

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

EKONOMICKÁ FAKULTA

KATEDRA OBCHODU A CESTOVNÍHO RUCHU

---

STUDIJNÍ PROGRAM: EKONOMIKA A MANAGEMENT

STUDIJNÍ OBOR: OBCHODNÍ PODNIKÁNÍ

## DIPLOMOVÁ PRÁCE

### **Simulační model řízení obchodní jednotky**

**Vedoucí diplomové práce**

**Autor diplomové práce**

Ing. Viktor Vojtko, Ph.D.

Karel Bryhcín

---

**2013**

## Prohlášení autora

Prohlašuji, že jsem diplomovou práci na téma:

*„Simulační model řízení obchodní jednotky“*

vypracoval samostatně pod odborným dohledem vedoucího diplomové práce s použitím uvedené literatury a zdrojů informací.

Prohlašuji, že v souladu s § 47 zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 12. srpna 2013

.....

## **PODĚKOVÁNÍ:**

Rád bych touto cestou poděkoval svému vedoucímu práce panu Ing. Viktorovi Vojtkovi, Ph.D. za odborné vedení, cenné rady, připomínky a konzultace při tvorbě mé diplomové práce. Dále bych chtěl poděkovat svojí rodině, zejména svojí sestře, za jejich trpělivost, podporu a poskytnuté rady.

## Obsah

1 Úvod.....	7
2 Cíl práce a metodický postup řešení .....	8
2.1 Metodický postup řešení .....	8
3 Maloobchod, maloobchodní síť .....	10
3.1 Maloobchodní jednotky .....	10
3.1.1 Typy provozních maloobchodních jednotek.....	12
3.2 Formy prodeje.....	16
3.2.1 Samoobslužný prodej.....	17
3.2.2 Pultový prodej.....	18
3.3 Obchodní provoz jako systém.....	18
3.3.1 Obchodně provozní operace .....	19
3.3.2 Space management .....	25
4 Manažerský simulátor.....	31
4.1 Mentální modely .....	31
4.1.1 Simulační modely .....	31
4.2 Multiagentní systém.....	32
4.2.1 Multiagentní simulační model .....	33
5 Výchozí simulační model - představení.....	34
6 Případové studie vycházející ze základního modelu .....	38

7 Marketingový výzkum .....	40
7.1 Metoda výzkumu .....	40
7.2 Průběh sběru dat.....	40
7.3 Sledované veličiny .....	41
7.4 Výsledky výzkumu .....	41
7 Kalibrace modelu pro případové studie .....	47
7.1 Úvod kapitoly .....	47
7.2 Layout .....	47
7.2.1 Případová studie Supermarket .....	47
7.2.2 Případové studie Diskont a Convenience store.....	49
7.2.3 Chybný výpočet zásoby zboží .....	49
7.3 Návštěvnost.....	50
7.4 Nákupní chování .....	51
7.5 Nákupní vozíky .....	51
7.6 Rychlost pohybu zákazníka, rychlost odbavení zákazníků na pokladnách .....	52
7.7 Doplnování zboží do regálů .....	54
7.7.1 Případová studie Supermarket .....	54
7.7.2 Případové studie Diskont a Convenience store.....	58
7.8 Režie prodejny .....	58
7.9 Prodejní marže .....	60
8 Metodika tvorby případových studií.....	63

9 Scénáře.....	65
9.1 Úvod kapitoly .....	65
9.2 Scénář č. 1.....	65
9.3 Scénář č. 2.....	68
9.4 Scénář č. 3.....	70
9.5 Scénář č. 4.....	72
9.6 Scénář č. 5.....	74
10 Návrhy .....	77
11 Závěr .....	80
12 Souhrn.....	83
13 Seznam použitých zdrojů.....	84

# 1 Úvod

Nákup drtivé většiny spotřebního zboží je dnes realizován v samoobslužných maloobchodních jednotkách. Prakticky každý člověk realizuje nákupy zboží denní potřeby v takovýchto prodejnách. Obchodní jednotky jsou většinou provozovány obchodními řetězci, které si vzájemně silně konkurují. Silná konkurence panuje nejen mezi jednotlivými obchodními řetězci, ale rovněž i mezi jednotlivými prodejny v určité lokalitě. Tato konkurence vytváří veliký tlak na řízení maloobchodních jednotek, neboť jen ty nejlepší mohou v konkurenčním boji uspět.

Úspěch maloobchodní jednotky je podmíněn celou řadou faktorů. Jedná se například o lokalitu, kde se jednotka nachází, spádovou oblast určující počet zákazníků, velikost realizovaných nákupů, řízení zaměstnanců, ale také prostorové řešení prodejny a mnoho dalších. Řadu těchto faktorů mohou manažeři prodejny ovlivnit svými rozhodnutími.

Řízení maloobchodní jednotky představuje pro management velice obtížný a zodpovědný úkol. Na jeho rozhodnutích závisí úspěch celé prodejny. Špatné hospodářské výsledky mohou znamenat ukončení činnosti jednotky.

V dnešní moderní době je snahou valné většiny organizací ušetřit si práci, náklady a eliminovat rizika. Jedním z nástrojů vhodných jako podpora rozhodování při řízení maloobchodních jednotek mohou být počítačové simulační modely. Obecně nejsou simulační modely v obchodě dosud příliš využívány (na rozdíl od jiných vědeckých disciplín), do budoucna mají ovšem veliký potenciál. Pokud je takovýto model vytvořen a zkalibrován pro potřeby určité případové studie, lze jej velice dobře využít jako nástroj pro podporu rozhodování managementu. V daném modelu je možné simulovat určité připravené scénáře a sledovat důsledky manažerských rozhodnutí. To může organizaci ušetřit mnoho nákladů, práce a minimalizovat rizika spojená se špatným manažerským rozhodnutím.

I v obchodě platí jednoduchý princip italského ekonoma Vilfreda Pareta, obecně známé pravidlo 80 na 20. Tedy osmdesát procent důsledků je zapříčiněno dvaceti procenty příčin. Simulační modely jsou sice zjednodušeným obrazem reality, ovšem i přesto lze s jejich pomocí identifikovat klíčové faktory podmiňující úspěch obchodních jednotek.

## 2 Cíl práce a metodický postup řešení

Cílem této práce je vytvoření třech konkrétních multiagentních simulačních modelů maloobchodních jednotek, a to na základě výchozího simulačního modelu vytvořeného v roce 2012 vedoucím této práce, panem Ing. Viktorem Vojtkem, Ph.D. Tyto modely budou vytvořeny kalibrací modelu výchozího a budou následně využitelné jako nástroj pro podporu rozhodování. Dále si práce klade za cíl vytvoření a ověření metodiky tvorby a použití případových studií s využitím simulačních modelů.

### 2.1 Metodický postup řešení

Prvním krokem vedoucím k naplnění cílů je studium odborných pramenů zabývajících se danou problematikou, především pak literatury.

Druhým krokem je navržení případových studií a definování údajů potřebných pro kalibraci modelů pro tyto případové studie.

Třetím krokem je návrh a provedení marketingového výzkumu zabývajícího se provozem maloobchodních podnikatelských jednotek. Podmínkou kalibrace modelů je provedení empirického výzkumu metodou pozorování, konkrétně nezúčastněným pozorováním skrytým.

Čtvrtým krokem je zakomponování výsledků marketingového výzkumu do počítačového modelu, neboli provedení samotné kalibrace modelů. Tato kalibrace byla prováděna prostřednictvím úpravy kódu programu simulačního modelu a prostřednictvím úpravy uživatelského rozhraní v programu NetLogo verze 5.0.3.

Pátým krokem je vypracování metodiky tvorby a použití případových studií samoobslužné maloobchodní jednotky využívající výchozí simulační model.

Šestým krokem je ověření této metodiky prostřednictvím scénářů.

Sedmým krokem je sepsání návrhů na další rozšíření a úpravu modelu pro budoucí rozvoj.



Osmým a posledním krokem je shrnutí problematiky a dosažených výsledků v závěru práce.

### 3 Maloobchod, maloobchodní síť

Pojem maloobchod definuje CIMLER a ZADRAŽILOVÁ (2007, str. 12) takto: *Maloobchod (anglicky detail trade, francouzsky commerce de detail, německy Einzelhandel) je podnik (nebo činnost) zahrnující nákup od velkoobchodu nebo od výrobce a prodej bez dalšího zpracování konečnému spotřebiteli. Maloobchod vytváří vhodné seskupení zboží – prodejní sortiment – co do druhů, množství, kvality, cenových poloh – vytváří pohotovou prodejní zásobu, poskytuje informace o zboží, zajišťuje vhodnou formu prodeje a předává marketingové informace dodavatelům.*

Podle CIMLERA (1998) je maloobchod uskutečňován prostřednictvím maloobchodní sítě, která je tvořena maloobchodními jednotkami. Maloobchodní síť představuje uspořádaný soubor maloobchodních jednotek. Tyto jednotky mají mezi sebou vazby plynoucí z konkurenčního prostředí, využívají však rovněž i principů kooperace a koordinace své činnosti.

HES (2004) uvádí, že soubor jednotek je zaměřený na konečnou realizaci zboží a je umístěn na určitém území. Hlavními funkcemi tohoto souboru je především tvorba nákupních podmínek pro uspokojování poptávky po spotřebním zboží a službách a dále zajištění ekonomické a společenské efektivnosti spojené s jejich realizací.

Jednou ze základních tendencí poslední doby v maloobchodu je růst prodejních kapacit při zmenšování počtu maloobchodních jednotek, jak uvádí PRAŽSKÁ (2002). Velké retailingové firmy disponují vlastním velkoobchodem, který je často hierarchicky členěný, vlastní autodopravou, školícími středisky, reklamní a marketingovou složkou a zejména velkými prodejními jednotkami.

#### 3.1 Maloobchodní jednotky

Funkci maloobchodních jednotek vytyčuje HES a kol. (2002, str. 101). *Maloobchodní provozní jednotky se uplatňují jako veřejně prospěšná účelová zařízení (součást občanského vybavení sídel) ve všech velikostních typech sídel (s výjimkou samot, osad a některých nejmenších vesnic do cca 200 – 300 obyvatel).*

HES a kol. (2002) dále vymezují maloobchodní jednotky pomocí několika prvků:

- lokalizace (umístění do obchodně urbanisticky vymezeného prostoru – stupňů obchodní vybavenosti),
- stavební řešení (jednouúčelové nebo víceúčelové objekty),
- technologická vybavenost,
- právní a organizační vztahy (forma vlastnictví a začlenění do organizační struktury firmy),
- provozně-technologické řešení (dispoziční řešení prodejny, formy a techniky prodeje, cesty zboží, pracovníků a zákazníků, rozložení provozní a prodejní doby, rozdělení pravomocí a odpovědnosti apod.).

Z hlediska působení v místě či oblasti CIMLER (1998) člení maloobchodní síť na základní a doplňkovou. Maloobchodní jednotky tvořící základní síť fungují po celý rok, pravidelně a mají neměnný akční rádius. Doplňková maloobchodní síť je pak tvořena jednotkami používanými při krátkodobém a místním zvýšení poptávky (např. v rekreačních oblastech, při masových akcích apod.).

Maloobchodní jednotky lze z hlediska místa působení členit na jednotky stacionární a jednotky ambulanti. Stacionární jednotky mají pevné stanoviště, jednotky ambulanti mění své místo působení.

HES a kol. (2002) definuje akční rádius prodejny. Ten je určen vzájemnou interakcí obchodní jednotky lokalizované na určitém území s dalšími obchodními jednotkami, s dodavateli, s obyvateli a s dalšími subjekty v okolí. Akční rádius prodejny není striktně vymezen. Pro různé prodejny je odlišný, stejně tak jako pro různé sortimentní skupiny zboží, pro různou obrátovou velikost obchodních jednotek nebo pro obchodní jednotky s různým počtem zaměstnanců.

### 3.1.1 Typy provozních maloobchodních jednotek

Členěním na různé typy maloobchodních provozních jednotek se ve svých publikacích zabývá řada autorů. Tato členění se u různých autorů částečně liší, ovšem většina základních typů maloobchodních jednotek se opakuje, stejně tak jsou obdobné i jejich charakteristiky.

PRAŽSKÁ (2002) zakládá členění maloobchodních jednotek je na několika kritériích. Vedle míry sortimentní specializace je to např. provozní charakter, forma prodeje aj. Samotné označení provozního typu jednotky vyjadřuje někdy míru specializace, jindy akceptuje velikost (např. supermarket, hypermarket, velkoobchod), jindy zase vyjadřuje stavebně technické řešení jednotky (obchodní dům) nebo strategii (diskontní prodejna).

PRAŽSKÁ (2002, str. 745) vymezuje hlavní druhy maloobchodních jednotek stále maloobchodní sítě v evropských poměrech takto:

- *specializované prodejny,*
- *úzce specializované prodejny,*
- *smíšené prodejny,*
- *obchodní domy (plnosortimentní),*
- *specializované obchodní domy,*
- *samoobslužné prodejny potravin (superrety),*
- *supermarkety,*
- *hypermarkety,*
- *specializované (odborné) velkoobchodní prodejny,*
- *diskontní prodejny.*

Pro účely této práce není třeba charakterizovat každý typ maloobchodních jednotek. Stěžejní pro nás bude charakteristika tří konkrétních typů maloobchodních jednotek, a to:

- supermarketu,
- diskontní prodejny,
- superrety.

#### **3.1.1.1 Supermarket**

JINDRA (1996) popisuje ve své publikaci vznik supermarketů. Označuje supermarket jako nejstarší formu diskontně orientované provozní jednotky. Tato forma vznikla v USA v období velké hospodářské krize ve třicátých letech. V tu dobu představovala revoluci v obchodě. Prodejny byly zřizovány ve starších opuštěných objektech a zaměřeny výhradně na levnější partie zboží. Rovněž byla zavedena samoobslužná forma prodeje. To umožnilo snížit ceny v té době o 30 až 40 %.

BURSTINER (1994) uvádí, že začátkem sedmdesátých let byly každé čtyři z pěti dolarů vydaných za potravinářské zboží utraceny v supermarketu (v USA). Dnešní samoobsluhy se však svým předchůdcům z času Velké deprese už nepodobají. Jsou to velké, vysokoobjemové obchody s uličkami, bohatým osvětlením a atraktivním vybavením.

LEVY a WEITZ (2009) pojmem „standardní supermarket“ dnes označují samoobslužnou prodejnu nabízející především potravinářské zboží a základní druhy nepotravinářského zboží, jako je například zboží drogistické. Zboží podléhající rychlé zkáze představuje více než 50 % prodeje supermarketu a obvykle je prodáváno s vyšší marží, než zboží s vyšší trvanlivostí.

CIMLER a ZADRAŽILOVÁ (2007) se zabývají velikostí prodejen supermarketů. Jako dolní hranice velikosti prodejen je téměř vždy uváděna velikost 400 m<sup>2</sup> prodejní plochy. Horní hranicí u tohoto typu prodejny bývá většinou 1 500 m<sup>2</sup> nebo 2 500 m<sup>2</sup> v závislosti

na tom, v jakém státu se prodejna nachází. Sortiment supermarketů obvykle zahrnuje 5 000 – 12 000 položek. Většina z nich představuje potravinářské zboží.

Umístění supermarketu je široké – od základní až po centrální vybavenost (i jako součást obchodních domů), regionální nákupní centra, dopravní uzly apod., jak uvádějí CIMLER (1998) a PRAŽSKÁ (2002).

#### **3.1.1.2 Diskontní prodejna**

PRAŽSKÁ (1997, str. 757) popisuje vznik diskontních prodejen. *Klasické diskontní prodejny (hard-discount) jsou produktem obchodu v USA třicátých let. Vysoké obchodní rozpětí neúnosně zvyšovalo ceny zboží a tak přímé vztahy obchodníků na výrobce a snižování provozních nákladů umožňovaly prodávat v této době za ceny o několik desítek procent nižší. Po 2. světové válce se staly diskontní prodejny součástí i evropské maloobchodní sítě. Míra (intenzita) diskontu může být však i v této skupině prodejen v praxi různá.*

Popisem tohoto typu obchodní jednotky se zabývá CIMLER A ZADRAŽILOVÁ (2007). Je pro ni příznačný omezený sortiment a nižší úroveň obslužného standardu. Počet nabízených druhů zboží je obvykle u potravinářských diskontů 1500, sortiment nejde příliš do hloubky. Omezena bývá především nabídka čerstvého zboží vyjma ovoce a zeleniny. Omezení sortimentu, vysoká obrátkovost a nižší obslužný standard umožňuje jednotce redukci provozních nákladů a dosažení příznivých cen pro spotřebitele. Diskontní prodejny jsou jedním z nejrychleji se rozvíjejících formátů v Evropě. Rozšiřují nabídku nepotravinářského zboží a tím zvyšují konkurenční tlak vůči ostatním velkoplošným formátům.

#### **3.1.1.3 Superreta**

JINDRA (1996) popisuje vznik této formy maloobchodních jednotek a charakterizuje ji několika znaky. Superreta vznikla později než supermarket a i v podmínkách menší velikosti se snažila uplatnit jeho výhody. Samotný název vznikl ve Francii a provozní

typ je příznačný zejména pro Evropu. V USA se pro obdobné prodejny používá většinou název Convenience Store.

Sortiment je tvořen potravinářským sortimentem, nemusí v něm být však zahrnuto maso a uzeniny. Prodej nepotravinářského zboží není vyžadován, ovšem většinou je i ten v omezeném rozsahu v sortimentu zahrnut.

Prodejní formou je samoobsluha. Obsluhované úseky se zřizují podle místních zvyklostí a potřeby, prodejní služby jsou ale v maximální možné míře omezeny.

Minimální prodejní plocha původně byla 200 m<sup>2</sup>. V jednotlivých zemích se však pohybuje obvykle v rozmezí od 160 do 250 m<sup>2</sup>. Horní hranicí je prodejní plocha o rozloze 400 m<sup>2</sup>.

Další charakteristiku této formy obchodních jednotek uvádí LEVY a WEITZ (2009). Podle těchto autorů superreta umožňují zákazníkům provést jejich nákup rychle, aniž by museli hledat zboží na rozsáhlé prodejní ploše a čekat dlouho ve frontách u pokladen. Obvykle jsou zásobovány denně kvůli nižším skladovacím kapacitám a ceny zboží v těchto obchodech jsou vyšší než v supermarketech.

*CIMLER a ZADRAŽILOVÁ (2007, str 149) se zabývají lokalizací superret. Původní prostor se těmito jednotkám otevíral především v řidší zástavbě, v základní vybavenosti měst a obcí nebo při přestavbě městských center. Dnes se uplatňují prodejny typu convenience na nádražích, letištích, podchodech, odpočívadlech dálnic, jako součást velkých čerpacích stanic apod.*

Význam těchto jednotek bude přetrvávat především v zemích, ve kterých dosud není rozvinuta síť supermarketů. Hlavním znakem těchto jednotek je blízkost spotřebiteli, ať už v bytové zástavbě či na cestách, uvádí CIMLER (1998).

## 3.2 Formy prodeje

Jedním ze základních kroků při tvorbě strategie obchodní firmy v maloobchodu je volba formy prodeje, uvádí CIMLER (1998). S vývojem maloobchodní činnosti se prodej rozdělil do čtyř základních fází:

### Nabídka zboží

V této fázi je zákazník informován o sortimentu obchodní jednotky. Obchodní jednotka zjišťuje přání a potřeby zákazníka a nabízí mu zboží obvykle formou vystavení zboží, prodejním rozhovorem, pomocí katalogu a fotografií.

### Výběr zboží

V této fázi si zákazník zboží vybírá, seznamuje se s jeho vlastnostmi a způsobem použití. Posléze se rozhoduje o koupi, a to buďto sám, nebo za přispění prodavače.

### Placení zboží

Zde se aplikují různé varianty řešení. Platba je provedena např. inkasem prodavačem, pokladní, platbou na dobírku, převodem z účtu, platební kartou, vhozením mince apod.

### Výdej zboží

Ukončuje prodej zboží tím, že zboží je předáno zákazníkovi. To probíhá nejčastěji přímo v prodejní místnosti, nebo na jiných místech (sklad, byt zákazník aj.).

