

# Simulace systému rozvozu objednávek jídel pomocí firmy DoDo

**Modelování a simulace - SHO v logistice**

**8. 12. 2021**

Vedoucí: Filip Solich (xsolic00)

Člen: Marek Sechra (xsechr00)

# Obsah

<b>Úvod</b>	<b>2</b>
1.1 Autoři	2
1.2 Zdroje pro validaci modelu	2
<b>Rozbor tématu a použitých metod/technologií</b>	<b>3</b>
2.1 Popis použitých postupů	3
2.2 Popis původu použitých metod	3
<b>Koncepce</b>	<b>4</b>
3.1 Vyjádření konceptuálního modelu	4
3.2 Formy konceptuálního modelu	4
<b>Architektura</b>	<b>5</b>
4.1. Mapování abstraktního modelu do simulačního	5
4.2. Kompilace a spuštění simulace	5
<b>Podstata simulačních experimentů a jejich průběh</b>	<b>6</b>
<b>Shrnutí simulačních experimentů a závěr</b>	<b>9</b>

# 1. Úvod

Během covidové epidemie se rozvoz jídla začal těšit velké oblíbenosti. Pro mnoho restaurací se stal rozvoz jídla jediným zdrojem příjmů, proto jsme se rozhodli zjistit, jak optimalizovat průběh rozvozu jídla, aby při něm nedocházelo ke zbytečnému čerpání zdrojů.

Jelikož zatíženost provozu se může v čase měnit například v čase oběda či večeře. Jedním ze smyslů simulace [\[1, snímek 10\]](#) je zjistit, jak mohou špičky objednávek ovlivnit samotný rozvoz jídla. Dále je potřeba simulovat počet rozvázejících jednotek a sledovat jestli je jejich počet dostatečný pro plynulý rozvoz během celé směny.

## 1.1 Autoři

Na práci se podíleli studenti Filip Solich a Marek Sechra. Jako konzultant byl zvolen Lukáš Kolaja, který je dlouhodobý brigádník ve společnosti DoDo a pomohl s prvotním návrhem systému [\[1, snímek 7\]](#).

## 1.2 Zdroje pro validaci modelu

Na ověření validity modelu jsme oslovili vedení firmy DoDo. Komunikovali jsme elektronicky pomocí emailu s paní Veronikou Kardošovou, která nám poskytla omezené validační data [\[1, snímek 10\]](#) pro Brno s rozvozem pro restaurační zařízení KFC.

## 2. Rozbor tématu a použitých metod/technologií

Všechna dosavadní fakta jsme zprůměrovali, abychom dostali hodnoty co nejvíce shodné s realitou.

Zákazník si pomocí interní aplikace KFC objedná své jídlo. Otevírací doba je 8:00 - 22:00, ale poslední objednávka může být doručena nejpozději do 22:30. Kurýr, pokud je dostupný, čeká na zhotovení jídla a poté jej exportuje k zákazníkovi, kde čeká na uvedené adrese na vyzvednutí objednávky zákazníkem. V 1% případů nastává interní chyba, špatná adresa, kurýr se spletl apod. Poté Kurýr odváží jídlo na správnou adresu. Pokud kurýr odveze úspěšně objednávku vrací se k restauraci a čeká na další rozvážku. Rychlost objednávek se může během směny měnit, příchozí objednávky v době oběda mají vyšší četnost, než objednávky ke konci směny. Firma DoDo má na jednu prodejnu KFC pět kurýrů. Cesta přes Brno kurýrovi zabere průměrně 9-10 minut.

### 2.1 Popis použitých postupů

Pro vytvoření modelu byl použit programovací jazyk C++. Pro jednoduché modelování jsme použili knihovnu SIMLIB. Tyto technologie jsou vhodné především z důvodů rozšiřitelnosti a multiplatformní použitelnosti. Knihovnu SIMLIB [\[2\]](#) jsme zvolili také z hlediska jednoduchosti, která umožňuje rychlou implementaci. Prvně jsme model navrhli v Petriho síti [\[1, snímek 124\]](#), jak nám bylo doporučeno v kurzu Modelování a simulace, poté jsme díky zdrojům z tohoto kurzu mohli rovněž implementovat pomocí knihovny SIMLIB.

