění barev. Zbývají dva zdroje jsou oba typu Priority, a reprezentují „volnost“ křižovatky. Jelikož na křižovatce se nachází auta, které budou chtít odbočovat doleva, tak budou muset dávat přednost autům jedoucím pro ně zprava. To je zaručeno právě těmito zdroji, kdy větší prioritu má entita v aktivitách s názvem Průjezd 1X, který reprezentuje auto, projíždějící křižovatkou rovně. Průjezd 2X má tedy menší prioritu a auta, odbočující doleva, projíždí křižovatkou jen když je Průjezd 1X prázdný.

Pro všechny aktivity Průjezd, je zároveň nastavena speciální vlastnost No Queue, která omezuje množství aut na křižovatce na jedno. Zamezuje to tvorbě front na křižovatce a umožňuje průjezd aut z Průjezdu 2 při změně zdroje u Semaforů.

Výsledky simulace:

Simulace běžela jednu hodinu, kdy pro obě aktivity Generate, byla nastavena exponenciální distribuce pro $\lambda = 2$. Aktivity Průjezd mají konstantní zdržení a to 5 vteřin.

Celkem do modelu přijelo 226 aut, které průměrně strávily v systému 26 vteřin. Za dobu simulace byla nejdelší fronta u aktivity Semafor 1B a to o velikosti 5. U Semaforu 1A se za celou dobu simulace nevyskytla fronta.



Obrázek 5.6: Model křižovatky v doplňku SimulTrek

Závěr

Cílem mé práce bylo vytvořit doplněk, který seznámí uživatele se základními principy simulačních programů a se simulací diskrétních událostí. Práce je psáno jako návod na používání doplňku a také popisuje jednotlivé funkce, které se v něm nachází.

Výsledkem práce je doplněk SimulTrek, jehož základní funkcí je možnost tvorby a zpracování jednoduchých simulací diskrétních událostí. Pomocí doplňku SimulTrek je možné tyto modely upravovat, a následně analyzovat výsledky získané v průběhu simulace. Po instalaci doplňku SimulTrek se v souborech MS Excel objeví nová karta v pásu karet, která slouží jako uživatelské prostředí pro doplněk SimulTrek. V této kartě uživatel nalezne veškeré procesy pro tvorbu modelů, a pro jejich následnou simulaci. Jelikož cílem práce bylo seznámit uživatele se základními principy simulačních programů, byl počet možných procesů minimalizován. Pomocí tohoto přístupu nebudou nezkušení uživatelé přehlaceni velkým množstvím funkcí, které nemusí v simulaci vůbec použít.

V bakalářské práci se v jednotlivých kapitolách kombinují teoretické části s částmi praktickými. Tento způsob opět podporuje nové uživatele k tomu, aby se nejen naučily pracovat s doplňkem SimulTrek, ale zároveň jim bylo i vysvětleno, jak jednotlivé procesy fungují a proč tomu tak je.

Zdrojový kód doplňku SimulTrek je volně přístupný a uživatelé si ho mohou zobrazit v aplikaci VBA. Obsahuje komentáře u jednotlivých procesů, a je možné ho jakkoliv upravit. Kód je napsaný tak, aby případné úpravy, bylo možné vytvořit co nejjednodušeji. Je tedy možné, dělat menší úpravy velmi jednoduše, a to bez velkých zásahů do kódu.











Doplněk SimulTrek je společně s příklady, které byly použity v bakalářské práci, dostupný na <https://github.com/D3rtex/Add-in-SimulTrek> a ve veřejných přílohách. Následný vývoj a aktualizace doplňku bude záležet na ohlasu uživatelů a případných změnách v samotném programu VBA.