CIMLER (1998, str. 79 – 80)) dále uvádí: *Různé organizační a technické řešení jednotlivých fází prodeje a jejich odlišná posloupnost i průběh vytvořily řadu variant (forem) prodeje. Ty jsou vymezovány nejčastěji s ohledem na:*

- *účast zákazníka na jednotlivých fázích (nejčastější hledisko),*
- *přístup zákazníka k zásobě zboží,*
- *způsob předání zboží zákazníkovi (dovoz do bytu, vlastní odběr zboží).*



*Podle účasti zákazníka na jednotlivých fázích prodeje se setkáváme s těmito hlavními skupinami forem prodeje:*

- *prodej, při kterém se zákazník obsluhuje převážně sám, tj. samoobslužný prodej,*
- *prodej s převážnou obsluhou prodáváčem, tj. pultový prodej, prodej volným výběrem, prodej podle vzorků, prodej na objednávku.*

*Kromě zmíněných forem prodeje v prodejních se rozeznává i prodej mimo prodejny (zásilkový, přímý, pochůzkový) se zcela odlišnou technologií provozu i odlišnou účastí zákazníka či personálu firmy na jednotlivých fázích prodeje.*

### **3.2.1 Samoobslužný prodej**

Podle PEŠTOVÉ A BIŇOVCE (1995) je typickým znakem této formy prodeje aktivní účast zákazníka na všech fázích prodeje. Zákazník si zboží sám vyhledává na prodejní ploše, prohlíží, vybírá, dopravuje k pokladně. Pro realizaci této formy prodeje je nezbytné, aby zboží bylo připraveno, byla ho na prodejní ploše dostatečná zásoba, zákazník měl volný přístup ke zboží nebo alespoň přehled o prodávaných druzích zboží a aby si jej mohl libovolně dlouho prohlížet.

Hlavním znakem této formy prodeje je pak optická nabídka zboží, které je určitým logickým způsobem uspořádáno a vystaveno.

Rozeznáváme dva druhy samoobslužných prodejen, samoobsluhu uzavřenou a samoobsluhu otevřenou.

#### Samoobsluha uzavřená

V této samoobsluze zákazník nemůže opustit prostor, ve kterém je zboží vystaveno, dokud neprojde pokladní přepážkou a zboží nezaplatí.

#### Samoobsluha otevřená

V této samoobsluze tvoří zařízení, na kterém je zboží vystaveno, samostatné ucelené ostrůvky s kontrolní pokladnou.

### 3.2.2 Pultový prodej

PRAŽSKÁ (2002) označuje pultový prodej za nejstarší formou prodeje v maloobchodě. Nabídka je zde uskutečňována formou prodejního rozhovoru, což zvyšuje náročnost na personál. Zákazník je fyzicky oddělen od zásoby zboží, což vytváří potřebu na zjištění jeho přání.

Pultový prodej je charakteristický zejména oddělením zákazníka od zboží. Toto ovlivní celou řadu provozních požadavků na řešení této formy prodeje. Poměr plochy se doporučuje více než polovina pro pohyb zákazníka, necelá polovina pro pracoviště prodavače včetně zařízení.

### 3.3 Obchodní provoz jako systém

CIMLER a ZADRAŽILOVÁ (2007, str. 200) uvádějí k logistickému pohybu zboží a systémovému přístupu v obchodě toto: *Nedílnou součástí logistického pohybu zboží od výroby ke spotřebiteli je fyzický pohyb zboží provozními jednotkami velkoobchodu (sklady) a maloobchodu (prodejny). Uplatnění logistických přístupů při zajištění fyzického pohybu zboží provozními jednotkami obchodu vyžaduje systémový přístup k chápání i řízení obchodního provozu.*

PRAŽSKÁ (2002) definuje podoby, které zboží v obchodním provozu nabývá.

- zboží jako obchodní sortiment,
- zboží jako zásoba,
- zboží jako technologická skupina.

V rámci logistického distribučního řetězce se však zboží nevyskytuje jinak, než jako přepravní jednotka.

Pohyb zboží realizují pracovníci s využitím mechanizačních prostředků a zařízení. Obchodní provoz jako systém je tvořen prvky se vzájemnými vazbami. Určující pro charakter obchodního provozu je předmět obchodní činnosti, tedy výše zmíněné

zboží. Jeho pohyb se uskutečňuje v rámci provozních jednotek obchodu a mezi provozními jednotkami obchodu. Ty jsou tvořeny stavebně technickým řešením, personálem a plochami, včetně jejich dispozičního řešení. Komplexní pojetí obchodně provozní jednotky předpokládá chápání provozu a jeho technologie jako určující součásti obchodně provozní jednotky.

#### 3.3.1 Obchodně provozní operace

CIMLER a ZADRAŽILOVÁ (2007, str. 205) definují obchodně provozní operace takto: *Obchodně provozní operace představují skladovací, prodejní, manipulační a dopravní činnosti, jimiž obchodně provozní jednotky plní svou funkci při zajišťování fyzického pohybu zboží ke spotřebiteli.*

Obchodně provozní operace můžeme rozdělit podle jednotlivých fází pohybu zboží na velkoobchodní a maloobchodní.

##### Velkoobchodní provozní operace

###### a) Příjem zboží

Tato obchodní operace začíná odběrem zboží (tzv. hrubým příjmem). Ten spočívá v prověření vnější neporušenosti uzavřených přepravních prostředků a obalových jednotek, dále v prověření jejich počtu případně počtu kusů zboží. Případně může dojít k ověření hrubé váhy dodávky. To vše se kontroluje podle průvodních dokladů.

Další fází příjmu zboží je čistá přejímka (kvantitativní a kvalitativní), včetně sortimentní a následuje případná reklamace, tvorba skladových manipulačních jednotek a vložení příslušných dat do informačního systému.

###### b) Skladování zboží

Zboží se skladuje v regálech, stohováním, volně ložené apod.

#### c) Expedice zboží (výdej zboží)

Spočívá ve sběru zboží z uložených zásob v množství a struktuře druhů podle objednávky odběratele. Probíhá kompletace spočívající v sestavování, kontrole a případném balení zboží do manipulačních jednotek podle objednávky odběratele a předání k rozvozu. Dílčími fázemi rozvozu je nakládka, vlastní přeprava a vykládka zboží u odběratele.

#### Model maloobchodních provozních operací

##### a) Příjem zboží

Primárně znamená odběr zboží. Jde tedy o fyzickou kontrolu dodané zásilky podle dodacího listu, dále podle charakteru zboží a dodávek, prověření jakosti daného zboží a zadržení zboží nejakostního pro reklamaci.

##### b) Skladování zboží

Tato operace začíná přípravou na skladování, která spočívá ve vybalení zboží, tvorbě manipulačních jednotek vhodných pro maloobchodní jednotku. Dále probíhá uložení zboží v hlavním nebo příručním skladu či přímo uložení zboží na prodejní ploše.

##### c) Příprava zboží k prodeji

Tato operace spočívá především v úpravě zboží. Ta zahrnuje především porcování, vážení, vybalování, označení cenou a v doplňování zboží do výstavního zařízení na prodejní ploše.

##### d) Prodej zboží

Tato operace zahrnuje nabídku zboží, výběr zákazníkem, placení a výdej zboží zákazníkovi, a to v různých variantách organizačního a technického řešení i účasti zákazníka (závislé na formě prodeje).

Toto členění obchodně provozních operací nalezneme v publikacích CIMLER a ZADRAŽILOVÁ (2007), PRAŽSKÁ (2002) a CIMLER (1998).

### 3.3.1.1 Cyklické nástupy zaměstnanců

MAINZOVÁ (2001, str. 41) ve své publikaci definuje pojem „prodej“. *Prodej je realizace zboží, je aktem směny, ke které dochází na trhu, kdy prodávající se zavazuje převést na kupujícího vlastnické právo k jeho užívání a kupující se zavazuje zaplatit kupní cenu.*

HESKOVÁ a PÍCHA (2004) uvádí, že každý obchod má stanovenou určitou prodejní dobu. Během této doby dochází ke směně. Vzhledem k technickému a personálnímu zajištění by v optimálním případě pro podnik zákazníci navštěvovali prodejnu rovnoměrně, tedy neexistovaly by výkyvy v návštěvnosti. K těmto výkyvům ovšem v praxi dochází, a to jak v rámci jednoho pracovního dne (v jednotlivých hodinách prodejní doby), tak i v rámci delšího období (týdny, měsíce).

Rozdíly ve frekvenci nakupujících v maloobchodní jednotce bývají tak veliké, že je většinou nutné přizpůsobovat personální vybavenost jednotky změnám oné frekvence. Přizpůsobit potřebám se musí doby nástupů zaměstnanců, doby přestávek a pracovní doby jednotlivých zaměstnanců. Převážně se jedná o prodavače a pokladní, ovšem toto se týká i dalších zaměstnanců.

HESKOVÁ a PÍCHA (2004, str. 54) nerovnoměrný rozvrh pracovní doby přizpůsobený frekvenci nazývají cyklické nástupy zaměstnanců. Určují dva způsoby jejich stanovení.

- a) *podle celkové průměrné nutné doby obsluhy jednoho zákazníka a průměrného počtu zákazníků v hodině,*
- b) *skutečný stav pracovníků prodejny se přizpůsobuje tzv. indexům frekvence kupujících v hodinách prodeje.*

HESKOVÁ A PÍCHA (2004) dále tyto dva způsoby stanovení cyklických nástupů zaměstnanců rozebírají.

#### Ad a)

První způsob stanovení cyklických nástupů zaměstnanců vychází z potřeb zákazníků, tedy z jejich nároků na obsluhu. Z nich je odvozeno, jak by měla být maloobchodní jednotka personálně vybavena v jednotlivých hodinách prodejní doby.

Průměrnou dobu obsluhy lze vypočítat na základě měření v reprezentativním období na dostatečném počtu obsluh. Průměrnou dobu obsluhy pak představuje aritmetický průměr ze získaných dat. Pokud získanou průměrnou dobu obsluhy vynásobíme postupně průměrným počtem zákazníků v jednotlivých hodinách prodeje, získáme výsledek potřebný pro nastavení cyklických nástupů zaměstnanců. Výsledek tohoto násobení představuje fond pracovní doby v minutách potřebný k obsluze. Pokud tento fond vydělíme 60, vyjde potřebný počet hodin práce (potřebný počet pracovníků). Tento čistý čas obsluhy je ovšem třeba ještě navýšit, neboť nezahrnuje čas čekání, čas na přestávky, či čas na další činnosti, které mohou tito zaměstnanci vykonávat (například doplňování zboží do regálů v případě pokladních).

Tabulka 1: Cyklické nástupy zaměstnanců – metoda a)

Prodejní doba	Průměrná doba obsluhy v min.	Průměrný počet zákazníků	Potřebný fond pracovní doby v min.	Potřeba pokladních
8 - 9	3	70	210	3,5
9 - 10	3	56	168	2,8
10 - 11	3	50	150	2,5
11 - 12	3	65	195	3,3
12 - 13	3	80	240	4,0
13 - 14	3	68	204	3,4
14 - 15	3	82	246	4,1
15 - 16	3	91	273	4,6
16 - 17	3	96	288	4,8
17 - 18	3	88	264	4,4
18 - 19	3	67	201	3,4

Zdroj: Hesková, Pícha, 2004

#### Ad b)

U druhého způsobu stanovení cyklického nástupu zaměstnanců je počet prodavačů nebo pokladních přizpůsoben indexům frekvence.

Podle HESKOVÉ A PÍCHY (2004, str. 55) toto přizpůsobení probíhá následujícím způsobem. Ze zjištěného frekvenčního testu se vypočtou indexy frekvence (hodnoty průměrné frekvence v hodinových intervalech se dělí aritmetickým průměrem této řady). Indexy frekvence ukazují kladné a záporné relativní odchylky od průměrné denní frekvence.

Dále se určí průměrný fond pracovní doby všech prodavačů nebo pokladních připadající na jednu hodinu pracovní doby. Ten závisí na celkové době prodeje. Prodejna je např. otevřena od 8. do 18. hodiny, celkem 10 hodin. Přítomnost

zaměstnanců v prodejně je od 7.30 do 18.30 hodin, tj. 11 hodin. Při 8 zaměstnancích s osmihodinovou pracovní dobou je průměrný fond pracovní doby na 1 hodinu prodejní doby:  $64 \text{ hodin} : 11 = 5,82$ .

Průměrný fond pracovní doby připadající na 1 hodinu prodeje pak vynásobíme indexy frekvence

Tabulka 2: Cyklické nástupy zaměstnanců – metoda b)

Prodejní doba	Index frekvence	Rovnoměrný počet prodavačů	Potřebný počet prodavačů
8 - 9	0,71	5,82	4,13
9 - 10	1,14	5,82	6,63
10 - 11	0,92	5,82	5,35

Zdroj: Hesková, Pícha, 2004

Výpočet indexu frekvence:

Výchozí data poskytuje frekvenční test, který udává, kolik bylo v jednotlivých hodinách prodeje nakupujících (průměr tří týdnů). Průměrnou frekvenci posléze vypočteme jako podíl frekvence celkové a počtu hodin prodeje. Indexy frekvence pro jednotlivé prodejní hodiny nakonec získáme vydělením počtu nakupujících v dané prodejní hodině a průměrné frekvence.

HESKOVÁ a PÍCHA (2004) dále zhodnocují obě popsané metody.

Druhá metoda (b) je jednodušší na výpočet, neboť je nutné pouze získat frekvenční test a z něho vypočítat index. Používá se v praxi zejména v provozech, kde je obtížné měnit daný stav zaměstnanců. Tato metoda fixuje stav zaměstnanců bez ohledu na potřebnou dobu obsluhy zákazníka.



První metoda (a) je pracnější a je u ní nutné správně odhadnout potřebné doby na ostatní činnosti (přestávky, čekání atd.). Z teoretického hlediska je správnější, neboť respektuje potřeby zákazníka na dobu obsluhy.

Po provedení výpočtu cyklických nástupů zaměstnanců je třeba výsledky přizpůsobit podmínkám.

- 1) Při daném počtu zaměstnanců lze dělit pracovní dobu jen do určitého stupně. V případě, že vypočtená hodnota předpokládá potřebu dvou a půl zaměstnance, je samozřejmě nutné číslo zaokrouhlit na 2 nebo 3 zaměstnance. Tento problém lze řešit tím, že přebytečná pracovní doba bude využita k jiným činnostem (např. doplňování zboží u pokladních).
- 2) Je třeba dbát na zákonná ustanovení týkající se pracovní doby. Nelze ji nastavit libovolným způsobem.

#### **3.3.2 Space management**

NOVÁČEK a kol. (1997) popisuje členění prostorů maloobchodní provozní jednotky. Toto členění musí být uspořádáno tak, aby se navzájem nekřížil pohyb zboží, pracovníků a zákazníků.

Prostory prodejny dělí na 2 základní skupiny:

- prostory hlavní,
- prostory pomocné.

Prostory hlavní pak jsou tvořeny prodejní plochou a prostory k poskytování služeb. Pomocné prostory představují místnostmi s přímým vztahem ke zboží (sklady), komunikačními prostory (cesty) a prostory pro pracovníky (prostory pro oddech, šatny, WC apod.).

Podle HESE a kol. (2002) prodejní plocha zahrnuje prostory obsazené prodejním a výstavním zařízením určeným pro prezentaci zboží a uložení zásob, prostory pro pohyb

zboží především při doplňování do prodejního a výstavního zařízení a prostory pro pohyb zákazníků a zaměstnanců.

HESKOVÁ a PÍCHA (2004) uvádějí, že v obchodě platí zásada, že prodejní plocha prodává, při zachování dostatečného prostoru pro pohyb zákazníků, manipulaci se zbožím, skladování zboží a nákupní prostory s pokladnami. Optimální velikost prodejní plochy by měla činit 70 – 80 % celkové plochy obchodu.

#### Dispoziční řešení obchodní jednotky

CIMLER a ZADRAŽILOVÁ (2007) definují dispoziční řešení obchodní jednotky jako prostorové uspořádání jednotlivých ploch. Velikost prodejní plochy musí odpovídat předpokládaným výkonům.

Všeobecně může dispozice prodejní místnosti vytvářet převážně:

- nucený pohyb zákazníka,
- volný pohyb zákazníka,
- nejčastěji kombinace předchozích dvou možností.

V zásadě rozlišujeme tyto modely dispozičního řešení (tzv. layouty):

#### Grid layout

Jedná se o pravidelné uspořádání charakteristické pro samoobsluhu. Výstavní zařízení je uspořádáno především rovnoběžně s bočními stěnami prodejní místnosti. Uspořádání vede zákazníka určitým směrem (řízený pohyb), s tímto směrem lze tedy počítat při řešení prezentace zboží. Právě v tom můžeme z podnikatelského hlediska spatřovat hlavní výhodu tohoto uspořádání. U zákazníka může toto řešení ovšem vzbuzovat pocit omezenosti v rozhodování.

LEVY a WEITZ (2009) uvádějí další výhodou tohoto layoutu, kterou je nákladová efektivnost. S tímto layoutem vzniká méně nevyužitého prostoru než u jiných prostorových řešení prodejní plochy. Uličky jsou široké právě natolik, aby pouze umožnily plynulý pohyb zákazníků po prodejní ploše.

Nákres grid layoutu je zobrazen na obrázku v příloze č. 1.

#### Free-flow layout

CIMLER a ZADRAŽILOVÁ (2007) označují tento layout jako model volného pohybu zákazníků. Zákazníky je přijímán pozitivně na základě možnosti přehlédnout celou nabídku prodejní místnosti, orientují se a pohybují dle vlastní vůle. Zákazníci tedy nejsou vedeni uspořádáním zařízení prodejní místnosti. Nevýhodou je, že využití prostoru je v tomto případě nižší než u grid layoutu. Tento model se nejčastěji používá u nepotravinářského, zejména pak oděvního zboží.

Nákres free-flow layoutu je zobrazen na obrázku v příloze č. 2.

#### Standard layout

Tento model představuje klasické uspořádání pultových prodejen, kdy uspořádání pultů bývá při stěnách místnosti a střed prodejny slouží k pohybu zákazníků.

#### Butique layout

Toto je uspořádání s polouzavřenými prodejními úseky. Jednotlivé sortimentní skupiny jsou uspořádány do samostatných, více či méně uzavřených prodejních úseků, které se liší jak sortimentním zaměřením, tak často i odlišnou cílovou skupinou, barevným provedením, designem, formou prodeje apod. Celá prodejní místnost takto uspořádaná pak působí specifickou nákupní atmosférou. Nejčastěji se toto řešení využívá u luxusních sortimentů univerzálních i specializovaných obchodních domů a specializovaných obchodních jednotek.

Nákres butique layoutu je zobrazen na obrázku v příloze č. 3.

### **3.3.2.1 Presentace zboží**

*Jako sortiment bývá označován cílevědomě soustředěný a utříděný soubor výrobků, ale i výkonů a služeb. Utřídění může být provedeno podle různých hledisek – podle*

výrobního materiálu, podle účelu použití, frekvence spotřeby (užití), apod., uvádí NOVÁČEK a kol. (1997, str. 78).

Z hlediska fáze reprodukce NOVÁČEK a kol. (1997) rozlišuje:

- výrobní sortiment,
- obchodní sortiment.

Obchodní sortiment dnes tvoří v podstatě cokoli, co je vyrobeno, a vstupuje to do sféry oběhu. Existují dvě, přibližně stejně velké skupiny zboží:

- spotřební zboží,
- výrobní prostředky (suroviny, materiál, stroje a zařízení).

A právě ta část obchodního sortimentu, označovaná spotřební zboží, je předmětem činnosti maloobchodu.

Primárním cílem prezentace zboží CIMLER a ZADRAŽILOVÁ (2007) označuje co nejúčinnější představení zboží zákazníkovi. Zákazníková pozornost musí být připoutána k výrobku a musí být podněcován jeho zájem o nákup výrobku.

CIMLER (1998, str. 104) označuje cíle prezentace zboží v maloobchodě pomocí 5 E:

- *Exciting (prezentace by měla podněcovat zákazníkuv zájem),*
- *Experimental (prezentace by měla působit neokoukaným, novým dojmem),*
- *Evocative (evokovat, vyvolávat asociace a tím i zájem o zboží),*
- *Enhancing (její působení by mělo zdůrazňovat vlastnosti zboží),*
- *Easy to shop (usnadňovat nákupní rozhodování a nákup samotný).*

CIMLER a ZADRAŽILOVÁ (2007) uvádějí, že podoba konkrétní prezentace zboží závisí na několika faktorech. Jedním z nich je zboží, respektive jeho vlastnosti. Těmi se myslí zejména jeho velikost, váha, obal a ostatní vlastnosti. Dalším faktorem je

velikost prostoru pro prezentaci. Podíl plochy určený nějakému druhu či skupině zboží závisí často na podílu na obratu či výnosech nebo přímé výrobové rentabilitě.

Nejdůležitějšími dvěma faktory účinnosti prezentace je umístění výrobku na prodejní ploše a umístění výrobku ve vztahu k zornému poli očí. V prodejně existují místa, která jsou zákazníkem vnímána intenzivněji než jiná. V tom je nutné vidět hodnotu takového místa.