### 2.2 Popis původu použitých metod

Byly použity základní funkce a knihovny jazyka C++, přičemž jsme dodržovali standard C++ 11. Pro překlad byl použit nástroj `make` a překladač `gcc`. Knihovna SIMLIB byla získána z oficiálních webových stránek. Byla použita nejnovější verze 3.08 zveřejněná 4.10.2021. Autory knihovny jsou Petr Peringer, David Martinek, David Leska.

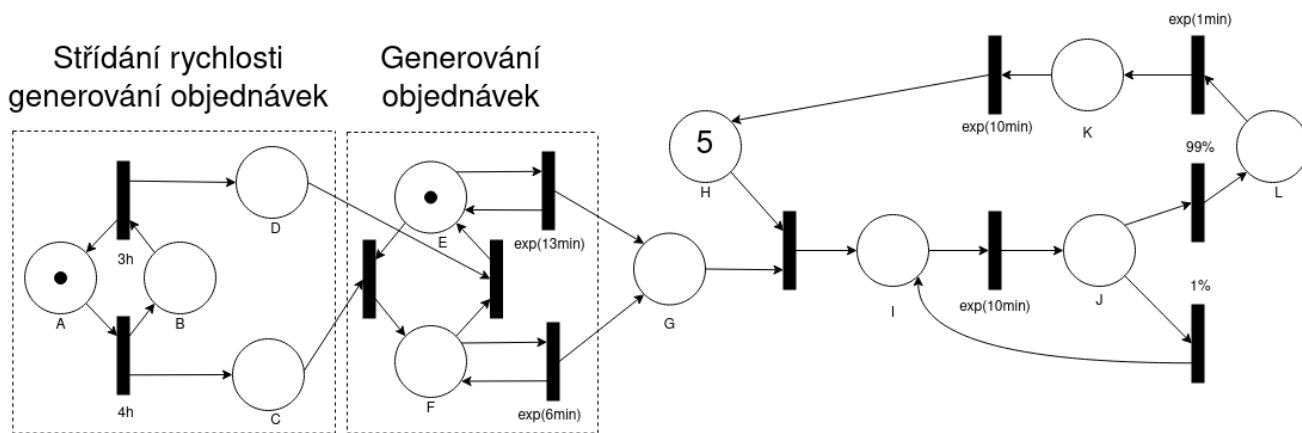
### 3. Koncepce

Náš abstraktní model [1. snímek 10] vychází z podstaty systému hromadné obsluhy [1. snímek 137]. V našem systému se může lišit průběh v provozu během víkendu či svátku nebo během pracovního dne. Dané časové údaje jsou zprůměrované, a proto je podstatné provádět simulaci v období jednoho dne. Některé vlastnosti reálného systému jsou v modelu vynechány. Otevírací doba provozoven KFC se může lišit, ale to není z hlediska pozorování důležité. Poruchy aut ve firmě DoDo, jsou tak raritní, kvůli pravidelnému servisu, že nemá smysl je do systému zahrnovat. Čas vyzvedávání objednávek v provozovně bývá v rámci sekund, na celkové zjištění by to nemělo žádný vliv.

#### 3.1 Vyjádření konceptuálního modelu

Model je popsán jednou Petriho sítí, kde první blok se věnuje přepínání rychlosti generovaných objednávek. Druhý blok se věnuje generování objednávek. Zbylá část Petriho sítě se věnuje procesu objednávky, tento proces je ukončen při doručení objednávky, pro vrácení auta do počátečního stavu je vytvořen proces, který auto navrátí.

#### 3.2 Formy konceptuálního modelu



Místa	Popis
A	Příznak znázorňující rychlejší generování objednávek
B	Příznak znázorňující pomalejší generování objednávek
C	Příznak znázorňující změnu rychlosti generování objednávek
D	Příznak znázorňující změnu rychlosti generování objednávek
E	Pomalý generátor objednávek
F	Rychlý generátor objednávek
G	Vygenerována nová objednávka.
H	Počet volných aut
I	Vyzvednutí objednávky kurýrem
J	Rozhodnutí jestli nedošlo k interní chybě
K	Kurýr se vrací k restauraci

## 4. Architektura

Program je rozdělen do jednotlivých modulů (jeden modul pro jednu třídu) a funkce `main` je v souboru `ims.cc`. V hlavičkovém souboru `config.hh` jsou definovány konstanty použité pro modelování a v souboru `stats.hh` jsou externí proměnné pro zapisování statistik.