Seznam použité literatury

1. BANKS, Jerry. *HANDBOOK OF SIMULATION: Principles, Methodology, Advances, Applications, and Practice*. New York: John Wiley & Sons, 1998. ISBN 0-471-13403-1.
2. BYRD, D.; MULLEN, L.; RENFRO, D.; HARRIS, T.A. Implementing a perioperative RN training program for recent graduates. *AORN Journal: The official voice of perioperative nursing*. 2015, roč. 102, č. 3. Dostupné také z: <https://doi.org/10.1016/j.aorn.2015.06.015>.
3. KOBLEN, Ivan; KOVÁCOVÁ, Jana. Selected information on flight simulators - main requirements, categories and their development, production and using for flight crew training in the both Slovak Republic and Czech Republic conditions. *Bucharest*. 2012, roč. 4, č. 3. Dostupné také z: <https://www.proquest.com/scholarly-journals/selected-information-on-flight-simulators-main/docview/1372335294/se-2>.
4. FABIAN, František; KLUIBER, Zdeněk. *Metoda Monte Carlo a možnosti jejího uplatnění*. Praha: Prospektrum, 1998. ISBN 80-7175-058-1.
5. DLOUHÝ, Martin; HLADÍK, Tomáš; JÁN, Fábry; KUNCOVÁ, Martina. *Simulace podnikových procesů*. Praha: BizBooks, 2007. ISBN 978-80-251-1649-4.
6. DRMOLA, Jakub. Systémová dynamika jako nástroj pro výzkum bezpečnosti. *Obrana a strategie*. 2014, roč. 14, č. 1. Dostupné také z: [doi:https://doi.org/10.3849/1802-7199.14.2014.01.015-028](https://doi.org/10.3849/1802-7199.14.2014.01.015-028).
7. SYCARA, Katia. Multiagent Systems. *AI Magazine*. 1998, roč. 19, č. 2. Dostupné také z: https://www.researchgate.net/publication/220605551_Multiagent_Systems.
8. *Modelování a simulace v oblasti projektového řízení*. 2001. ISBN 80-214-1734-X. Dostupné také z: <http://prog-story.technicalmuseum.cz/images/autorske/Lacko/Modelovani-a-simulace.pdf>.
9. KLEIJNEN, Jack; SANCHEZ, Susan; LUCAS, T.W.; CIOPPA, Thomas. A user's guide to the brave new world of designing simulation experiments. 2003. Dostupné také z: https://www.researchgate.net/publication/4760338_A_user's_guide_to_the_brave_new_world_of_designing_simulation_experiments.
10. WISE, M. Norton. On the narrative form of simulations. *Studies in History and Philosophy of Science Part A*. 2017, roč. 62, s. 74–85. ISSN 0039-3681. Dostupné z DOI: <https://doi.org/10.1016/j.shpsa.2017.03.010>. SI: Narrative in Science.
11. WELLMANN, Janina. Gluing life together. Computer simulation in the life sciences: an introduction. *History and Philosophy of the Life Sciences*. 2018, roč. 40. Dostupné z DOI: 10.1007/s40656-018-0235-9.
12. JABLONSKÝ, Josef. *OPERAČNÍ VÝZKUM: kvantitativní modely pro ekonomické rozhodování*. Praha: Professional publishinh, 2007. ISBN 978-80-86946-44-3.