CIMLER (1998, str. 106 – 107) rozděluje s ohledem na schopnost zákazníka na místě vnímat prezentovanou nabídku regál na tři zóny:

- *zorného pole očí (110 až 160 cm) jako nejatraktivnější zóna,*
- *zóna dolní (odběr zboží v předklonu) a*
- *zóna horní (odběr zboží vzhůru nataženou rukou).*

#### Obvyklé prezentační techniky:

Vertikální prezentace – vhodná pro větší prodejní jednotky, zboží stejného druhu je prezentováno v regálu pod sebou, šíře nabídky je zdůrazňována v horizontálním směru.

Horizontální prezentace – vhodná pro menší prodejní jednotky, je opakem vertikální prezentace.

Otevřená prezentace – aktivní přístup zákazníka, který si zboží prohlíží, zkouší, porovnává s jiným.

Tématická prezentace – různé druhy zboží prezentované pod jedním společným tématem.

Prezentace životního stylu – prezentace různých druhů zboží určených pro segment zákazníků s podobným životním stylem.

Prezentace příbuzného zboží – zákazník je při nákupu určitého druhu zboží myšlenkově veden ke zboží komplementárnímu.

Prezentace v blocích – prezentace vhodná pro zboží nové, unikátní, za speciální cenu, náročná na prodejní prostor.

Tento popis obvyklých prezentačních technik je možné nalézt v publikacích od CIMLERA (1998), a CIMLERA a ZADRAŽILOVÉ (2007).

## 4 Manažerský simulátor

### 4.1 Mentální modely

MILDEOVÁ a VOJTKO (2008 str. 37) definují mentální model takto: *Pod pojmem mentální model chápeme veškeré naše představy, týkající se konkrétního problému.*

Mentální modely reprezentují okolní svět a vztahy mezi jeho částmi. V jejich rámci jsou stanoveny hranice problémů (uvažované proměnné) a rovněž časový horizont. Mentální modely působí také jako informační filtr.

Okolní svět je velmi složitý a tudíž jeho reprezentace mentálními modely je značně omezená. Mentální modely jsou často nekompletní, mlhavé a jsou postaveny na nedostatečně formulovaných předpokladech.

CAHLÍK a kol. (2006, str. 7) uvádí: *Realita je ve společenských vědách a ekonomii často chápána jako komplexní systém.*

Naše schopnosti zpracování komplexity jsou značně omezené, jak uvádí MILDEOVÁ a VOJTKO (2008). Je možné ale využít jako nástroj či pomůcku počítače. Počítačové modely se dělí do dvou hlavních kategorií, jsou to optimalizační modely a simulační modely.

První myšlenka na využití počítačů jako podpory pro rozhodování se objevila již v 60. letech 20. století, uvádí WIERENGA (2008). V 70. letech se pak rozvíjela aplikace operačního výzkumu v marketingu. Používané modely ovšem vyhovovaly spíše pro „tvrdé“ systémy, tedy takové, které mají jasně definované problémy. Toto ovšem není případ marketingu, použití se tedy ukázalo jako diskutabilní.

#### 4.1.1 Simulační modely

V rámci této práce se dále budeme zabývat pouze modely simulačními. Latinské slovo „simulare“ znamená napodobovat. Cílem simulačních modelů je tedy napodobit chování nějakého systému, abychom jej mohli lépe prozkoumat.

Díky vývoji výpočetních technologií je dnes možné vytvářet simulované virtuální světy. V těchto světech probíhají interakce podobné interakcím ve světě reálném. Ovšem na rozdíl od světa reálného, uživatel simulačních modelů má pod kontrolou různé parametry modelu a může provádět experimenty, jejichž realizace by v reálném světě byla značně problematická, uvádí VOJTKO (2010).

Simulační modely lze dále rozdělit na dvě základní skupiny. Multiagentní simulační modely a modely systémové dynamiky. V rámci této práce se budeme dále zabývat pouze multiagentními systémy.

### 4.2 Multiagentní systém

*Zřejmě nejvýznamnějším ze zakladatelů multiagentního přístupu je Thomas Schelling (1969, 1978), který v roce 2005 obdržel za svou průkopnickou práci Nobelovu cenu za ekonomii, uvádí VOJTKO (2010, str. 50)*

CAHLÍK (2006) vysvětluje pojem „agent“, se kterým se v multiagentních systémech často setkáváme. Tento pojem je využíván místo pojmu podsystem. Tím se vyjadřuje jednak jeho dílčí samostatnost, ale rovněž i jeho inteligence při rozhodování.

PYKA a FAGIOLO (2005) agenty definují jako „populaci“ určitého systému. Agentem může být např. člověk, firma, město, počítačový program atd.

*Přístup k řešení úloh v multiagentním systému spočívá v hledání postupů, jak zajistit vhodnou a tedy účinnou spolupráci mezi libovolnými samostatnými agenty. Snahou je maximálně zachovat autonomnost a vytvořit motivující podmínky, které dovolí dosáhnout agentům v rámci jejich spolupráce požadovaného řešení úlohy. Takto vysvětluje CAHLÍK (2006, str. 25) princip řešení úloh v multiagentním systému.*

CAHLÍK (2006) dále uvádí, že John H. Holland představil metodu, jak do multiagentních systémů začlenit adaptabilní chování agentů. Jedná se o systém založený na množině jednoduchých logických pravidel. Tato pravidla se nazývají klasifikátory a každé z nich má strukturu: „POKUD <podmínka> POTOM <akce>“.



Agent je rovněž schopen využít dané pravidlo i tehdy, pokud se určitá situace neshoduje s podmínkou ve všech znacích, ale jen v těch, které agent pokládá za významné.

### 4.2.1 Multiagentní simulační model

CAHLÍK (2006) popisuje práci s multiagentním simulačním modelem. Studium určitého systému probíhá ve dvou základních krocích. Prvním krokem je sestavení modelu, druhým krokem je experimentování s ním.

Multiagentní simulační model realizovaný počítačem je tvořen dvěma základními typy pravidel. Tato pravidla jsou „spontánní“ a „vytvořená“. Agentům je určeno, jaký je jejich cíl, jaké proměnné mají sledovat a optimalizovat. Způsob, jakým tohoto cíle dosáhnou, je ponechán na nich samotných. Mohou využívat všechny dostupné informace a přizpůsobovat své chování vzhledem k podmínkám.

*Návrh multiagentního systému je obecně složitý díky své nesmírné univerzálnosti, a proto ani neexistuje jeho formalizovaný postup, i když se snahy návrh formalizovat čas od času objevují, uvádí CAHLÍK (2006, str. 25 – 26).*

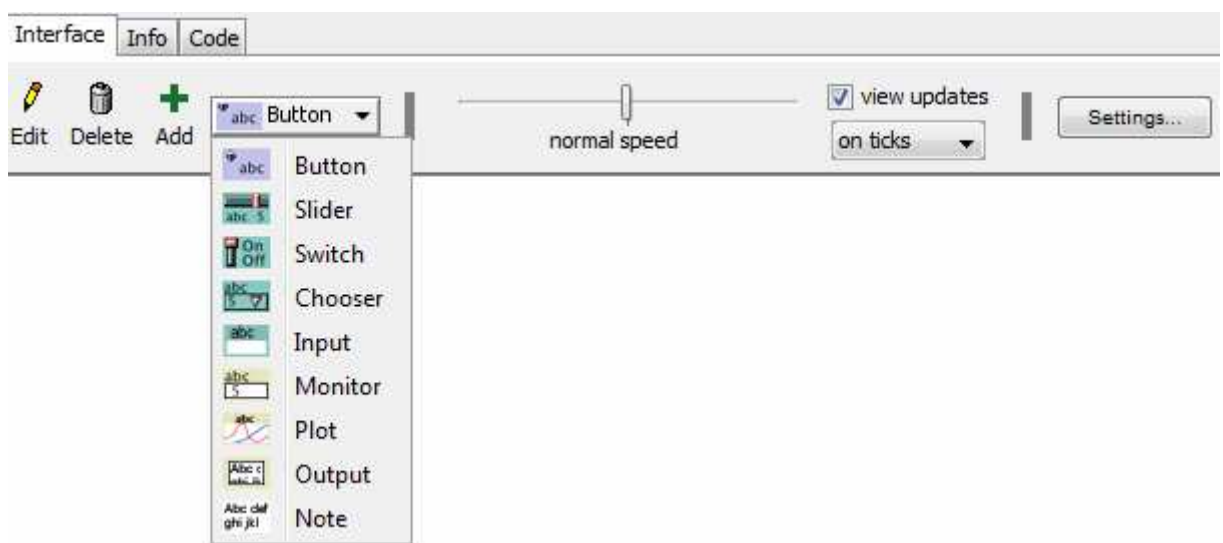
Multiagentní systémy se proto navrhují individuálně pro konkrétní úlohy. Každý autor či tým řešitelů tvoří unikátní multiagentní systém. Nejčastěji se jedná o nejjednodušší typy multiagentních systémů.

## 5 Výchozí simulační model - představení

Základem, na který tato diplomová práce navazuje, je multiagentní simulační model maloobchodní jednotky. Tento model byl vytvořen vedoucím této diplomové práce, panem Ing. Viktorem Vojtkem, Ph.D. v roce 2012.

Program simuluje chod samoobslužné maloobchodní jednotky během jednoho dne. Otevírací doba této jednotky je od 8:00 do 20:00. V modelu je nastaveno velké množství proměnných a nechybí zde na několika místech i prvek náhody. Uživatel má možnost zahrát si na manažera, nastavit určité parametry a sledovat důsledky svých rozhodnutí. Layout (tj. model dispozičního řešení) řízeného supermarketu vidí z ptačí perspektivy a může sledovat pohyb zákazníků po prodejní jednotce.

*Obrázek 1: Hlavička simulačního modelu*



*Zdroj: vlastní zpracování*

Na obrázku 1 je zobrazena základní hlavička simulačního modelu. Obsahuje tři záložky.

Záložka „Interface“ zobrazuje základní obrazovku modelu. V této záložce se odehrává samotná simulace. Je zde možné upravovat, odebírat a přidávat různá tlačítka, choosery

(umožňují volbu varianty), grafy a další. Slider (posuvník) uprostřed nastavuje rychlost probíhající simulace. Zaškrtnutá možnost „view updates“ povoluje zobrazování pohybu zákazníků po obchodní jednotce. Tuto možnost je možné vypnout za účelem zrychlení probíhající simulace, neboť samotná simulace je poměrně hodně náročná na operační paměť počítače.

Záložka „Info“ obsahuje základní informace o simulačním modelu.

Záložka „Code“ obsahuje počítačový kód programu, který je možné v této záložce editovat.

Obrázek 2: Základní obrazovka simulačního modelu

Jeden den manažerem supermarketu ...

6 7

Hodiny 14:01  
Zákazníci s vozíky 140

© 2012 Viktor Vorko  
Ekonomická fakulta, Jihočeská univerzita v Českých  
Budějovicích

10

2

Návštěvnost  
Nákupní chování  
Nákladní nákup (20 položek)  
Frekvence-dopřívání-... 3x za den

3

Iniciální

4

Spust/Zastav

5

Pokladny-v-průzvu

Vášim úkolem je nastavením dopřívání zboží a počtu pokladen co nejlépe řídit sítěný supermarket, tj. dosáhnout co nejvyššího zisku.

Nejdříve nastavte návštěvnost, nákupní chování a frekvenci dopřívání zboží. Následně klikněte na tlačítko [Iniciální].

Poše spustíte simulaci (kliknutím na tlačítko [Spust/Zastav]). Za běhu simulace můžete měnit pokladny v průzvu. Simulaci lze zastavit opětovným kliknutím na tlačítko [Spust/Zastav].

1

131 počet

Zákazníci

0 15

Fronty

0 9

Délka fronty

Pokladna

100 %

Dostupnost zboží

0 364

minuty

Tržby celkem 278553

Náklady celkem 299282

Hospodářský výsledek celkem -20729

Náklady na pořízení zboží 185702

Réžie prodejny 100000

Náklady na dopřívání zboží a pokladni 13580

8

9

Zdroj: vlastní zpracování

Obrázek 2 zobrazuje základní obrazovku simulačního modelu (záložka „Interface“). Skládá se z několika částí – v obrázku popsány čísly.

1. Základní pokyny pro uživatele.
2. Choosery a slidery sloužící k volbě parametrů před spuštěním simulace.
3. Tlačítko Inicializuj – poté co jsou nastaveny parametry z bodu 1 uživatel klikne na toto tlačítko a model aplikuje zvolené nastavení. Po inicializaci je možné spustit simulaci.
4. Tlačítko Spust'/Zastav – tímto tlačítkem se spustí a zastavuje simulace.
5. Slider pokladny v provozu – tímto posuvníkem může uživatel měnit počet pokladen, které budou v provozu. Toto nastavení lze měnit operativně během simulace.
6. Hodiny – udávají čas, ve kterém se simulace nachází.
7. Zákazníci s vozíky – ukazuje aktuální počet zákazníků s vozíky.
8. Výstupy – v této části uživatel vidí průběžně náklady, tržby a hospodářský výsledek, kterého jím řízená jednotka dosahuje.
9. Grafy – graf Zákazníci zobrazuje počet zákazníků. Jsou barevně odlišeny podle legendy. Graf Dostupnost zboží zobrazuje procentickou dostupnost zboží v regálech, graf Fronty pak aktuální fronty na jednotlivých pokladnách. Všechny tři tyto grafy se aktualizují průběžně během simulace.
10. Layout prodejny – v této části vidí uživatel řízenou prodejnu z ptáčích perspektivy. Modrá barva představuje vstup do prodejny, zelená barva představuje vstup na prodejní plochu, která je značena barvou šedou. V zelenkavé zóně (mezi pokladnami a vstupem do prodejny) si zákazník bere a odkládá nákupní košík, ty jsou znázorněny barvou fialovou. Regály se zbožím jsou znázorněny oranžovou barvou, regály s vyšší pravděpodobností nákupu pak barvou žlutou. Červeně jsou znázorněny pokladny, tmavě oranžovými body pak inkasní místa. Zobrazování pohybu zákazníků lze vypnout (viz výše).

## 6 Případové studie vycházející ze základního modelu

Jedním z cílů této práce bylo vytvoření tří případových studií, vycházejících ze základního simulačního modelu. Byly vybrány tři existující samoobslužné maloobchodní jednotky, na které byly případové studie vypracovány. Jednalo se o supermarket, diskont a convenience store. Pro tvorbu těchto případových studií bylo nezbytné získat informace o těchto maloobchodních jednotkách. Za tímto účelem byl navržen a realizován marketingový výzkum, podrobněji popsáný v kapitole 7. Následně byl model upraven a zkalibrován podle výstupů z provedeného marketingového výzkumu. Byla vytvořena metodika tvorby případových studií a scénáře sloužící k ověření této metodiky.

### Supermarket

První vybranou maloobchodní jednotkou byl supermarket Billa, nacházející se blízko centra Českých Budějovic. Na první pohled působí prodejna upraveným, čistým dojmem. Layout prodejny byl přibližně v polovině roku 2012 kompletně změněn. Vedle prodejny se nachází menší parkoviště, které není řešeno zrovna nejlépe a dochází na něm občas k problematickým situacím (nemožnost zajet dostatečně blízko k místu kontroly parkovacího lístku a posléze nutnost vystoupení z vozu, obtížná manipulace s vozidlem v úzkých uličkách mezi parkovacími místy).

### Diskont

Druhým zkoumaným subjektem byla diskontně orientovaná maloobchodní jednotka Lidl. Tato jednotka se nachází blízko zkoumaného supermarketu, tedy rovněž v docházkové vzdálenosti od centra Českých Budějovic. Prodejna je větší než zkoumaný supermarket a to jak velikostí budovy, tak i velikostí prodejní plochy. Rozdíl však není markantní. Diskontní orientace je na jednotce na první pohled jednoznačně vidět. Prodejna není zdaleka tak upravená jako zkoumaný supermarket, interiér je řešen co možná nejjednodušeji. Layout prodejny je pravděpodobně standardizovaný (naprosto totožný layout byl zaznamenán v jiné jednotce tohoto řetězce v Plzni). Vedle prodejny je poměrně velké parkoviště a manévrování s vozidly je zde velmi snadné (včetně větších vozidel).

### Convenience store

Třetím zkoumaným subjektem byl convenience store. Tato jednotka se nachází rovněž v blízkosti centra Českých Budějovic. Ze všech zkoumaných prodejen se tato nejvíce odlišuje. Je jednoznačně nejmenší ze všech a není provozována řetězcem, ale živnostníky.

První dvě zkoumané jednotky si vzájemně silně konkurují, tato jednotka představuje konkurenci slabou. Dispozice stavby, ve které se jednotka nachází, neumožňuje příliš vysokou variabilitu při tvorbě layoutu. Dopravní dostupnost této prodejny je podstatně nižší než u předcházejících případů. V blízkosti prodejny lze sice zastavit, ale větší parkovací plocha chybí.

Tyto jednotky byly vybrány do výzkumu účelově pro svoji odlišnost. Byla předpokládána heterogenita naměřených hodnot zkoumaných veličit.

## 7 Marketingový výzkum

Pro účely kalibrace simulačního modelu bylo nutné v první řadě provést marketingový výzkum. Bylo potřeba získat základní představu o fungování samoobslužných maloobchodních jednotek, o počtu nakupujících, informace o layoutech maloobchodních jednotek a další informace.

Data získaná z tohoto výzkumu posléze sloužila jako základní vstupní data při úpravě kódu počítačového modelu.

Marketingový výzkum se uskutečnil v období od 1.2.2013 do 31.3.2013 v Českých Budějovicích. Byly vybrány tři samoobslužné maloobchodní jednotky, ve kterých byl výzkum proveden (viz předchozí kapitola).

### 7.1 Metoda výzkumu

Pro tento výzkum byla zvolena metoda pozorování, konkrétně se jednalo o nezúčastněné pozorování skryté (účastníci výzkumu nebyli informováni o tom, že jsou pozorováni). Výběr zkoumaných jednotek byl proveden účelově. Důvodem byla snaha o dosažení různorodosti zjištěných údajů.

### 7.2 Průběh sběru dat

Pozorování bylo uskutečněno v každé jednotce zvlášť. Každá tato jednotka byla sledována ve třech různých dnech v týdnu, konkrétně se jednalo o dny:

- úterý
- pátek
- neděle



V každém z těchto dní byla každá jednotka pozorována po dobu 4 hodin, průřezově během otevírací doby, jednalo se o časy:

- 10:00 – 11:00
- 13:00 – 14:00
- 16:00 – 17:00
- 19:00 – 20:00

Mimo pozorování v tyto doby byla prováděna pozorování dalších veličin, které nebyly závislé na čase, kdy bylo pozorování prováděno. Celková časová náročnost výzkumu bez času potřebného k vyhodnocení nasbíraných dat činila zhruba 45 hodin.

### 7.3 Sledované veličiny

Počet zákazníků – byl sledován počet zákazníků navštěvujících obchodní jednotku v uvedených časech.

Počet nakupovaných položek – u každé z obchodních jednotek byl průřezově spočítán počet nakupovaných položek u dvaceti zákazníků.

Layout – u jednotlivých prodejen byl graficky znázorněn jejich layout.

Pokladny – byla sledována rychlost markování zboží na pokladnách.

Horké zóny – byla sledována místa nejvyšší koncentrací zákazníků v rámci prodejní plochy.

### 7.4 Výsledky výzkumu

#### Počet zákazníků

V tabulkách 3, 4 a 5 jsou zaznamenány počty zákazníků v jednotlivých dnech a jednotlivých hodinách v supermarketu, diskontu a v convenience store. V posledních řádcích tabulek můžeme vidět součet zákazníků v jednotlivých dnech (pouze sledované

hodiny), v posledních sloupečcích vpravo pak průměrný počet zákazníků ve sledovaných hodinách.

*Tabulka 3: Počet zákazníků ve sledovaných dnech a hodinách v supermarketu*

Čas	Úterý	Pátek	Neděle	Průměr
10:00 - 11:00	211	189	148	182,67
13:00 - 14:00	216	193	121	176,67
16:00 - 17:00	246	228	196	223,33
19:00 - 20:00	118	75	87	93,33
Součet	791	685	552	

*Zdroj: vlastní zpracování*

*Tabulka 4: Počet zákazníků ve sledovaných dnech a hodinách v diskontu*

Čas	Úterý	Pátek	Neděle	Průměr
10:00 - 11:00	182	158	165	168,33
13:00 - 14:00	215	205	202	207,33
16:00 - 17:00	241	232	210	227,67
19:00 - 20:00	95	102	99	98,67
Součet	733	697	676	

*Zdroj: vlastní zpracování*

Tabulka 5: Počet zákazníků ve sledovaných dnech a hodinách v convenience store

Čas	Úterý	Pátek	Neděle	Průměr
10:00 - 11:00	55	48	33	45,33
13:00 - 14:00	44	46	43	44,33
16:00 - 17:00	37	31	34	34,00
19:00 - 20:00	17	24	24	21,67
Součet	153	149	134	

*Zdroj: vlastní zpracování*

Z výsledků vyplývá, že nejsilnějším ze sledovaných dní je pro supermarket úterý, následuje pátek, nejslabším dnem je neděle. Nejvíce zákazníků pak supermarket navštívilo mezi 16. a 17. hodinou (Tabulka č. 3).