### 4.1. Mapování abstraktního modelu do simulačního

- **Třída Shift**

Tato třída se stará o změnu rychlosti generování objednávek. Jedná se o proces a tedy dědí ze třídy `Process` z knihovny `SIMLIB`. Tento proces v modelu existuje pouze jeden a proto se jeho hlavní činnost odehrává v nekonečné smyčce. Proces během svého života vždy postupně vypne rychlý generátor objednávek tím že zabaví *zařízení* `fast_facility`, uvolní *zařízení* `slow_facility` a počká než uběhne doba kdy se mají objednávky generovat pomalu. Poté zabaví *zařízení* `slow_facility`, uvolní *zařízení* `fast_facility` a začne čekat než uběhne doba po kterou se mají objednávky generovat rychle. Tento cyklus se opakuje až do konce simulace.

- **Třída Generator**

Přes třídu `Generator` jsou implementovány generátory objednávek. Třída dědí ze třídy `Event` z knihovny `SIMLIB`, aby bylo možné tento generátor opakovaně spouštět. V modelu existují 2 instance této třídy: pomalý a rychlý generátor. Když se generátor aktivuje tak vytvoří nový proces třídy `Order`.

- **Třída Order**

Objednávka je hlavní proces který v modelu sledujeme. Při svém vzniku si *vezme* auto ze *skladu* `cars` a čeká po dobu, která simuluje doručení objednávky. Pokud objednávka selže ([popsáno v kapitole 2](#)) tak je čekání na doručení opakováno dokud doručení neselže. Poté se simuluje doba předávání objednávky. Na konec proces vygeneruje nový proces třídy `CarReturns`.

- **Třída CarReturns**

Proces `CarReturns` je zodpovědný za *vrácení* auta zpět do *skladiště* `cars` a simulování doby této *zpáteční cesty*.

### 4.2. Kompilace a spuštění simulace

Simulační model [\[1. snímek 9\]](#) se překládá pomocí příkazu `make`. Opačný příkaz pro smazání překladu a spustitelného souboru je `make clean`. Simulaci lze spustit příkazem `make run` nebo `./ims`.

## 5. Podstata simulačních experimentů a jejich průběh

### Experiment 1

Validační experiment, má za úkol potvrdit náš model jestli je validní.

Počet aut	Doba generování v normálním režimu [min]	Doba generování ve špičce [min]	Průměrná doba doručení objednávky [min]	Průměrná doba čekání na volného kurýra [min]	Počet objednávek, které čekali
5	13	6	10.97	1.22	19

Z Tabulky je zřejmé, že náš model je validní. Zjistili jsme, že firma DoDo nenechává žádnou rezervu pro vyšší četnost objednávek a proto může docházet k čekání objednávky než si ji kurýr vyzvedne. V tomto experimentu takových objednávek bylo 19.

### Experiment 2

Tento experiment má za úkol najít ideální počet aut pro situaci kdy doba pomalých objednávek je 3 hodiny a doba rychlých objednávek 4 hodiny, tak aby se průměrná doba čekání na volného kurýra podobala 0.

Doba generování pro v normálním režimu byla vždy 13 minut a pro špičku 6 minut.

Průměrná doba doručení objednávky nezávisí na tomto experimentu, proto jsme ji do experimentální tabulky neuváděli.