Přílohy

A. Průvodce doplňku SimulTrek

<div> <div>Guide to MS Excel Add-In SimulTrek</div> <div> <div>This is short guide on how to use add-in SimulTrek. It discuss various modules and different types of resources in short detail.</div> </div> </div>	
<div> <div>Modules:</div> <div> <div> <div>  </div> <div> <div>Modules are objects from which the simulation models is created. Each of them has special use and properties.</div> </div> </div> </div> </div>	<div> <div>Resources:</div> <div> <div> <div> <div>Resources are used in delay modules. There are four types of resources with different attributes. Each resource can have different capacity, count and measure that can be adjusted in delay modules.</div> </div> <div> <div> <ul style="list-style-type: none"> Capacity - Must be one or higher. It says how many entities can enter module at the same time without creating queue. Count - Any positive number or "Inf". It says how many times we can use this resource in module. If module is type fixed it be will closed forever. Measure - Only in use if count is not "Inf". Then it is used to select type of "count". If time value is selected than the value in count means for how long resource stays in module. Order - It says in what order is resource choosing modules when transferring. Expect for priority. </div> </div> </div> </div> </div>
<div> <div>Generate</div> <div> <div>  </div> <div> <div>Generate module is creating entities based on set intervals. You can choose between six distributions. There can be multiple generate modules in model with different intervals. Please make sure that there is at least one connection from this module.</div> </div> </div> </div>	<div> <div>Type Fixed</div> <div> <div> <div> <div>Type Fixed can be assign to only one module. Its use are mainly to set up capacity and count if needed.</div> </div> <div> <div>Type Cycle</div> <div> <div> <div>Type Cycle is type of resource that is going through 2 or more modules in "circle". Meaning if we have two modules it just switches between them unless one of them has count set to "Inf".</div> </div> <div> <div>Type Flow</div> <div> <div> <div>Type Flow is similar to Cycle. The difference is that once it reaches last module it will stay there indefinitely. Make sure to put last module count to "Inf" to make module stay in active state.</div> </div> <div> <div>Type Priority</div> <div> <div> <div>Type Priority is the most complicated one. It chooses where to come based on order. Lower number in order is higher in priority. If someone will arrive to module with higher priority resource will go there despite queue in this current module.</div> </div> </div> </div> </div> </div> </div></div></div></div></div>
<div> <div>Delay</div> <div> <div>  </div> <div> <div>Delay module is the most complicated one. It delays entities that enters based on the selected distribution. Resources can be add to delay modules to make it active. If no resource is selected capacity of delay module is 1 with no limit. With resource every delay module has order. That mean in what order will be this module selected by resource. In every delay module you can choose special properties. There are:</div> <ul style="list-style-type: none"> Go Through - if module closed then go through it without getting service. Go Through (Empty) - only go through if closed and resource in this module is empty (does not work in type cycle resource) No Queue - This makes that there is no possible queue for this module. If one chooses do go here and module is in use it will stop in previous module and wait. </div> </div> </div>	<div> <div>Module State</div> <div> <div> <div>Module State is based on resource. If resource is assign and is in module than is closed. When closed no services is provided in module.</div> </div> </div> </div>
<div> <div>Departure</div> <div> <div>  </div> <div> <div>Departure module is used to clear entity that is leaving simulation. Make sure that no connection are from this module because it will be invalid.</div> </div> </div> </div>	<div> <div>Add resource with name and selected type. Type can then adjusted in "adjust".</div> <div> <div>  </div> <div> <div>Adjust can adjust any resource. Here you can adjust travel time of resource between modules (default 1 sec.), order of modules and type.</div> </div> </div> </div>
<div> <div>Connection</div> <div> <div>  </div> <div> <div>Connection create wat between two modules. If two or more connections are made from the same module probability needs to be selected. if the probability is not equal to 1 it will be adjusted during simulation. Connection can have travel time but if empty the transfer is instant.</div> </div> </div> </div>	<div> <div>Clear</div> <div> <div> <div>  </div> <div> <div>Clear will delete all data and objects in model. Be aware after clicking this button you can not go back.</div> </div> </div> </div> </div>
<div> <div>Start</div> <div> <div>  </div> <div> <div>Start will start simulation. First we need to select time for how long the simulation will last. Next we can select forced clear. This will stop new entities to enter model and than end simulation after every entity that is sill in simulation left (if it can).</div> </div> </div> </div>	<div> <div>Reset simulation and will clear our data from previous simulation. Model will stay.</div> <div> <div>  </div> </div> </div>
<div> <div>Help</div> <div> <div>  </div> <div> <div>Help will open Github with this guide. (currently not working)</div> </div> </div> </div>	

Obrázek A.1: Průvodce doplňku SimulTrek v angličtině

B. VBA funkce pro generování hodnot distribucí

Exponenciální distribuce:

```
Function Exp_Gen(Par1) As Double
    Exp_Gen = (-1 / Par1) * Log(1 - Rnd)
End Function
```

kde "Par1" je předem určený parametr λ .

Rovnoměrná distribuce:

```
Function Uni_Gen(Par1, Par2) As Double
    Uni_Gen = Par1 + (Par2 - Par1) * Rnd
End Function
```

kde "Par1" je parametr a a "Par2" je parametr b .

Normální distribuce:

```
Function Nor_Gen(Par1, Par2) As Double
    Again:
    V1 = 2 * Rnd - 1
    V2 = 2 * Rnd - 1
    w = (V1) ^ 2 + (V2) ^ 2
    If w >= 1 Then GoTo Again:
    y = ((-2 * Log(w)) / w) ^ (1 / 2)
    Nor_Gen = V1 * y * (Par2) ^ (1 / 2) + Par1
End Function
```

kde "Par1" je parametr μ a "Par2" je parametr σ^2 .

Geometrická distribuce:

```
Function Geo_Gen(Par1) As Double
    Geo_Gen = 0
    Again:
    If Rnd < Par1 Then GoTo Skip
    Geo_Gen = Geo_Gen + 1
    GoTo Again:
```

```
Skip:  
End Function
```

kde "Par1" je parametr p .

Binomická distribuce:

```
Function Bin_Gen(Par1, Par2) As Double  
    Bin_Gen = 0  
    For i = 1 To Par1  
        If Rnd <= Par2 Then Bin_Gen = Bin_Gen + 1  
    Next i  
End Function
```

kde "Par1" je parametr n a "Par2" je parametr p .

Posissonova distribuce:

```
Function Poi_Gen(Par1) As Double  
    l = Exp(-lambda)  
    k = 0  
    p = 1  
    Do  
        k = k + 1  
        p = p * Rnd()  
    Loop While p > l  
    Poi_Gen = k - 1  
End Function
```

kde "Par1" je parametr λ