U diskontu je situace obdobná jako u supermarketu. Nejsilnějším ze sledovaných dní je úterý, následuje pátek, nejslabším dnem je neděle. Nejvíce zákazníků pak diskont navštívilo mezi 16. a 17. hodinou. (Tabulka č. 4)

Convenience store se se svými hodnotami nejvíce odlišuje. Nejsilnějším dnem bylo z pohledu počtu zákazníků rovněž úterý, ovšem rozdíly mezi sledovanými dny jsou minimální. Oproti supermarketu a diskontu čelí prodejna největšímu náporu zákazníků v dopoledních a brzkých odpoledních hodinách (Tabulka č. 5).

### Nákupní chování

Tabulka č. 6 zobrazuje počet položek v košíku u dvaceti zákazníků v daných obchodních jednotkách. Tito zákazníci byli vybráni náhodně v různých dnech a v různých hodinách. Pro účely kalibrace simulačního modelu jsou zásadní hodnoty průměrů a směrodatných odchylek – v tabulce zvýrazněné poslední dva řádky.

Tabulka 6: Počet nakupovaných položek

Zákazník	Počet položek		
	Supermarket	Diskont	Conv. Store
1	17	19	4
2	7	9	1
3	17	6	6
4	29	18	7
5	9	10	2
6	23	9	10
7	10	6	6
8	11	11	3
9	9	2	1
10	12	11	5
11	3	33	4
12	28	38	3
13	21	8	4
14	3	45	7
15	40	18	12
16	1	2	3
17	18	6	4
18	15	3	6
19	32	32	8
20	18	15	1
Průměr	16,15	15,05	4,85
Směrodatná odchylka	10,38	12,52	2,96

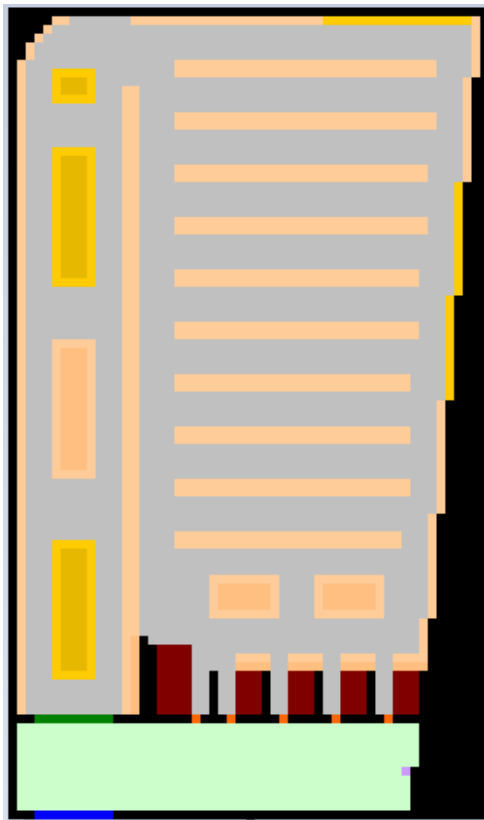
*Zdroj: vlastní zpracování*

Supermarket a diskont se svými hodnotami odlišují jen mírně. Convenience store se hodnotami odlišuje zásadně. Jedná se o malou prodejnu s užším sortimentem, která není obvykle využívána k velkým nákupům. Průměrný počet nakupovaných položek je podstatně nižší než u ostatních sledovaných jednotek, stejně tak i směrodatná odchylka.

#### Layout sledovaných jednotek

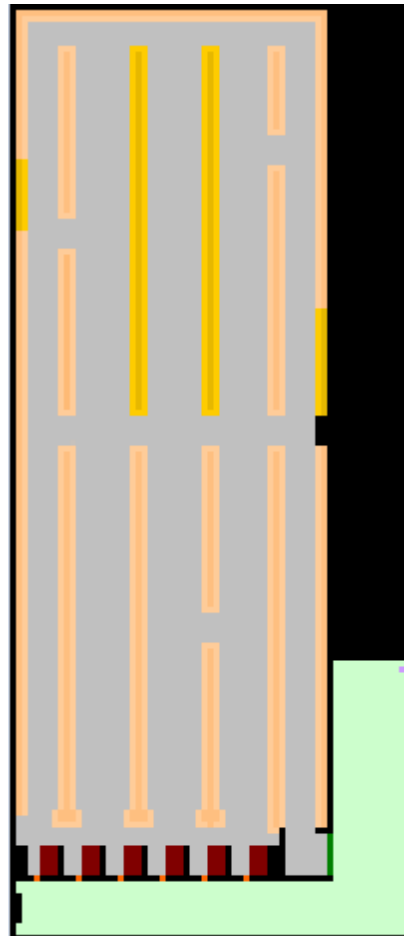
Obrázky 3, 4 a 5 zobrazují layouty prodejních jednotek překreslené do podoby, s níž pracuje simulační model. Při psaní tohoto dokumentu byly obrázky velikostně upraveny. Obrázky sloužící jako vstupní data pro simulační model byly překresleny ve stejném měřítku (1 pixel = 35 cm) a liší se svojí velikostí v závislosti na velikostech jednotlivých obchodních jednotek.

*Obrázek 3: Layout supermarketu*



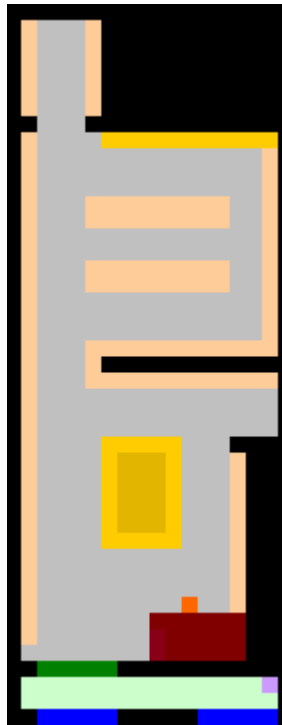
*Zdroj: vlastní zpracování*

*Obrázek 4: Layout diskontu*



*Zdroj: vlastní zpracování*

*Obrázek 5: Layout convenience store*



*Zdroj: vlastní zpracování*

### Závěr marketingového výzkumu

Primárním účelem tohoto výzkumu bylo získání základních dat a informací, na základě kterých bude možné provést kalibraci simulačního modelu. Výzkumem bylo možné sledovat pouze omezený počet faktorů, výsledky ovšem určují základní rámec, se kterým je v kalibraci pracováno.

## 7 Kalibrace modelu pro případové studie

### 7.1 Úvod kapitoly

Na základě informací získaných z marketingového výzkumu byly provedeny úpravy a kalibrace základního modelu pro případové studie. Jednotlivé podkapitoly popisují kalibraci určitých částí simulačního modelu, včetně úpravy programového kódu (v záložce Code) a uživatelského rozhraní (v záložce Interface). V rámci této kapitoly jsou také vysvětleny základní principy fungování simulačního modelu.

### 7.2 Layout

#### 7.2.1 Případová studie Supermarket

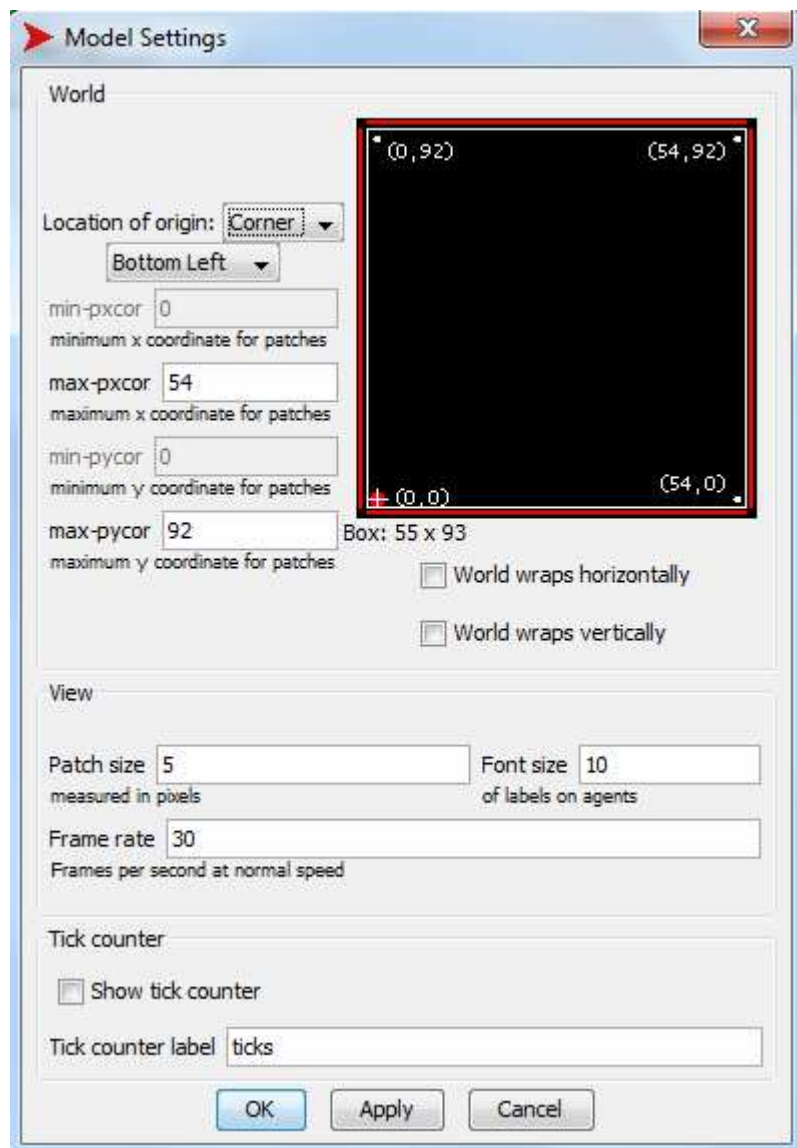
Nejprve musel být do modelu vložen obrázek layoutu supermarketu. Základní popis layoutu v modelu jsme již zmínili v kapitole 5. Layout pro případovou studii Supermarket je znázorněn na obrázku 3. Simulační model s layoutem pracuje pomocí barev, z toho důvodu je nezbytné při tvorbě nových layoutů původní barvy zachovat. Program reaguje na barvy stanovené v kódu modelu a podle nich interaguje v závislosti na nastavených proměnných. Obrázek byl vytvořen v programu Malování.

V kódu programu je obrázek vložen příkazem:

```
to setup
  clear-all
  import-pcolors "Layout supermarket.png"
  set-const
end
```

Výraz „Layout supermarket.png“ značí název obrázku, který je do modelu vložen. Tento obrázek se musí nacházet v adresáři modelu. Obrázek je veliký 55 x 93 pixelů a na hlavní stránce modelu (záložka Interface) se po inicializaci špatně zobrazoval. Proto bylo nutné v záložce Interface upravit nastavení zobrazování layoutu. (obrázek 6).

Obrázek 6: Nastavení zobrazování layoutu v záložce Interface



*Zdroj: vlastní zpracování*

Podle velikosti obrázku bylo nutné nastavit hodnoty „max-pxcor“ a „max-pycor“. Způsobem pokus-omyl bylo zjištěno, že zadané hodnoty musí být o 1 nižší, než je velikost obrázku. Hodnotou „Patch size“ se nastavuje velikost zobrazení layoutu v základní obrazovce modelu. Velikost této hodnoty už nemá vliv na správnost zobrazování po inicializaci modelu, jedná se pouze o estetickou záležitost.



Každému příchozímu zákazníkovi jsou modelem vygenerovány položky, které si chce v prodejně nakoupit. Jejich počet je v kódu programu definován střední hodnotou a směrodatnou odchylkou, je zde tedy zakomponován prvek náhody. Položky jsou náhodně rozmístěny po prodejně, u regálů označených žlutě je vyšší pravděpodobnost výskytu položky. Pohyb zákazníků po prodejně je řízen algoritmem šíření vlny. Zákazníci tedy automaticky hledají nejkratší cestu po prodejně k vygenerovaným položkám a k pokladně. Rychlost pohybu zákazníků po prodejně je dána počtem kroků za minutu. 1 krok = 1 pixel, v případě layoutů vytvořených pro kalibraci 1 pixel = 35 centimetrů.

### 7.2.2 Případové studie Diskont a Convenience store

Pro další dvě případové studie byly rovněž vytvořeny obrázky layoutů (obrázky 4 a 5). Při tvorbě těchto layoutů bylo zachováno stejné měřítko, jako v případě prvním (1 pixel = 35 centimetrů). Všechny tři obrázky se proto vzájemně odlišují svojí velikostí, neboť i zvolené obchodní jednotky měli různou rozlohu. Obrázky layoutů byly do modelů vloženy stejným příkazem, jakým byl vložen obrázek u případové studie Supermarket. Kód se liší pouze názvem obrázku, který je vkládán. Rovněž bylo nutné upravit zobrazování layoutů v záložce Interface. Stejně jako v případě výše byly hodnoty „max-pxcor“ a „max-pycor“ nastaveny o 1 nižší, než činily velikostní proporce obrázků.

### 7.2.3 Chybný výpočet zásoby zboží

Při další práci bylo zjištěno, že obrázky layoutů je nutné ještě upravit. Každý regál, se kterým model počítá, je představován jedním pixelem příslušné barvy. Zákazníci mohou v prostředí modelu interagovat pouze s regály, které sousedí s šedivou plochou. Pouze po šedivé ploše se totiž mohou zákazníci pohybovat. Toto ovšem po vytvoření obrázků neplatilo pro všechny regály, respektive pro všechny pixely příslušných barev, které regály představují. Obrázky byly vytvořené tak, aby co nejvíce odpovídaly reálnému dispozičnímu řešení prodejen. Aby toto platilo, byly do obrázků zakresleny i regály (pixely příslušných barev), které s šedivou plochou nesousedí. Příčinou tohoto

byla různorodá velikost regálů v prodejnách. Z příslušných regálů zákazníci nemohly odebírat zboží, ale při kalkulaci procentuální dostupnosti zboží byly tyto regály do výpočtů rovněž zahrnuty. Z tohoto důvodu zobrazovala procentuální dostupnost zboží zavádějící data. Procentuální dostupnost zboží v prodejně byla ve skutečnosti nižší, než hodnota modelem zobrazována. Proto byly pixely představující regály, které nesousedí s šedivou plochou, v obrázku vyznačeny tmavším odstínem příslušných barev, simulační model již při výpočtech tyto regály nyní nebere v potaz.

### 7.3 Návštěvnost

Návštěvnost (počet zákazníků) je do modelu explicitně vložena. Pro každou hodinu otevírací doby je stanoven počet zákazníků, který do jednotky přijde. Tito zákazníci se náhodně rozloží do probíhající hodiny. Pro kalibraci byla použita data z marketingového výzkumu týkající se návštěvnosti (tabulky 3, 4 a 5).

Stejně jako v základním modelu si uživatel může zvolit jednu ze tří možností nastavení (návštěvnost „Slabá“, „Střední“, „Silná“).

Na základě naměřených hodnot z marketingového výzkumu byly stanoveny počty zákazníků pro jednotlivé hodiny otevírací doby a tyto hodnoty byly použité pro nastavení „Střední“ hodnoty návštěvnosti místo původních hodnot. Dále byly tyto hodnoty zvýšeny a sníženy o 10 %. Tím byly definovány ostatní možnosti nastavení míry návštěvnosti.

Do počítačového kódu byly hodnoty vloženy příkazem:

```
to set_const
  if (Návštěvnost = "Slabá") [set hourly-arrivals (list 140 144 162 144 135 162 171 189 198 171 126 86)]
  if (Návštěvnost = "Střední") [set hourly-arrivals (list 155 160 180 160 150 180 190 210 220 190 140 95)]
  if (Návštěvnost = "Silná") [set hourly-arrivals (list 171 176 198 176 165 198 209 231 242 209 154 105)]
end
```

Hodnoty použité v kódu výše platí pro případovou studii Supermarket. Podle hodnot zjištěných v marketingovém výzkumu byla obdobně provedena úprava kódu i u modelů případových studií Diskont a Convenience store. Čísla v kódu představují počty zákazníků, kteří prodejnu v danou hodinu navštíví při jednotlivých variantách nastavení tohoto parametru.

## 7.4 Nákupní chování

Jak bylo řečeno výše, každému zákazníkovi je vygenerován určitý počet položek, který si chce koupit. Tento počet je definován střední hodnotou a směrodatnou odchylkou. V původním modelu si uživatel mohl zvolit jednu ze tří možností nákupního chování. “Velké nákupy”, “Střední nákupy” a “Malé nákupy”. Tato možnost volby byla při kalibraci ponechána, ovšem bylo nutné upravit střední hodnoty a směrodatné odchylky, které jsou touto volbou uživatelem vybrány.

Při kalibraci byla použita data z marketingového výzkumu týkající se nákupního chování (Tabulka 6). Získané hodnoty byly použity jako hodnoty pro “Střední nákupy” a poměrným zvýšením a snížením těchto hodnot byly definovány ostatní možnosti nastavení nákupního chování.

Do počítačového kódu byly hodnoty vloženy příkazem:

```
to set_const
  if (Nákupní-chování = "Malé nákupy (13 položek)") [
    set purchase-size 13
    set purchase-std-dev 8
  ]
  if (Nákupní-chování = "Střední nákupy (16 položek)") [
    set purchase-size 16
    set purchase-std-dev 10
  ]
  if (Nákupní-chování = "Velké nákupy (19 položek)") [
    set purchase-size 19
    set purchase-std-dev 12
  ]
end
```

Data v kódu zobrazeném výše jsou platná pro případovou studii Supermarket. Analogicky byla vložena data do modelů případových studií Diskont a Convenience store, střední hodnoty a směrodatné odchylky byly upraveny podle tabulky 6.

## 7.5 Nákupní vozíky

Původní model obsahuje několik proměnných, které se týkají nákupních vozíků a reakce zákazníků na dostupnost nákupních vozíků.

Jedná se o tyto proměnné:

```
cars_capacity      ; počet nákupních vozíků k dispozici
pcarts            ; patch, na kterém bude začínat fronta vozíků
pcarts_dir        ; směr zásobníku vozíků (1 - doprava, -1 - doleva)
pcarts_rows       ; v kolika řadách budou vozíky umístěny
pcarts_cap        ; počet vozíků na 1 patch
prob_departure    ; pravděpodobnost odchodu v případě, že není nákupní vozík
```

Během pozorování v rámci marketingového výzkumu nebyl v žádné sledované prodejně zaznamenán nedostatek nákupních vozíků. Nebyl zaznamenán žádný případ, kdy by zákazník neměl k dispozici volný nákupní vozík nebo ruční nákupní košík. Z tohoto důvodu bylo rozhodnuto, že vliv dostupnosti nákupních vozíků bude při kalibraci modelů u případových studií eliminován.

Kód programu byl upraven tímto způsobem:

```
to set_const
  set pcarts one-of patches with [ pcolor = 117.7 ] ; nastaví podle barvy počáteční patch zásobníku vozíků
  set cars_capacity 5000 ; kolik vozíků je k dispozici pro zákazníky?
  set pcarts_dir -1 ; nové vozíky doleva
  set pcarts_cap 500 ; počet vozíků na 1 patch
  set pcarts_rows 2 ; 2 řady vozíků
  set prob_departure 0.5 ; pravděpodobnost odchodu v případě čekání na nákupní vozík
end
```

Počáteční patch zásobníku vozíků (výchozí bod řazení vozíků) je na obrázku layoutu znázorněn tečkou příslušné barvy (1 pixel). Od tohoto bodu jsou nákupní vozíky řazeny. Nastavením hodnoty 5000 u proměnné “cars\_capacity” je vyloučen vliv této proměnné v simulačním modelu, neboť toto číslo převyšuje počet zákazníků, který může v průběhu simulace do jednotky přejít. Pravděpodobnost odchodu zákazníka v případě čekání na nákupní vozík (proměnnou “prob\_departure”) tedy nebylo třeba měnit, neboť k této situaci během simulace nikdy nedojde. Zbývající proměnné (“pcarts\_dir”, “pcarts\_cap” a “pcarts\_rows”) mají v tomto případě pouze estetický účinek.