Počet aut	Průměrná doba čekání na volného kurýra [min]	Počet všech objednávek	Počet objednávek, které čekali
3	15.07	106	74
3	29.31	94	82
3	19.62	86	60
4	2.64	112	29
4	6.46	83	37
4	3.86	78	38
5	0.42	101	9
5	0.40	108	11
5	1.09	115	14
6	0.14	101	5
6	0.04	92	4

6	0.55	100	13
7	0.003	106	1
7	0	89	0
7	0.04	114	2

Z experimentu je patrné že při použití 7 aut a 7 kurýrů se velice dobře eliminuje počet objednávek, které museli čekat na kurýra. Zároveň nemělo smysl počet aut v experimentu zvyšovat, jelikož by počet objednávek, které čekali, osciloval kolem nuly. Pro vyhodnocení zda toto přidání aut oproti běžným 5 autům je pro pobočku výhodné je třeba zjistit ekonomické náklady na přidání auta a možné vyšší výnosy z nezpožděných objednávek. Tento experiment zároveň ukazuje druhý extrém, kdy má pobočka málo aut (některé auto se pokazilo, kurýr náhle onemocněl, ...). Při použití pouze 3 aut na směnu se průměrné zpoždění dostane až na zhruba 30 minut.

### Experiment 3

Tento experiment má za úkol najít mezery v systému, při situaci kdy nastanou výrazně vyšší špičky než u předchozích experimentů, čili dobu generovaných objednávek zrychlíme o 50%. Tedy na 3 minuty u špiček a 7 v normálním režimu. Chceme tímto simulovat dny ve kterých dochází k vysokému objednávání jídel jako jsou populární svátky.

Počet aut	Průměrná doba čekání na volného kurýra [min]	Počet všech objednávek	Počet objednávek, které čekali
5	5.46	170	125
5	20.37	158	129
5	10.63	161	111
6	5.16	173	97
6	7.95	167	104
6	8.92	176	116
7	7.68	187	103
7	1.09	178	49
7	1.75	170	56
8	0.11	167	6
8	0.59	181	31
8	0.13	175	15



9	0.09	149	2
9	0.12	159	7
9	0.10	184	9

Při sledování experimentu 3 lze vypožorovat, že pro snížení zátěže kurýrů a čekací doby objednávky na hodnoty podobné jako ty, které vycházeli z experimentu 2 je třeba v dobách větší vytíženosti restaurace přidat alespoň 4 auta a kurýry. Tímto se dostaneme na průměrně méně než 10 objednávek, které museli na kurýra čekat.

## 6. Shrnutí simulačních experimentů a závěr

V experimentu 1 jsme kontrolovali validitu našeho modelu z výsledků jsme zjistili jsme, že náš model je identický vůči reálnému systému firmy Dodo pro rozvoz jídla restaurace KFC. V experimentu 2 jsme hledali nejvhodnější počet aut, tak aby se doba vyzvednutí objednávky kurýrem blížila k nule. Zjistili jsme, že pro toto zkoumání by byl vhodnější počet aut s kurýry navýšit na 7. Vzhledem k širšímu kontextu čili zjevnému navýšení provozních nákladů toto lze firmě doporučit jen bez ohledu na finanční zdroje firmy.

V experimentu 3 jsme chtěli ukázat mezery pro situaci, kdy zátěž systému bude dvojnásobná. Pro stejný výsledek jako v běžný den s 5ti auty je potřeba 9 aut s kurýry. Studie prokázala, že navržený systém je optimální pro běžné dny, pro situaci, kdy se objem příchozích objednávek zdvojnásobí bereme jako raritní a a doporučili bychom firmě DoDo pro tyto dny počet aut s kurýr navýšit na 9.

V rámci projektu vznikl nástroj pro simulaci rozvážky jídel restaurace KFC firmou DoDo. Tento nástroj byl vytvořen v jazyce C++ pomocí knihovny SIMLIB. Program nebere žádné argumenty ze standardního vstupu a na výstupu vypisuje statistiky.

## References

- [1] Peringer, P., & Hrubý, M. (2021, 9. 20.). *Modelování a simulace*. Retrieved 12. 7., 2021, from <https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIMS-IT%2Flectures%2FIMS-2021-09-20.pdf&cid=14664>
- [2] Peringer, P., Leska, D., & Martinek, D. (2021, 10 4). *SIMLIB*. Retrieved 12 7, 2021, from <https://www.fit.vutbr.cz/~peringer/SIMLIB/>