### 7.6 Rychlost pohybu zákazníka, rychlost odbavení zákazníků na pokladnách

V této části kalibrace modelů se vyskytl problém. Při simulacích, které měly ověřit funkčnost modelu při novém nastavení, bylo zaznamenáno, že zákazníci se hromadí v prodejně. Pohyb zákazníků po prodejní ploše byl pomalý a realizace nákupu trvala

příliš dlouhou dobu. Z toho důvodu bylo nutné provést další úpravu kódu programu. V něm je nastaveno, kolik kroků stihne zákazník udělat za jeden “tick” – jinak řečeno rychlost pohybu zákazníka po prodejní ploše. Jeden krok představuje jeden pixel a jeden “tick” jednu minutu. Jeden pixel dále představuje podle měřítka zpracovaných layoutů 35 centimetrů. V původním modelu toto měřítko použito nebylo a počet kroků za minutu byl nastaven na 5. Pokud by tato hodnota zůstala při kalibraci nezměněna, znamenalo by to, že zákazník se za minutu posune pouze o 1,75 metru. Tato hodnota je velmi nízká, což ukázaly simulace a důsledkem bylo ono zmíněné kumulování zákazníků v prodejně. Proto byla provedena změna nastavení a počet kroků byl zvýšen na 20 za minutu (za 1 tick). V přepočtu podle měřítka to znamená posun zákazníka o 7 metrů za minutu.

První tři kroky zákazníka zapsané v kódu vypadají takto:

```
to shopping
  ask customers with [ color = orange ] [
    ifelse (not empty? path) [
      ; 1. krok / 1 tick
      move-to first path
      set path but-first path

      ; 2. krok / 1 tick
      if (not empty? path) [
        move-to first path
        set path but-first path
      ]

      ; 3. krok / 1 tick
      if (not empty? path) [
        move-to first path
        set path but-first path
      ]
    ]
  ]
```

Kód pokračuje ve stejné podobě až do 20. kroku a je zakončen příkazem “end”.

Výše zmíněnou úpravou nebyl problém zcela vyřešen. Při dalších ověřovacích simulacích bylo zpozorováno, že zákazníci se pohybují sice dostatečně rychle, ale hromadí se ve frontách u pokladen, přestože jich byl otevřený dostatečný počet. Bylo tedy nutné provést další úpravu. Problematickou proměnnou byla hodnota počtu položek, které pokladna stihne za minutu odbavit. Původně byla tato hodnota nastavena na 12 položek, došlo ke zvýšení na 30 položek u případových studií Supermarket a Diskont.

Příkaz v kódu vypadá takto:

```
to set_const
  set cashout_capacity 30
end
```

U případové studie Convenience store byla hodnota počtu položek odbavených pokladnou za minutu nastavena na 20 položek. Příčinou tohoto rozdílu oproti ostatním případovým studiím bylo to, že v této prodejně není používána technologicky vyspělejší pokladna schopná číst čárové kódy, ale pokladna manuální, do které je nutné zadávat ručně cenu každého nakupovaného výrobku. Rychlost odbavení je tudíž u této pokladny nižší.

## 7.7 Doplnování zboží do regálů

### 7.7.1 Případová studie Supermarket

Každý regál v prodejně má určitou kapacitu a je představován pixelem příslušné barvy. Obsáhne tedy jen určité množství zboží, které zákazníci postupně odebírají.

Kapacita regálů je v kódu vyjádřena tímto způsobem:

```
to setup
  ask patches with [pcolor = color-shelf1 or pcolor = color-shelf2] [
    set goods-capacity 5 + random 151 ]
end
```

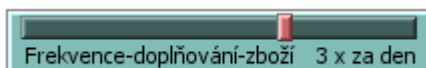
Z kódu je patrné, že konečná hodnota je tvořena částí pevnou a částí, která zahrnuje prvek náhody.

V průběhu simulace zákazníci zboží z regálů odebírají a tím klesá zásoba zboží. Pokud je zákazníkovi vygenerována určitá položka a on ji v regálu nenajde, pokračuje v nákupu a danou položku si koupit nemůže. Nízká dostupnost zboží tedy představuje pokles tržeb a s ním i dosahovaného zisku.

Před inicializací modelu má uživatel možnost zvolit na příslušném slideru (posuvníku) frekvenci doplňování zboží (obrázek 7). Minimální hodnota je stanovena v kódu na 1 a

znamená, že regály jsou zcela doplněné na začátku simulace (dojde k doplnění zboží do regálů před otevřením prodejny). Maximální volitelná hodnota je stanovena na 4.

Obrázek 7: Slide volby frekvence doplňování zboží



Zdroj: vlastní zpracování

Každé doplnění zboží ovšem současně představuje náklady. Jednotka musí zaplatit zaměstnance, kteří doplnění provedou. Tyto náklady se na základní obrazovce modelu zobrazují v kolonce “Náklady na doplňování zboží a pokladní”. Částka, o kterou se zvednou náklady při každém doplnění zboží, byla v základním modelu stanovena pevně. Částka je při simulaci přičtena za každé doplnění zboží vyjma doplnění prvního, před začátkem simulace. Toto první doplnění nebylo vůbec do nákladů zahrnuto.

Obrázek 8: Procentuální dostupnost zboží v prodejně během simulace



Zdroj: vlastní zpracování

Obrázek výše znázorňuje graf exportovaný z modelu, který zobrazuje procentuální dostupnost zboží v prodejně během simulace. Během ní došlo ke třem doplněním zboží, jednomu na začátku simulace a dvěma v průběhu.

Před úpravou barev regálů v obrázcích layoutů prodejen byly výsledky v grafu na obrázku 8 chybné. Graf počítal dostupnost zboží ze všech regálů, tedy i těch, které samostatnými regály nebyly. Graf tedy zobrazoval vyšší dostupnost zboží, než jaká ve skutečnosti byla. Úpravou barev byl tento problém vyřešen.

Při kalibraci byl nejdříve do modelu zahrnut náklad na první doplnění zboží.

Příkaz v kódu programu zněl takto:

```
to set_const
  set other-cost 8000
end
```

Hodnota 8000 představuje náklad v Kč na první doplnění zboží do regálů.

Dále bylo nutné změnit částku, která se k nákladům přičítá při každém dalším doplnění zboží.

Příkaz v kódu programu byl upraven takto:

```
to goods_replenishment
  let replenishment-times (list)
  let i 1
  repeat Frekvence-doplňování-zboží - 1 [
    set replenishment-times lput (i * round (720 / Frekvence-doplňování-zboží)) replenishment-times
    set i i + 1
  ]
  if member? ticks replenishment-times [
    ask patches with [pcolor = color-shelf1 or pcolor = color-shelf2] [
      set goods-amount goods-capacity
    ]
    set other-cost other-cost + 8000
  ]
end
```

Číslo 8 000 v kódu značí částku v Kč, o kolik se zvýší other-cost při každém doplnění zboží vyjma prvního.

Posléze bylo rozhodnuto model v tomto bodě rozšířit. Pro uživatele byla přidána možnost volby výše mezd, které jsou vypláceny pracovníkům doplňujícím zboží.



Za tímto účelem bylo nutné do počítačového kódu zadat novou proměnnou a upravit kód, aby reagoval na uživatelskou volbu.

Nová proměnná:

```
labor-costs          ; náklady na pracovníky doplňující zboží
```

Úprava kódu:

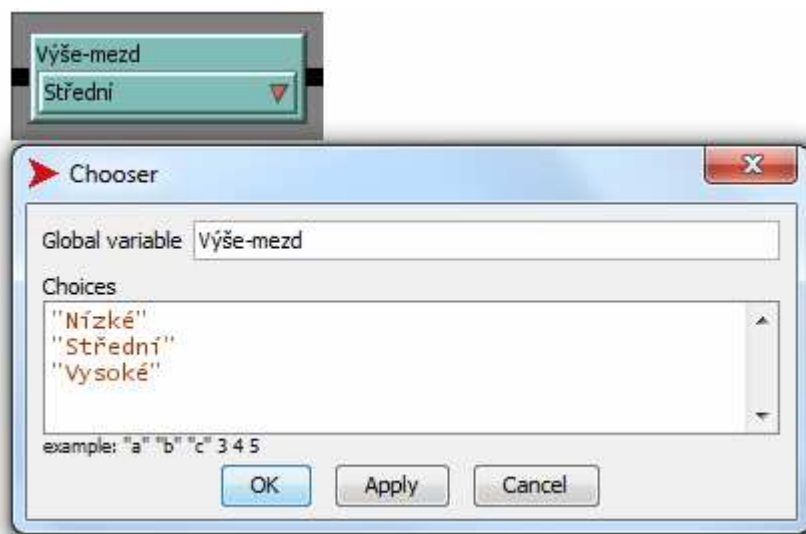
```
to set_const
  if (Výše-mezd = "Nízké") [
    set other-cost 8000
    set labor-costs 8000
  ]
  if (Výše-mezd = "Střední") [
    set other-cost 10000
    set labor-costs 10000
  ]
  if (Výše-mezd = "Vysoké") [
    set other-cost 12000
    set labor-costs 12000
  ]
end

to goods_replenishment
  let replenishment-times (list)
  let i 1
  repeat Frekvence-doplňování-zboží - 1 [
    set replenishment-times lput (i * round (720 / Frekvence-doplňování-zboží)) replenishment-time
    set i i + 1
  ]
  if member? ticks replenishment-times [
    ask patches with [pcolor = color-shelf1 or pcolor = color-shelf2] [
      set goods-amount goods-capacity
    ]
    set other-cost other-cost + labor-costs
  ]
end
```

Z kódu je patrná možnost nastavení třech hodnot výše mezd (8 000 Kč pro variantu “Nízké”, 10 000 Kč pro variantu “Střední”, 12 000 Kč pro variantu “Vysoké”). Touto volbou jsou definovány dvě proměnné. Proměnná “other-cost” značí náklady na první doplnění zboží a proměnná “labor-costs” je již zmíněná nová proměnná, která je využita v druhé části kódu místo původní pevné hodnoty 8 000 Kč.

V základní obrazovce modelu byl přidán nový chooser, který uživateli umožňuje volbu varianty (viz obrázek. 9). Tento chooser je nutné navolit před inicializací modelu.

Obrázek 9: Editace chooseru pro volbu varianty výše mezd



*Zdroj: vlastní zpracování*

### 7.7.2 Případové studie Diskont a Convenience store

U případové studie Diskont byly hodnoty nastaveny stejně, jako u případové studie Supermarket. U případové studie Convenience store bylo nutné zohlednit výrazně nižší velikost prodejní plochy. Náklady na doplňování zboží nemohly být stejně vysoké jako u ostatních případových studií. Z toho důvodu byly proměnné other-costs a labor-costs nastaveny pro jednotlivé varianty na hodnoty 1 800 Kč (varianta „Nízké“), 2 000 Kč (varianta „Střední“) a 2 200 Kč (varianta „Vysoké“).

Choosery u zbylých dvou modelů byly nastaveny totožně jako u případové studie Supermarket.

### 7.8 Režie prodejny

Režie prodejny v modelu představuje denní režijní náklady, se kterými je simulace zahájena. Stejně jako u nákladů na doplňování zboží i zde byla nejprve pouze

přenastavena její výše, posléze bylo rozhodnuto o rozšíření modelu v této oblasti za účelem zvýšení variability nastavení parametrů modelů.

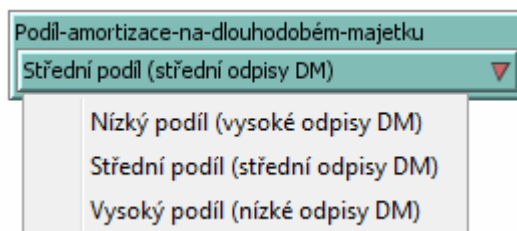
Logika tohoto rozšíření vyplývá z toho, že prodejna může mít různou výši amortizace dlouhodobého majetku. Výše amortizace bude mít vliv na režijní náklady prostřednictvím výše odpisů. Čím vyšší bude podíl amortizace na dlouhodobém majetku (tj. opotřebením majetku - co již bylo odepsáno), tím nižší budou odpisy (tj. co zbývá odepsat) a tím pádem bude i nižší režie prodejny (odpisy jsou součástí režijních nákladů).

Příkaz v kódu vypadá takto:

```
to set_const
  if (Podíl-amortizace-na-dlouhodobém-majetku = "Nízký podíl (vysoké odpisy DM)") [
    set overheads 320000
  ]
  if (Podíl-amortizace-na-dlouhodobém-majetku = "Střední podíl (střední odpisy DM)") [
    set overheads 300000
  ]
  if (Podíl-amortizace-na-dlouhodobém-majetku = "Vysoký podíl (nízké odpisy DM)") [
    set overheads 280000
  ]
end
```

Uživateli byly opět dány tři možnosti nastavení. Svoji volbu realizuje před inicializací modelu na hlavní stránce v chooseru, který byl nově přidán do modelů všech tří případových studií. (viz obrázek 10).

*Obrázek 10: Chooser pro volbu varianty podílu amortizace na dlouhodobém majetku*



*Zdroj: vlastní zpracování*

Nastavení zobrazené v kódu výše (tj. hodnota 320 000 Kč pro nízký podíl amortizace na dlouhodobém majetku, 300 000 Kč pro střední podíl, 280 000 Kč pro vysoký podíl) je platné pro případovou studii Supermarket. Pro ostatní případové studie byly zvoleny jiné hodnoty režijních nákladů. U případové studie Diskont byla velikost režie nastavena hodnoty 250 000 Kč, 230 000 Kč a 210 000 Kč pro jednotlivé varianty. Tedy režie nižší než u případové studie Supermarket. Důvodem byl rozdílný charakter prodejen. U případové studie Convenience store byly hodnoty velikosti režie nastaveny na 3 500 Kč, 3 000 Kč a 2 500 Kč. Důvodem byla výrazně nižší velikost prodejny.

Výše nastavených hodnot byly voleny takovým způsobem, aby obchodní jednotky při středních hodnotách nastavení volitelných proměnných dosahovaly přiměřeného zisku.

### **7.9 Prodejní marže**

Nastavení prodejní marže definuje, jaká část z tržeb obchodní jednotky představuje zisk. Při kalibraci byla i v tomto případě nejprve změněna samotná výše marže, posléze došlo i zde k rozšíření modelu.

Prodejní cena zboží je v modelu neměnná, výše marže ji tedy neovlivňuje. S marží je zde počítáno metodou shora. Tedy pokud od fixní prodejní ceny odečteme přírážku nastavenou v modelu, výsledkem budou náklady na pořízení zboží od dodavatelů. Úroveň marže v modelu tedy přímo ovlivňuje tyto náklady. Tento princip fungování modelu příliš neodpovídá realitě. V té je obvykle k pořizovací ceně zboží přičtena zvolená přírážka a výsledkem je prodejní cena. Model v současné podobě ovšem neumožňuje aplikaci tohoto principu, proto níže popsané rozšíření vychází ze současného stavu modelu.

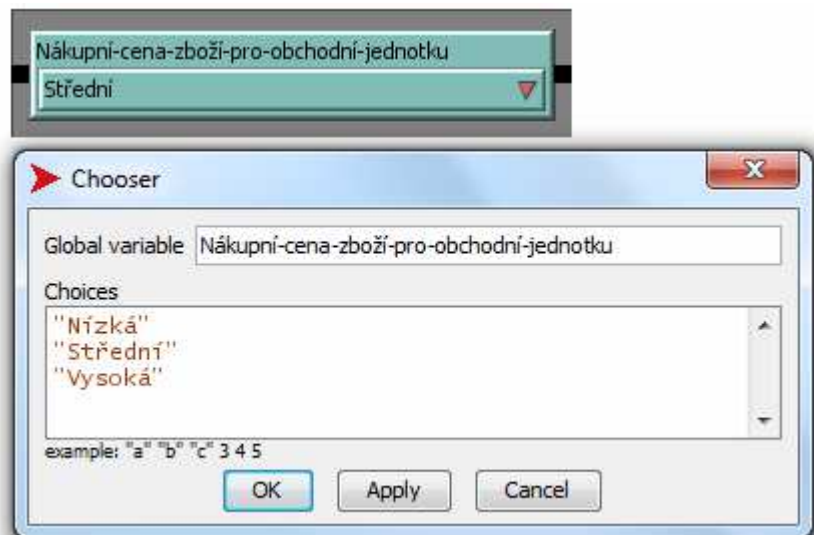
Rozšíření spočívá v tom, že byla uživateli dána možnost zvolit si výši nákupní ceny zboží pro obchodní jednotku, a to právě prostřednictvím nastavení výše marže. Čím nižší bude velikost marže, tím vyšší bude nákupní cena zboží.

Nastavení v kódu vypadá takto:

```
to set_const
  if (Nákupní-cena-zboží-pro-obchodní-jednotku = "Nízká") [
    set average-margin 0.42
  ]
  if (Nákupní-cena-zboží-pro-obchodní-jednotku = "Střední") [
    set average-margin 0.4
  ]
  if (Nákupní-cena-zboží-pro-obchodní-jednotku = "Vysoká") [
    set average-margin 0.38
  ]
]
end
```

Uživatel má možnost zvolit si ze třech možností nastavení nákupní ceny zboží pro obchodní jednotku ("Nízká" při velikosti marže 42 %, "Střední" při velikosti marže 0,40 % a "Vysoká" při velikosti marže 0,38 %), a to před inicializací modelu. Za tímto účelem bylo nutné na hlavní stránku modelu přidat nový chooser do všech modelů. (viz obrázek 11).

*Obrázek 11: Editace chooseru pro volbu varianty nákupní ceny zboží pro obchodní jednotku*



*Zdroj: vlastní zpracování*

Nastavení v kódu zobrazené výše je platné pro případovou studii Supermarket. Stejně jako v případě nastavení režie pro jednotlivé případové studie, i zde je třeba jednotlivé prodejny od sebe odlišit. Kvůli charakteru prodejen musí být marže u případové studie Diskont nižší než u případové studie Supermarket (hodnoty 32 %, 30 % a 28 %). U případové studie Convenience store pak musí být marže ještě nižší (hodnoty 17 %, 15 % a 13 %).

## 8 Metodika tvorby případových studií

V předcházející kapitole byla kalibrací modelů dokončena tvorba případových studií. Následující kapitola má za cíl vytvořit obecnou metodiku tvorby případových studií simulace chodu maloobchodní jednotky s využitím simulačního modelu. Metodika je vytvořena na základě poznatků vyplývajících z dosavadní práce.

### Definice maloobchodní jednotky

Nejprve musí být definována maloobchodní jednotka, pro kterou je případová studie vytvářena. V této práci byly případové studie vytvářeny na základě existujících maloobchodních jednotek, ta může být ale rovněž i fiktivní. V obou případech je však nutné ji definovat určitými parametry, jako jsou charakter prodejny, velikost prodejny, uspořádání prodejny atd. Dále je nutné definovat informace, které je nutné zjistit z marketingového výzkumu (např. průměrný počet nakupovaných položek, počet lidí nakupujících v prodejně v jednotlivých hodinách a další). V případě fiktivní jednotky nebude marketingový výzkum prováděn a potřebné informace je třeba stanovit na základě odhadu.

### Marketingový výzkum

Následuje návrh a realizace marketingového výzkumu (u případové studie založené na existující obchodní jednotce). Zjištěná data jsou pak využita při kalibraci simulačního modelu. Vhodnou metodou pro realizaci výzkumu je empirický výzkum provedený metodou pozorování. Pro zpracování dat je vhodný program Microsoft Excel.

### Tvorba layoutu prodejní jednotky

Touto částí začíná proces kalibrace simulačního modelu. Po definování prodejní jednotky je nutné vytvořit samotný layout obchodní jednotky. Je třeba využít nějaký grafický nástroj jako např. program Malování. Při tvorbě je nutné respektovat barvy, se kterými model pracuje. Hotový obrázek je posléze třeba vložit do kódu programu a v záložce Interface upravit jeho zobrazování.

### Editace proměnných v kódu programu a úprava záložky Interface

Na základě údajů zjištěných v marketingovém výzkumu nebo stanovených na základě kvalifikovaného odhadu je nutné upravit nastavení proměnných v kódu programu a v závislosti na tom editovat ovládací prvky v záložce Interface. Model je možné případně rozšířit, jako tomu bylo u případových studií vytvářených v rámci této práce. Volba proměnných, které jsou v počítačovém kódu editovány, závisí na zjištěných (v případě fiktivní obchodní jednotky stanovených) údajích. Řada proměnných se k editaci nabízí, jako například nákupní chování zákazníků (počet nakupovaných položek), počet lidí navštěvujících prodejnu, režie prodejny a další. S těmito proměnnými základní model již pracuje a jejich editaci je schopný provést i člověk bez hlubší znalosti programování (stejně jako autor této práce).

### Testování a zkušební simulace

V průběhu kalibrace je vhodné provádět testování modelu skrze zkušební simulace. Během tohoto testování je možné identifikovat špatně pracující prvky modelu a upravit je. Autor této práce se setkal během testování se dvěma základními problémy. Prvním bylo kumulování zákazníků v prodejně. Tento problém byl vyřešen úpravou rychlosti pohybu zákazníků a úpravou rychlosti odbavování zákazníků na pokladnách. Druhým problémem byl špatně pracující graf zobrazující dostupnost zboží v prodejně. Tento problém byl vyřešen editací barev v layoutech prodejen, díky které již model do výpočtu dostupnosti zboží nezahrnuje regály pro zákazníky nedostupné.



## 9 Scénáře

### 9.1 Úvod kapitoly

Tato kapitola se zabývá ověřením metodiky tvorby případových studií prostřednictvím scénářů. Bylo navrženo 5 scénářů, které byly aplikovány na vytvořené případové studie. U každého scénáře s výjimkou posledního byly simulace při každé variantě nastavení provedeny třikrát a výsledky byly zprůměrnovány. Důvodem opakování simulací byl prvek náhody, který je do simulačních modelů na několika místech zahrnut.

Během kalibrace modelů pro jednotlivé případové studie byl model několikrát rozšířen. V původním modelu mohl uživatel nastavit parametry „Návštěvnost“, „Nákupní chování“, „Frekvenci doplňování zboží“ a mohl v průběhu simulace řídit počet otevřených pokladen. Do modelu byly kalibrační vloženy nové ovládací prvky, se kterými může uživatel operovat. Těmito novými prvky jsou:

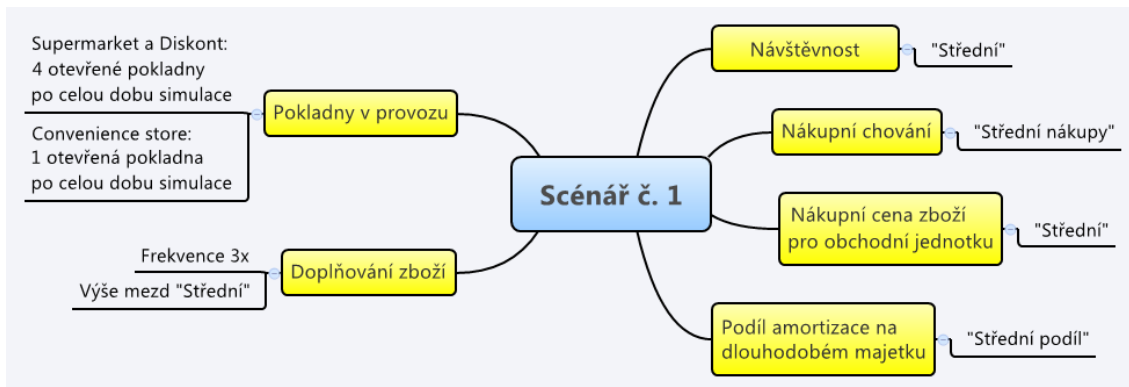
1. nastavení podílu amortizace na dlouhodobém majetku,
2. nastavení výše mezd pro pracovníky doplňující zboží do regálů,
3. možnost nastavení nákupní ceny zboží pro obchodní jednotku.

Funkčnost nových prvků je v této kapitole rovněž ověřována.

### 9.2 Scénář č. 1

Scénář č. 1 má za cíl ověřit obecnou funkčnost simulačních modelů. Nastavení modelů bylo provedeno podle schématu 1.

Schéma 1: Nastavení parametrů modelů u scénáře č. 1



Zdroj: vlastní zpracování

Kalibrace modelů byla prováděna takovým způsobem, aby všechny simulační modely dosahovaly při středních hodnotách nastavení volitelných proměnných přiměřeného zisku. V tabulce 7 můžete vidět hodnoty výstupů simulací u jednotlivých případových studií, hodnoty jsou průměrem výstupů ze tří provedených simulací. Ani v jednom případě simulace nevykázala zápornou hodnotu hospodářského výsledku, přesto se ale hodnoty z jednotlivých simulací od sebe lišily. Důvodem je již zmíněný prvek náhody.

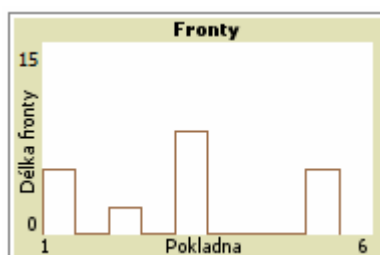
Tabulka 7: Vybrané výstupy provedených simulací u scénáře č. 1 (v Kč)

	Supermarket	Diskont	Convenience store
Průměrné tržby	1 326 927	1 365 974	100 169
Průměrné náklady	1 285 832	1 318 819	98 054
Průměrný hospodářský výsledek	41 095	47 155	2 115

Zdroj: vlastní zpracování

Z tabulky výše je patrné, že při daném nastavení dosahovaly všechny prodejny kladného hospodářského výsledku. U případových studií Supermarket a Diskont je hospodářský výsledek podobný. U případové studie Convenience store dosáhla prodejna sice rovněž zisku, ten je však výrazně nižší, než u ostatních případových studií. To logicky odpovídá předpokladům vzhledem k povaze prodejny.

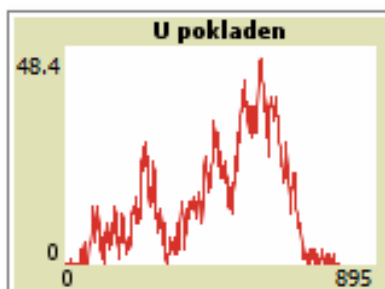
*Graf 1: Fronty na jednotlivých pokladnách*



*Zdroj: vlastní zpracování*

Graf 1 znázorňuje aktuální počet zákazníků u jednotlivých pokladen v určitém momentu probíhající simulace. V okamžiku zachyceném grafem jsou v provozu 4 pokladny a u každé je několik lidí čekajících na odbavení ve frontě. Tento graf byl pořízen během jedné z probíhajících simulací u tohoto scénáře u případové studie Supermarket.

*Graf 2: Celkový počet zákazníků čekajících ve frontách u pokladen*



*Zdroj: vlastní zpracování*

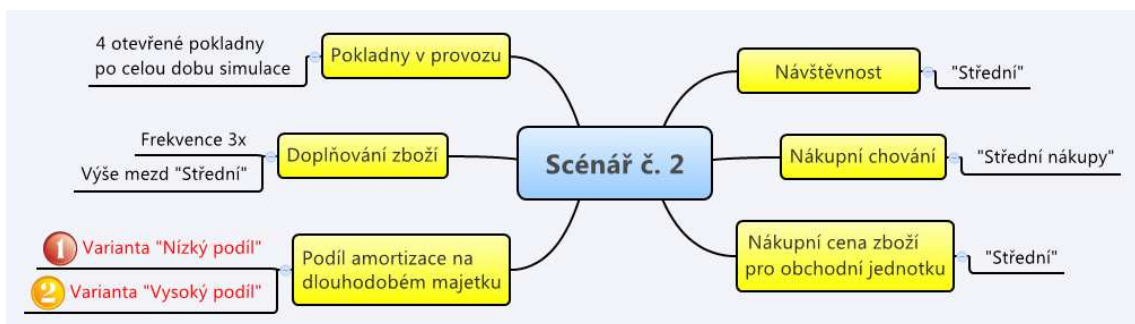
Graf č. 2 zobrazuje celkový počet zákazníků čekajících ve frontách u pokladen v průběhu simulace. Tento graf byl pořízen po skončení jedné ze simulací tohoto scénáře u případové studie Supermarket. Z grafu je patrné, že v době nejvyšší špičky se celkově u pokladen nacházelo zhruba 45 zákazníků, což při čtyřech otevřených pokladnách značí v průměru frontu cca 11 zákazníků na každé pokladně.

### 9.3 Scénář č. 2

Scénář č. 2 má za cíl ověřit funkčnost prvního nového prvku, tedy možnosti nastavení podílu amortizace na dlouhodobém majetku.

Pro účely tohoto scénáře byly zvoleny dvě varianty nastavení modelu. Tyto varianty jsou znázorněny ve schématu 2.

*Schéma 2: Nastavení parametrů modelů u scénáře č. 2*



*Zdroj: vlastní zpracování*

Tento scénář byl prováděn u modelu případové studie Supermarket. Schéma 2 značí 2 varianty nastavení modelu, přičemž tyto varianty se od sebe liší nastavením podílu amortizace na dlouhodobém majetku. Ostatní proměnné zůstaly beze změny. Simulace byly pro jednotlivé varianty provedeny třikrát a výsledky byly zprůměrovány.

*Tabulka 8: Vybrané výstupy provedených simulací u scénáře č. 2 (v Kč)*

Sledované hodnoty	Varianta 1	Varianta 2
Režijní náklady	320 000	280 000
Průměrné tržby	1 445 462	1 446 279
Průměrné celkové náklady	1 390 373	1 350 920
Průměrný zisk	55 089	95 359

*Zdroj: vlastní zpracování*

*Tabulka 9: Vybrané výstupy provedených simulací u scénáře č. 2  
(procentické vyjádření)*

Sledované hodnoty	Varianta 1	Varianta 2
Režijní náklady	100 %	88 %
Průměrné tržby	100 %	100 %
Průměrné celkové náklady	100 %	97 %
Průměrný zisk	100 %	173 %

*Zdroj: vlastní zpracování*

Tabulky výše zobrazují porovnání sledovaných hodnot u variant 1 a 2. V tabulce 8 jsou uvedeny výsledky v absolutních číslech. Tabulka 9 pak zobrazuje procentuální podíl výsledných hodnot z varianty 2 na variantě 1.

Hodnota režijních nákladů je přímo ovlivněna volbou varianty podílu amortizace na dlouhodobém majetku. Ostatní hodnoty v tabulkách pak zobrazují důsledky této volby. Průměrné tržby se vlivem odlišného nastavení dané proměnné zásadně nezměnily. Na průměrných celkových nákladech se změna nákladů režijních již mírně podepsala. Rozdíl nebyl vysoký, což je dáno tím, že režijní náklady tvoří pouze menší část nákladů celkových. Zásadní rozdíl můžeme vidět u hodnoty průměrného zisku. V absolutních

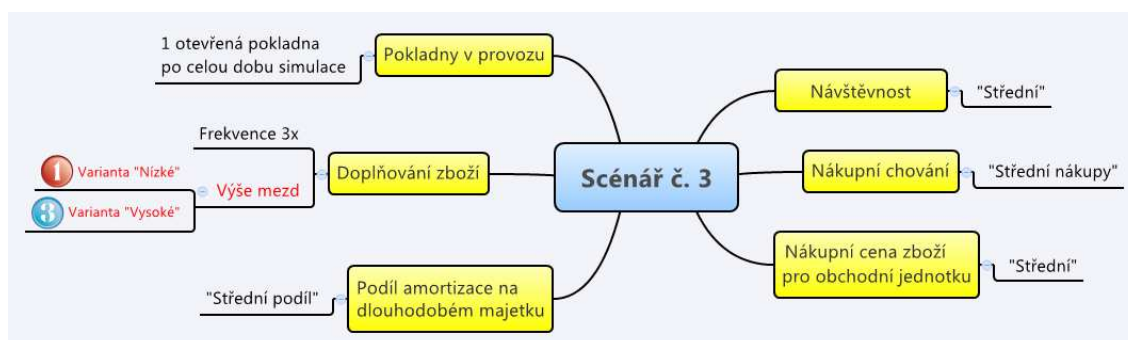
číslech byly režijní náklady sníženy o 40 000. O prakticky stejnou hodnotu v závislosti na tom vzrostl dosažený zisk.

### 9.4 Scénář č. 3

Scénář č. 3 má za cíl ověřit, jaký vliv na výsledky bude mít druhý nový prvek, tedy možnosti nastavení výše mezd pro pracovníky doplňující zboží.

Pro účely tohoto scénáře byly zvoleny dvě varianty nastavení modelu. Tyto varianty jsou znázorněny ve schématu 3.

Schéma 3: Nastavení parametrů modelů u scénáře č. 3



Zdroj: vlastní zpracování

Tento scénář byl prováděn u modelu případové studie Convenience store. Schéma 3 značí 2 varianty nastavení modelu, přičemž tyto varianty se od sebe liší nastavením výše mezd pro pracovníky doplňující zboží. Ostatní proměnné zůstaly beze změny. Simulace byly pro jednotlivé varianty provedeny třikrát a výsledky byly zprůměrovány.

*Tabulka 10: Vybrané výstupy provedených simulací u scénáře č. 3 (v Kč)*

Hodnota	Varianta 1	Varianta 2
Náklady na 1 doplnění zboží	1 800	2 200
Průměrné náklady na doplňování zboží a pokladny	7 350	8 550
Průměrné tržby	98 365	97 727
Průměrné celkové náklady	95 885	96 530
Průměrný zisk	2 480	1 197

*Zdroj: vlastní zpracování*

*Tabulka 11: Vybrané výstupy provedených simulací u scénáře č. 3  
(procentické vyjádření)*

Hodnota	Varianta 1	Varianta 2
Náklady na 1 doplnění zboží	100 %	122 %
Průměrné náklady na doplňování zboží a pokladny	100 %	116 %
Průměrné tržby	100 %	99 %
Průměrné celkové náklady	100 %	101 %
Průměrný zisk	100 %	48 %

*Zdroj: vlastní zpracování*

Tabulky výše zobrazují porovnání sledovaných hodnot u variant 1 a 2. V tabulce 10 jsou uvedeny výsledky v absolutních číslech. Tabulka 11 pak zobrazuje procentuální podíl výsledných hodnot z varianty 2 na variantě 1.

Hodnota nákladů na 1 doplnění zboží je přímo ovlivněna volbou varianty výše mezd pro zaměstnance provádějící doplňování zboží. Průměrné náklady na doplňování zboží a pokladny pak stouply na 116 % původní hodnoty. Na průměrné tržby neměla odlišná volba výše mezd dle předpokladů vliv. Absolutní rozdíl v nákladech na 1 doplnění

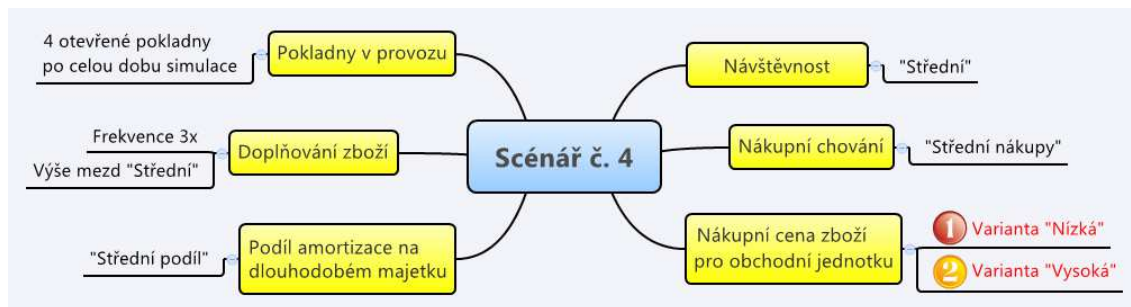
zboží mezi variantou 1 a variantou 2 činil 400 Kč. Při frekvenci třech doplnění za simulaci tento absolutní rozdíl činí 1 200 Kč. Vzhledem k výši celkových průměrných nákladů (cca 96 000 Kč) nepůsobí toto zvýšení v jejich procentuálním vyjádření zásadní rozdíl, má ovšem veliký vliv na dosahovaný výsledek hospodaření, jak je patrné z posledního řádku tabulek. U druhé varianty klesl dosahovaný zisk na 48 % hodnoty z varianty 1.

## 9.5 Scénář č. 4

Scénář č. 4 má za cíl ověřit funkčnost třetího nového prvku, tedy možnosti nastavení nákupní ceny zboží pro obchodní jednotku

Pro účely tohoto scénáře byly zvoleny dvě varianty nastavení modelu. Tyto varianty jsou znázorněny ve schématu 4.

*Schéma 4: Nastavení parametrů modelů u scénáře č. 4*



*Zdroj: vlastní zpracování*

Tento scénář byl prováděn u modelu případové studie Diskont. Schéma 4 značí 2 varianty nastavení modelu, přičemž tyto varianty se od sebe liší nastavením nákupní ceny zboží pro obchodní jednotku. Ostatní proměnné zůstaly beze změny. Simulace byly pro jednotlivé varianty provedeny třikrát a výsledky byly zprůměrovány.



*Tabulka 12: Vybrané výstupy provedených simulací u scénáře č. 4 (v Kč)*

Hodnota	Varianta 1	Varianta 2
Průměrné tržby	1 520 447	1 484 085
Průměrné variabilní náklady	1 151 854	1 159 441
Průměrné celkové náklady	1 419 864	1 427 558
Průměrný zisk	100 583	56 527

*Zdroj: vlastní zpracování*

*Tabulka 13: Vybrané výstupy provedených simulací u scénáře č. 4  
(procentické vyjádření)*

Hodnota	Varianta 1	Varianta 2
Průměrné tržby	100 %	98 %
Průměrné variabilní náklady	100 %	101 %
Průměrné celkové náklady	100 %	101 %
Průměrný zisk	100 %	56 %

*Zdroj: vlastní zpracování*

Tabulky výše zobrazují porovnání sledovaných hodnot u variant 1 a 2. V tabulce 12 jsou uvedeny výsledky v absolutních číslech. Tabulka 13 pak zobrazuje procentuální podíl výsledných hodnot z varianty 2 na variantě 1.

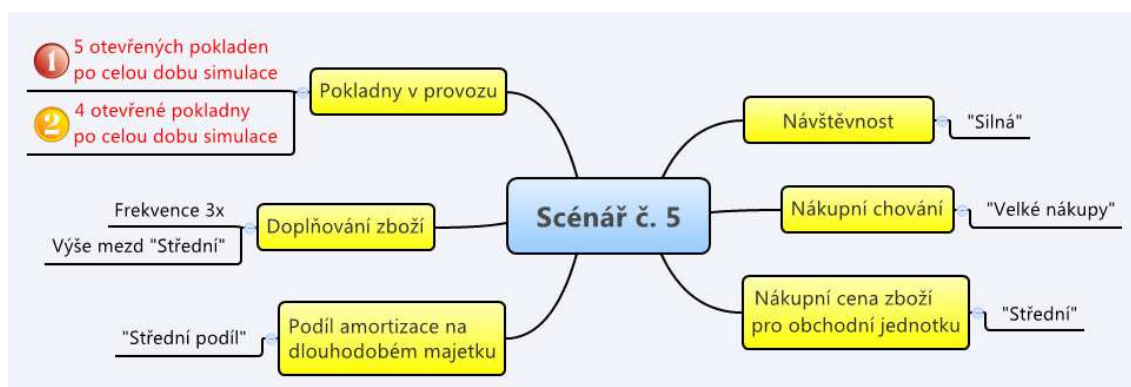
Z výsledků vyplývá, že zvolená varianta měla opět zásadní vliv na výši dosaženého hospodářského výsledku. Při zvýšení nákupní ceny pro supermarket se tržby nezměnily (mírný rozdíl je zapříčiněn prvky náhody, které jsou v modelu obsaženy).

U variabilních a celkových nákladů byla změna rovněž nepatrná v porovnání s jejich celkovou výší, zisk však klesl na 56 % původní hodnoty.

## 9.6 Scénář č. 5

Cílem tohoto scénáře je posoudit, jak se prodejna kapacitně vypořádá s vyšším zatížením. Toto zatížení představuje zvýšená návštěvnost a zvýšený počet nakupovaných položek, jak je znázorněno ve schématu 5.

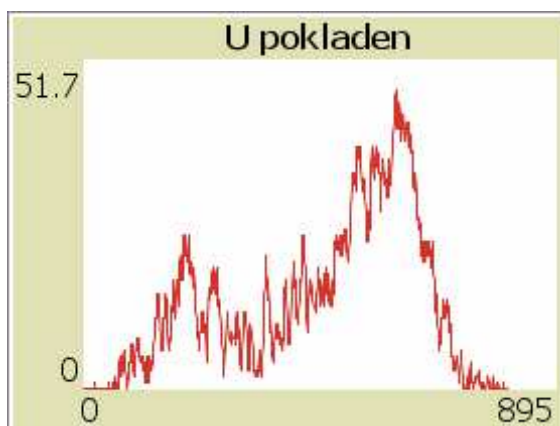
*Schéma 5: Nastavení parametrů modelů u scénáře č. 5*



*Zdroj: vlastní zpracování*

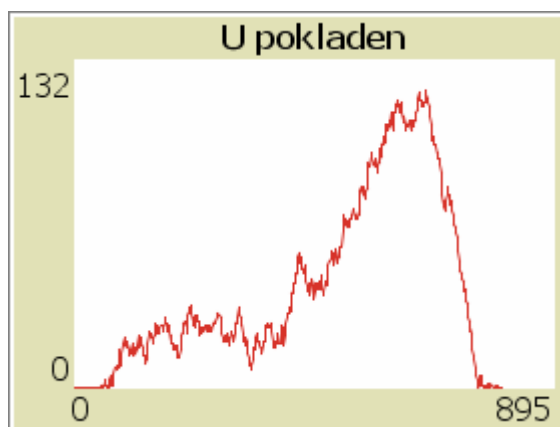
Tento scénář byl prováděn u případové studie Supermarket. Byly stanoveny 2 možnosti nastavení modelu, které se od sebe liší počtem otevřených pokladen během simulace. Simulace každé varianty byla provedena jednou.

*Graf 3: Celkový počet zákazníků čekajících ve frontách u pokladen při 1. variantě nastavení*



*Zdroj: vlastní zpracování*

*Graf 4: Celkový počet zákazníků čekajících ve frontách u pokladen při 2. variantě nastavení*



*Zdroj: vlastní zpracování*

Graf 3 a 4 znázorňují počet zákazníků čekající ve frontách u pokladen během simulace. U první varianty (graf 3), kdy byl počet otevřených pokladen nastaven na 5, můžeme vidět, že v době nejvyšší špičky na pokladnách čekalo zhruba 50 lidí. Při pěti otevřených pokladnách to značí v průměru frontu deseti lidí na každé pokladně. Tato

---

velikost front není ani v reálu nijak zvlášť neobvyklá a dá se tedy říci, že při všech otevřených pokladnách prodejna nápor zákazníků zvládla.

Odlišnou situaci představuje graf 4, který je výstupem simulace varianty 2. Během simulace této varianty byly po celou dobu simulace ponechány otevřené pouze 4 pokladny, což mělo fatální důsledky. Můžeme vidět, že pokladny v tomto případě představují úzké místo v simulačním modelu, kde se zákazníci kumulují. Kapacita pokladen byla nižší než nápor zákazníků a postupně docházelo k navyšování front až na v průměru cca 30 lidí na jednu pokladnu.

## 10 Návrhy

Simulační model chodu maloobchodní jednotky skýtá celou řadu příležitostí pro úpravy a rozšíření. Tento model má veliký potenciál do budoucna, v této části práce jsou popsány návrhy na úpravu tohoto modelu.

### Průběžné doplňování zboží do regálů

V současné době model pracuje tak, že je před inicializací modelu zvolena frekvence doplňování zboží. V závislosti na ní jsou pak regály v prodejně během simulace v určitý čas okamžitě doplněny do své maximální kapacity. Každé toto doplnění představuje finanční obnos, který se přičte k nákladům. Vhodnější a realističtější formou se jeví průběžné doplňování zboží do regálů. Bylo by vhodné model upravit takovým způsobem, aby si uživatel mohl zvolit určitý počet pracovníků, kteří budou po celou dobu simulace zboží do regálů doplňovat. Tito zaměstnanci by měli určitou produktivitu práce vyjádřenou počtem položek nebo regálů, které doplní za určité časové období. Kalibrační modelu byla uživateli přidána možnost volby výše mezd pro zaměstnance, kteří doplňování zboží provádějí, respektive byl kalibrační upraven náklad, který každé toto doplnění pro obchodní jednotku představuje. V souvislosti s výše uvedeným návrhem by bylo vhodné do modelu implementovat přímo volitelnou hodinovou mzdu pro zaměstnance, a to včetně pokladních. Výše této mzdy by poté měla další efekty, viz níže.

### Nastavení počtu otevřených pokladen před spuštěním simulace

Počet otevřených pokladen může uživatel volit operativně během probíhající simulace. Pro účely zobrazení důsledků manažerských rozhodnutí uživateli by bylo možné model upravit takovým způsobem, aby uživatel mohl předem navolit počet otevřených pokladen pro jednotlivé hodiny otevírací doby, a to podle svého odhadu potřeby. Po provedené simulaci by pak viděl důsledky své volby v podobně počtu lidí čekajících ve frontách u pokladen na odbavení.

### Volba marže, nákupní ceny a prodejní ceny

V původním modelu byla marže, nákupní cena i prodejní cena nastavena fixně. Po kalibraci bylo možné ovlivňovat nákupní cenu pro obchodní jednotku prostřednictvím nastavení odlišné úrovně marže. Jak již bylo zmíněno v kapitole 7.9, princip se kterým model v současné době pracuje neodpovídá realitě. Bylo by vhodné model upravit takovým způsobem, aby prodejní cena byla závislou proměnnou vycházející z nastavení nákupní ceny a výše přírážky.

### Index spokojenosti zaměstnanců

Dalším návrhem na úpravu modelu je implementace indexu zobrazující spokojenost zaměstnanců. Tento návrh souvisí s návrhy popsány výše. Index by byl ovlivňován především výší mzdy a měl by vliv na produktivitu práce zaměstnanců. Produktivitu práce by u pokladních představoval počet položek, které na pokladně odbaví během určitého časového úseku a u zaměstnanců doplňující zboží by se jednalo o počet položek nebo regálů, které za určitý časový úsek doplní.

### Index spokojenosti zákazníků

Implementace indexu spokojenosti zákazníků rovněž souvisí s návrhy popsány výše. Index by byl ovlivňován několika faktory. Jedním z klíčových by byla cenová hladina (prodejní cena zboží v obchodě), dále pak počet zákazníků v prodejně (míra přeplněnosti obchodní jednotky), velikost front u pokladen, případně dostupnost nákupních košíků. Bylo by vhodné rovněž implementovat nový prvek do vzorce chování zákazníků, totiž pravděpodobnost odchodu zákazníka z prodejny bez realizace nákupu v případě, že je před ním na pokladně určitý vyšší počet čekajících zákazníků (odchod v případě, že fronta je delší nebo rovna nastavené hodnotě). Při implementaci tohoto prvku by mohl vzniknout problém, který by představovalo zboží zákazníkem vložené do košíku a posléze zanechané v prodejně bez realizace nákupu. Toto zboží by mohlo být vráceno zpět do regálu pracovníky, kteří zboží do regálů doplňují.

### Opakování simulací s vlivem indexů

Implementované indexy by mohly být využity při opakování simulací. Hodnota indexů po první simulaci by představovala změnu parametrů pro simulaci následující. Například pokud by index spokojenosti zákazníků dosáhl během první simulace špatných hodnot, v další simulaci by byl snížen počet zákazníků, kteří do prodejny přijdou.

## 11 Závěr

Simulační modely mohou být v řadě případech využity jako systém pro podporu rozhodování v obchodě. Počítačové modely budou vždy zjednodušeným obrazem reality, avšak i přesto mohou být výstupy z nich plynoucí pro management obchodních jednotek velmi hodnotné.

Výchozí simulační model, na němž tato práce staví, byl popsán na začátku praktické části diplomové práce. Model se skládá ze dvou klíčových záložek, záložky Interface (uživatelské prostředí modelu) a záložky Code (počítačový kód modelu). Právě editace těchto dvou záložek je zásadní při kalibraci modelu, tedy tvorbě samostatných konkrétních modelů pro konkrétní případové studie.

### Návrh případových studií

Za tímto účelem byly nejprve navrženy tři případové studie, které se zakládají na existujících samoobslužných maloobchodních jednotkách v Českých Budějovicích. Jednalo se o tyto případové studie:

- Supermarket,
- Diskont,
- Convenience store.

Kalibrace simulačních modelů pro tyto případové studie byla podmíněna získáním základních údajů o fungování těchto maloobchodních jednotek.

### Marketingový výzkum

Z důvodu uvedeném výše byl navržen a proveden marketingový výzkum, jehož realizace probíhala v období od 1.2.2013 do 31.3.2013 v Českých Budějovicích. Jednalo se o empirický výzkum provedený metodou pozorování, konkrétně nezúčastněným pozorováním skrytým. Údaje získané marketingovým výzkumem byly zpracované pomocí programu Microsoft Excel.



### Kalibrace modelů pro případové studie

Informace získané marketingovým výzkumem byly následně využity při procesu kalibrace simulačních modelů pro případové studie. Byly vytvořeny unikátní layouty (obrázky dispozičních řešení daných maloobchodních jednotek) a tyto obrázky byly do modelů vloženy. Dále byly na základě dat z marketingového výzkumu provedeny v modelech další úpravy. Jednalo se o:

- nastavení návštěvnosti obchodních jednotek,
- nastavení nákupního chování zákazníků,
- eliminaci prvku nákupních vozíků,
- úpravu rychlosti pohybu zákazníků po prodejní ploše,
- úpravu rychlosti odbavení zákazníků na pokladnách,
- úpravu nastavení doplňování zboží do regálů,
- úpravu režijních nákladů maloobchodních jednotek,
- úpravu prodejní marže.

V závislosti na změnách v programovém kódu modelů bylo dále upraveno uživatelské prostředí modelů.

### Rozšíření modelu

Během kalibrace byly výsledné modely rozšířeny o několik nových prvků. Oproti výchozímu simulačnímu modelu byla uživateli přidána možnost volby třech nových parametrů. Těmito parametry jsou:

- nastavení podílu amortizace na dlouhodobém majetku,
- nastavení výše mezd pro pracovníky doplňující zboží do regálů,
- nastavení nákupní ceny zboží pro obchodní jednotku.

Výsledkem procesu kalibrace byly tři konkrétní simulační modely pro tři případové studie.

### Metodika tvorby případových studií

Následně byla vytvořena obecná metodika tvorby a použití případových studií s využitím výchozího multiagentního simulačního modelu. Tato metodika byla ověřena pěti scénáři, kterými byla testována funkčnost vytvořených modelů případových studií.

### Scénáře

První scénář ověřil obecnou funkčnost výsledných simulačních modelů při středních hodnotách nastavení volitelných parametrů.

Druhý, třetí a čtvrtý scénář ověřil vliv nově přidaných prvků popsaných výše.

Pátý scénář sledoval pohyb zákazníků při zvýšeném zatížení obchodní jednotky.

### Návrhy

Multiagentní simulační model samoobslužné maloobchodní jednotky má veliký potenciál pro budoucí rozvoj, proto byly v další části práce sepsány návrhy na další možnou úpravu a vylepšení modelu. Jednalo se o tyto návrhy:

- úprava doplňování zboží do regálů na průběžné,
- vložení možnosti nastavení počtu otevřených pokladen pro jednotlivé hodiny před spuštěním simulace,
- rozšíření volby marže, nákupní ceny a prodejní ceny,
- implementace indexu spokojenosti zaměstnanců,
- implementace indexu spokojenosti zákazníků,
- vložení možnosti opakování simulací s vlivem navrhovaných indexů.

## 12 Souhrn

V této diplomové práci je shrnut teoretický základ o maloobchodu a simulačních modelech využitelných jako podpora rozhodování při řízení maloobchodních jednotek. Jsou zde popsány specifika maloobchodu a specifika maloobchodních jednotek z pohledu jejich členění a základní teoretická východiska pro tvorbu simulačních modelů. Dále je v práci popsán výchozí multiagentní simulační model vytvořený vedoucím této práce, panem Ing. Viktorem Vojtkem, Ph.D., na kterém tato diplomová práce staví. V rámci této práce je popsána tvorba případových studií využívající multiagentní simulační model chodu maloobchodní jednotky, a to včetně procesu kalibrace modelů pro tyto případové studie. Na základě tvorby případových studií je dále vypracována obecná metodika tvorby takovýchto případových studií. Vytvořená metodika je následně ověřena scénáři a v poslední části práce jsou popsány návrhy na další úpravu simulačního modelu.

## Summary

In this work are summarized the theoretical basis of retail and simulation models usable as decision making support in the management of retail units. There are described the specifics of retail and specifics of retail units in terms of their classification and basic theoretical foundations for the creation of simulation models. The work also describes the default multi-agent simulation model created by the leader of this work, Ing. Viktor Vojtko, Ph.D., on which this work builds. Then work describes creation of case studies using multi-agent simulation model, including the calibration process of models for these case studies. General methodology of creating case studies is described in next part of the work. Then the created methodology is varified by scenarios and the last part describes proposals for further editing of simulation model.

## 13 Seznam použitých zdrojů

- [1] BURSTINER, I. [přeložila Lidmila Janečková]: *Základy maloobchodního podnikání*, Praha : Victoria Publishing, 1994, ISBN 80-85605-55-4
- [2] CAHLÍK, T. aj.: *Multiagentní přístupy v ekonomii*, 1. vyd., Praha : Karolinum, 2006, ISBN 80-246-1223-2
- [3] CIMLER, P.: *Retail management : Lokalizace a provoz maloobchodu*, 1. vyd., Praha : Vysoká škola ekonomická, 1998, ISBN 80-7079-596-4
- [4] CIMLER, P.; ZADRAŽILOVÁ, D. aj.: *Retail management*, 1. vyd., Praha : Management Press, 2007, ISBN 978-80-7261-167-6
- [5] HES, A. aj.: *Obchodní nauka*, 4. vyd., Praha : Česká zemědělská univerzita, 2002, ISBN 80-213-0751-X
- [6] HES, A.: *Velkoobchod a maloobchod*, 1. vyd., Praha : Česká zemědělská univerzita, 2004, ISBN 80-213-1163-0
- [7] HESKOVÁ, M.; PÍCHA, K.: *Vybrané problémy obchodního provozu*, 1. vyd., České Budějovice : Jihočeská univerzita v Českých Budějovicích, 2004, ISBN 80-7040-735-2
- [8] HRUSCHKA, H.: *Neural nets and genetic algorithms in Marketing*. In WIERENGA, B. (Ed.): *Handbook of Marketing decision models*, New York : Springer Science + Business Media, 2008, ISBN 978-0-387-78212-6
- [9] JINDRA, J.: *Obchodní firmy : Mezinárodní retailing*, 1. vyd., Praha : Vysoká škola ekonomická, 1996, ISBN 80-7079-918-8
- [10] LEVY, M.; WEITZ, B.: *Retail management*, 7th ed., Boston, MA : McGraw-Hill / Irwin, 2009, ISBN 978-0-07128424-0
- [11] MAINZOVÁ, E.: *Řízení obchodních činností*, 1. vyd., Plzeň : Západočeská univerzita, 2001, ISBN 80-7082-721-1

- [12] NOVÁČEK, M.; KVAPIL, I.; SUCHÁNEK, P.: *Obchodní podnikání*, 1. vyd., Brno : Masarykova univerzita, 1997, ISBN 80-210-1634-5
- [13] PEŠTOVÁ, S.; BIŇOVEC, K.: *Ekonomika a provoz obchodu*, 1. vyd., Praha : Fortuna, soukromé pedagogické nakladatelství, 1995, ISBN 80-7168-191-1
- [14] PRAŽSKÁ, L.; JINDRA, J.: *Obchodní podnikání : Retail management*, 1. vyd., Praha : Management Press, 1997, ISBN 80-85943-48-4
- [15] PRAŽSKÁ, L.; JINDRA, J.: *Obchodní podnikání : Retail management*, 2. přepracované vyd., Praha : Management Press, 2002, ISBN 80-7261-059-7

## Seznam obrázků

<i>Obrázek 1: Hlavička simulačního modelu .....</i>	34
<i>Obrázek 2: Základní obrazovka simulačního modelu .....</i>	36
<i>Obrázek 3: Layout supermarketu .....</i>	45
<i>Obrázek 4: Layout diskontu .....</i>	45
<i>Obrázek 5: Layout convenience store .....</i>	46
<i>Obrázek 6: Nastavení zobrazování layoutu v záložce Interface .....</i>	48
<i>Obrázek 7: Slide volby frekvence doplňování zboží .....</i>	55
<i>Obrázek 8: Procentuální dostupnost zboží v prodejně během simulace .....</i>	55
<i>Obrázek 9: Editace chooseru pro volbu varianty výše mezd .....</i>	58
<i>Obrázek 10: Chooser pro volbu varianty podílu amortizace na dlouhodobém majetku .....</i>	59
<i>Obrázek 11: Editace chooseru pro volbu varianty nákupní ceny zboží pro obchodní jednotku .....</i>	61

## Seznam tabulek

<i>Tabulka 1: Cyklické nástupy zaměstnanců – metoda a)</i> .....	23
<i>Tabulka 2: Cyklické nástupy zaměstnanců – metoda b)</i> .....	24
<i>Tabulka 3: Počet zákazníků ve sledovaných dnech a hodinách v supermarketu</i> .....	42
<i>Tabulka 4: Počet zákazníků ve sledovaných dnech a hodinách v diskontu</i> .....	42
<i>Tabulka 5: Počet zákazníků ve sledovaných dnech a hodinách v convenience store</i> .....	43
<i>Tabulka 6: Počet nakupovaných položek</i> .....	44
<i>Tabulka 7: Vybrané výstupy provedených simulací u scénáře č. 1 (v Kč)</i> .....	66
<i>Tabulka 8: Vybrané výstupy provedených simulací u scénáře č. 2 (v Kč)</i> .....	69
<i>Tabulka 9: Vybrané výstupy provedených simulací u scénáře č. 2 (procentické vyjádření)</i> .....	69
<i>Tabulka 10: Vybrané výstupy provedených simulací u scénáře č. 3 (v Kč)</i> .....	71
<i>Tabulka 11: Vybrané výstupy provedených simulací u scénáře č. 3 (procentické vyjádření)</i> .....	71
<i>Tabulka 12: Vybrané výstupy provedených simulací u scénáře č. 4 (v Kč)</i> .....	73
<i>Tabulka 13: Vybrané výstupy provedených simulací u scénáře č. 4 (procentické vyjádření)</i> .....	73

## Seznam schémat

<i>Schéma 1: Nastavení parametrů modelů u scénáře č. 1</i> .....	66
<i>Schéma 2: Nastavení parametrů modelů u scénáře č. 2</i> .....	68
<i>Schéma 3: Nastavení parametrů modelů u scénáře č. 3</i> .....	70
<i>Schéma 4: Nastavení parametrů modelů u scénáře č. 4</i> .....	72
<i>Schéma 5: Nastavení parametrů modelů u scénáře č. 5</i> .....	74



## Seznam grafů

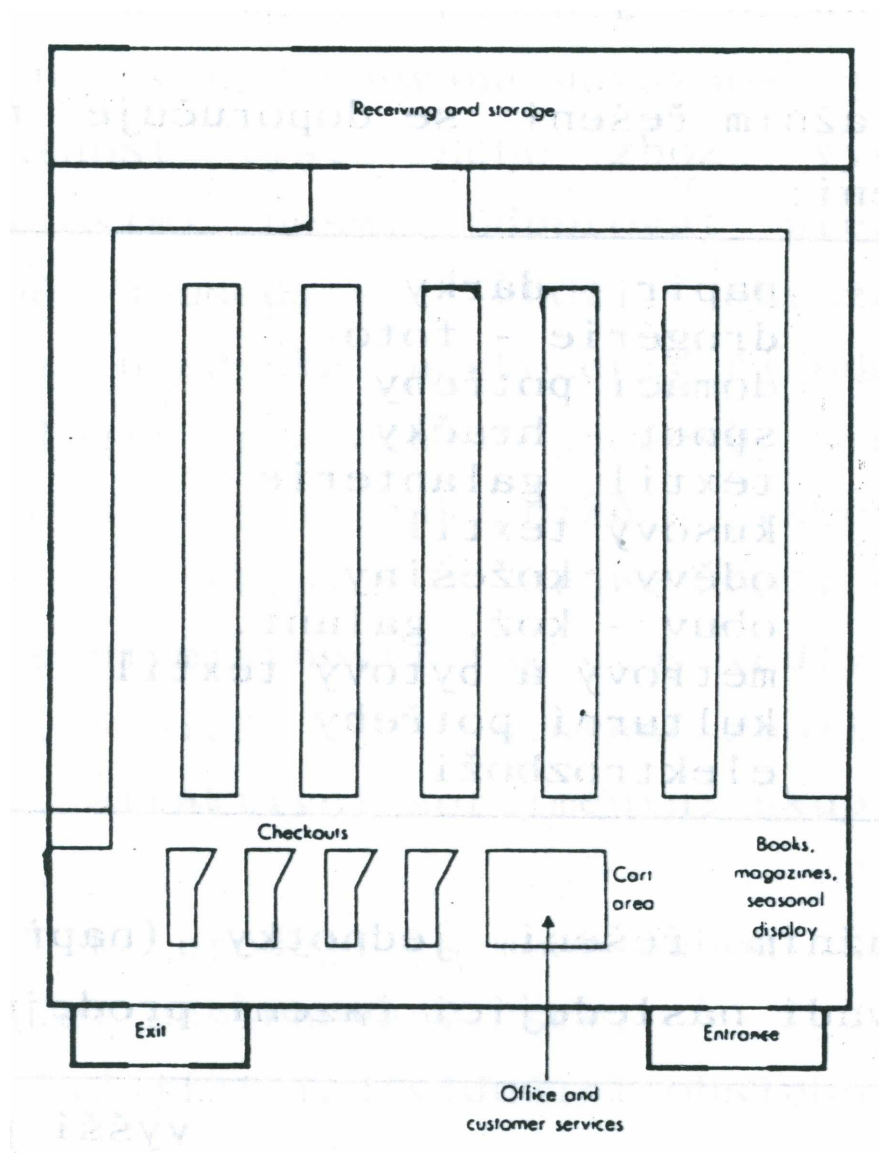
<i>Graf 1: Fronty na jednotlivých pokladnách .....</i>	<i>67</i>
<i>Graf 2: Celkový počet zákazníků čekajících ve frontách u pokladen .....</i>	<i>67</i>
<i>Graf 3: Celkový počet zákazníků čekajících ve frontách u pokladen při 1. variantě nastavení .....</i>	<i>75</i>
<i>Graf 4: Celkový počet zákazníků čekajících ve frontách u pokladen při 2. variantě nastavení .....</i>	<i>75</i>

## Seznam příloh

<i>Příloha 1: Grid layout .....</i>	91
<i>Příloha 2: Free-flow layout .....</i>	92
<i>Příloha 3: Boutique layout .....</i>	93
<i>Příloha 4: Kompletní počítačový kód původního modelu .....</i>	94
<i>Příloha 5: Kompletní počítačový kód modelu pro případovou studii Supermarket .....</i>	98
<i>Příloha 6: Kompletní počítačový kód modelu pro případovou studii Diskont .....</i>	103
<i>Příloha 7: Kompletní počítačový kód modelu pro případovou studii Convenience store .....</i>	108
<i>Příloha 8: Multiagentní simulační modely maloobchodní jednotky na přiloženém paměťovém médiu .....</i>	113

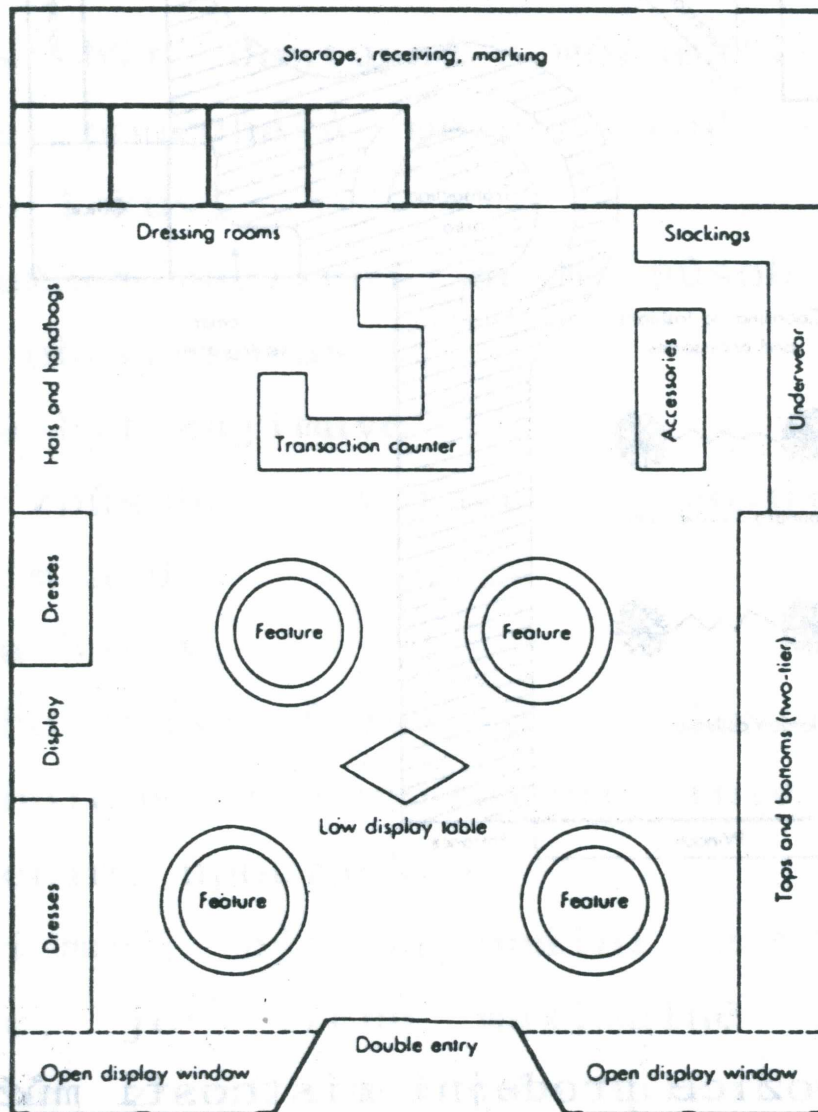
## Přílohy

*Příloha 1: Grid layout*



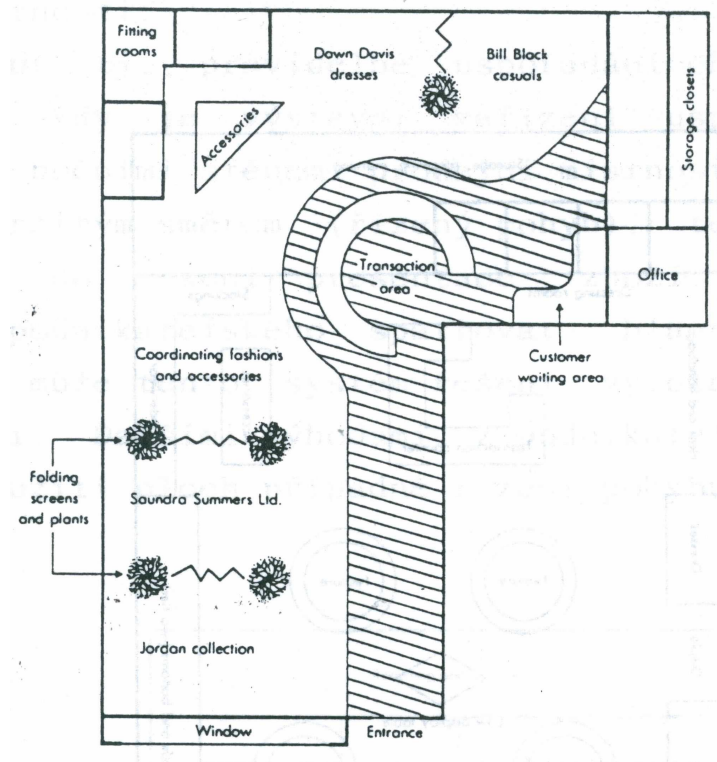
*Zdroj: Cimler, 1998, str. 90*

Příloha 2: Free-flow layout



Zdroj: Cimler, 1998, str. 91

*Příloha 3: Boutique layout*



*Zdroj: Cimler, 1998, str. 92*

## Příloha 4: Kompletní počítačový kód původního modelu

```

; 1 tick = 1 minuta
; algoritmus pro vyhledávání cesty - wavefront, šíření vlny

breed [customers customer]
breed [checkouts checkout]
breed [employees employee]

globals [
carts_capacity ; počet nákupních vozíků k dispozici
cashout_capacity ; kapacita pokladny (položek/tick)
cashout_maxtime ; maximální doba provozu pokladny
bez přestávky
break-duration ; doba trvání přestávky
average-margin ; průměrná přírážka
input_price ; vstupní cena
pcarts ; patch, na kterém bude začínat fronta vozíků
pcarts_dir ; směr zásobníku vozíků (1 - doprava, -1 - doleva)
pcarts_rows ; v kolika řadách budou vozíky umístěny
pcarts_cap ; počet vozíků na 1 patch
move_dir ; směr pohybu po prodejně (1 - levotočivě, -1 - pravotočivě)
prob_departure ; pravděpodobnost odchodu v případě, že není košík
cf ; frekvence příchodů zákazníků po minutách
hourly-arrivals ; příchody zákazníků po hodinách
purchase-size ; velikost nákupu
purchase-std-dev ; směrodatná odchylka velikosti nákupu
checkouts_to_be_open ; kolik má být otevřeno pokladen

color-floor
color-shelf1
color-shelf2
color-checkout
color-entrance-building
color-entrance-store

revenue
overheads
other-cost
]

customers-own [
amount_demanded ; definuje množství produktů, které zákazník plánuje nakoupit -> doba strávená v obchodě, hodnota nákupu
purchase-price ; aktuální cena nákupu
cart? ; má zákazník vozík?
attractors ; seznam atraktorů, generováno na základě amount_demanded
path ; seznam patches, po kterých dojde k cíli
to_checkout? ; směřuje k pokladně?
finished? ; již zákazník nakoupil?
]

checkouts-own [
queue ; kolik zákazníků je ve frontě u pokladny?
state ; stav pokladny (0 - zavřena, 1 - otevřena, 2 - končí)
oper_duration ; doba provozu
countdown_cust ; odpočet položek z aktuálního nákupu
]

employees-own [
working_hours
]

patches-own [
wavevalue ; hodnota pro algoritmus šíření vlny
goods-capacity ; maximální kapacita regálu - kolik ks zboží zde může být umístěno?
goods-price ; průměrná cena za ks v regálu
goods-amount ; aktuální množství zboží, které je k dispozici v regálu
]

to set_const
set color-floor 7.2
set color-shelf1 28.1
set color-shelf2 44.7
set color-checkout 24.9
set color-entrance-building 104.7
set color-entrance-store 62.7

set carts_capacity 400 ; kolik vozíků je k dispozici pro zákazníky?
set cashout_capacity 12 ; kolik položek za minutu v průměru pokladna vyřídí?
set input_price 10 ; kolik je vstupní cena při nákupu widgets?
set pcarts one-of patches with [ pcolor = 117.7 ] ; nastaví podle barvy počáteční patch zásobníku vozíků
set pcarts_dir -1 ; nové vozíky doleva
set pcarts_cap 10 ; počet vozíků na 1 patch
set pcarts_rows 2 ; 2 řady vozíků
set move_dir 0 ; levotočivý pohyb po prodejně
set prob_departure 0.5 ; pravděpodobnost odchodu v případě čekání na nákupní vozík
set cf (list) ; prázdný seznam příchodů zákazníků po minutách podle nastavení

if (Návštěvnost = "Slabá") [set hourly-arrivals (list 50 50 40 40 60 50 80 90 100 80 60 10)]
if (Návštěvnost = "Střední") [set hourly-arrivals (list 80 80 70 70 90 80 110 120 130 110 90 20)]
if (Návštěvnost = "Silná") [set hourly-arrivals (list 110 110 100 100 120 110 140 150 160 140 120 25)]

if (Nákupní-chování = "Malé nákupy (10 položek)") [
set purchase-size 10
set purchase-std-dev 3
]
if (Nákupní-chování = "Střední nákupy (20 položek)") [
set purchase-size 20
set purchase-std-dev 5
]
if (Nákupní-chování = "Velké nákupy (30 položek)") [
set purchase-size 30
set purchase-std-dev 8
]

set average-margin 0.5
set revenue 0
set overheads 100000
set other-cost 0
end

to setup
clear-all
import-pcolors "Layout2.png"
set_const

ask patches [
set goods-capacity 0
set goods-price 0
]

```

## Přílohy

---

```
set goods-amount 0
]

; vygeneruj pokladny
ask patches with [pcolor = color-checkout] [
  sprout-checkouts 1 [
    hide-turtle
    set state 0
    set queue (list)
  ]
]

ask patches with [pcolor = color-shelf1 or pcolor = color-shelf2] [
  set goods-capacity 5 + random 151
  set goods-price 10 + random 71
  set goods-amount goods-capacity
]

reset-ticks
end

to go
  if hour > 20 and count customers = 0 [ stop ]
  set_checkouts
  goods_replenishment
  arrivals
  shopping
  checking_out
  departures
  redraw_carts
  tick
end

to goods_replenishment
  let replenishment-times (list)
  let i 1
  repeat Frekvence-doplňování-zboží - 1 [
    set replenishment-times lput (i * round (720 / Frekvence-doplňování-zboží)) replenishment-times
    set i i + 1
  ]
  if member? ticks replenishment-times [
    ask patches with [pcolor = color-shelf1 or pcolor = color-shelf2] [
      set goods-amount goods-capacity
    ]
    set other-cost other-cost + 10000
  ]
end

to arrivals
  if minutes = 0 [ gen_CF ]
  ask one-of patches with [pcolor = color-entrance-building] [
    sprout-customers item minutes cf [
      set shape "person"
      set size 1.5
      set color green
      set amount_demanded round (random-normal purchase-size purchase-std-dev)
      if amount_demanded <= 0 [ set amount_demanded 1 ]
      set cart? false
      set to_checkout? false
      set finished? false
      set purchase-price 0

      ; nastavení atraktorů, 2 vždy ze základních oblastí, zbytek
      ; z ostatních (vybírání se sousedící místo v uličce)
      ifelse amount_demanded > 2 [
        set attractors (patch-set (n-of 2 patches with [pcolor =
        color-floor and [pcolor] of max-one-of neighbors4 [pcolor]
        = color-shelf2]) (n-of (amount_demanded - 2) patches with
        [pcolor = color-floor and [pcolor] of max-one-of neighbors4
        [pcolor] = color-shelf1]))
      ] [
        set attractors (n-of amount_demanded patches with
        [pcolor = color-floor and ([pcolor] of max-one-of
        neighbors4 [pcolor] = color-shelf2 or [pcolor] of max-one-
        of neighbors4 [pcolor] = color-shelf1))
      ]
      face min-one-of patches with [pcolor = color-entrance-
      store] [distance myself]
    ]
  ]

  ask customers with [ color = green ] [

    ifelse ([pcolor] of (patch-ahead 1) != color-entrance-store)
    [
      forward min (list 5 (distance min-one-of patches with
      [pcolor = color-entrance-store] [distance myself] - 1))
    ] [
      ifelse (carts_capacity - count customers with [cart?]) > 0 [
        set cart? true
        set color orange
        set shape "person cart"
        forward 1
        face patch-at move_dir 0
        if (is-patch? min-one-of attractors [distance myself]) [
          set path findpath patch-here (min-one-of attractors
          [mod_distance myself self])
        ]
      ] [
        if random-float 1 < probab_departure [
          set color turquoise
          set path min-one-of patches with [pcolor = color-
          entrance-building] [distance myself]
          face path
        ]
      ]
    ]
  ]

  to shopping
    ask customers with [ color = orange ] [
      ifelse (not empty? path) [
        ; 1. krok / 1 tick
        move-to first path
        set path but-first path

        ; 2. krok / 1 tick
        if (not empty? path) [
          move-to first path
          set path but-first path
        ]

        ; 3. krok / 1 tick
        if (not empty? path) [
          move-to first path
          set path but-first path
        ]

        ; 4. krok / 1 tick
        if (not empty? path) [
          move-to first path
          set path but-first path
        ]

        ; 5. krok / 1 tick
        if (not empty? path) [
          move-to first path
          set path but-first path
        ]
      ]
    ]
  ]
end
```

## Přílohy

---

```
if length path = 0 [
  ; vlož zboží do košíku a vydej se na další cestu!
]

if (not to_checkout?) [
  let purchase-price-change 0
  ask one-of neighbors4 with [pcolor = color-shelf1 or
pcolor = color-shelf2] [
    if goods-amount > 0 [
      set goods-amount goods-amount - 1
      set purchase-price-change goods-price
    ]
  ]
  set purchase-price purchase-price + purchase-price-change
]

set attractors attractors with [ self != [patch-here] of
myself ]
if (is-patch? min-one-of attractors [distance myself]) [
  set path findpath patch-here (min-one-of attractors
[mod_distance myself self])
]
] [
  ifelse (not to_checkout?) or (count checkouts-here with
[state != 1] = 1) [
    set to_checkout? false
    set path findpath patch-here choose_checkout
    if not empty? path [
      set to_checkout? true
    ]
  ] [
    set color red - 1
    ask one-of checkouts-here [ set queue (sentence ([queue]
of (one-of checkouts-here)) myself) ]
  ]
]
end

to checking_out
ask checkouts with [ state != 0 ] [
  if length queue > 0 [
    if countdown_cust = 0 [ set countdown_cust
[amount_demanded] of item 0 queue]
    set countdown_cust countdown_cust - cashout_capacity
    if countdown_cust <= 0 [
      ask item 0 queue [
        set revenue revenue + purchase-price
        set finished? true
        set color turquoise
        move-to patch-at 0 -1
        set path min-one-of patches with [pcolor = color-
entrance-building] [distance myself]
        face path
      ]
      set queue but-first queue
      set countdown_cust 0
    ]
  ]
]
end

to departures
ask customers with [ color = turquoise ] [
  ifelse ([pcolor] of (patch-ahead 1) != color-entrance-
building) [
    forward min (list 5 (distance min-one-of patches with
[pcolor = color-entrance-building] [distance myself] - 1))
  ] [
    die
  ]
]

]
end

to set_checkouts
set checkouts_to_be_open Pokladny-v-provozu
ask checkouts with [state = 2 and length queue = 0] [
  set state 0
]
if checkouts_to_be_open != count checkouts with [state =
1] [
  ifelse checkouts_to_be_open > count checkouts with [state
= 1] [
    ask n-of (checkouts_to_be_open - count checkouts with
[state = 1]) checkouts with [state = 0] [
      set state 1
    ]
  ] [
    ask n-of (count checkouts with [state = 1] -
checkouts_to_be_open) checkouts with [state = 1] [
      set state 2
    ]
  ]
]
ask checkouts [
  if state = 0 [set pcolor red]
  if state = 1 [set pcolor lime]
  if state = 2 [set pcolor orange - 1]
]
set other-cost other-cost + count checkouts with [state = 1
or state = 2] * 2.5
end

; =====
; POMOCNÉ PROCEDURY A FUNKCE
; =====

to-report hour
report 8 + floor (ticks / 60)
end

to-report minutes
report remainder ticks 60
end

to-report formathm
let hm ""

ifelse hour < 10 [
  set hm (word "0" hour ":")
] [
  set hm word hour ":"
]

ifelse minutes < 10 [
  set hm (word hm "0" minutes)
] [
  set hm word hm minutes
]

report hm
end

to gen_CF
let minute-arrivals 0
if hour < 20 [set minute-arrivals floor ((item (hour - 8)
hourly-arrivals) / 60)]
set cf (list)
repeat 60 [
  set cf sentence cf minute-arrivals
]

if hour < 20 [
  let random-minute 0
```



```

repeat item (hour - 8) hourly-arrivals - sum cf [
  set random-minute (random 60)
  set cf replace-item random-minute cf (1 + item random-
minute cf)
]
]
end

```

```

to-report genwaves [level finish2]
let found false
ask patches with [wavevalue = level] [
ask neighbors4 with [wavevalue = -1] [
if finish2 = self [
set found true
]
]
if wavevalue = -1 [
ifelse (pcolor = color-floor) [
set wavevalue level + 1
] [
set wavevalue -2
]
]
]
]
report found
end

```

```

to-report findpath [start finish]
let rpath finish
let pos start

```

```

if-else (is-patch? start and is-patch? finish) [
ask patches with [wavevalue != -1] [
set wavevalue -1
]
ask finish [
set wavevalue 0
]
let i 0
let t false
while [not t] [
set t genwaves i (start)
set i i + 1
]
while [pos != finish] [
ask pos [ set pos min-one-of neighbors4 with [wavevalue
>= 0] [wavevalue]]
set rpath sentence rpath pos
]
report but-first rpath
] [
report (list)
]
end

```

```

to-report choose_checkout
; v tomto případě si zákazník vybírá nejkratší frontu
if-else (count checkouts with [state = 1] > 0) [
report [patch-here] of min-one-of (checkouts with [state =
1]) [length queue]
] [
report nobody
]
end

```

```

to-report mod_distance [cust patch1]
let dir_weight 2
let dist 0

```

```

ask patch1 [
set dist (distance cust)
]

```

```

ask cust [
set dir_weight ifelse-value (xcor > [pxcor] of patch1)
[dir_weight] [1]
]

```

```

report dir_weight * dist
end

```

```

to-report variable-cost
report revenue / (1 + average-margin)
end

```

```

to redraw_carts
let i 0
let j 0
ask pcarts [
repeat ceiling (carts_capacity / pcarts_cap) [
if i * pcarts_cap >= carts_capacity / pcarts_rows [
set i 0
set j j - 1
]
ask patch-at (pcarts_dir * i) j [
ifelse ceiling ((carts_capacity - count customers with
[cart?]) / pcarts_cap) > i - j * ceiling (carts_capacity /
pcarts_rows / pcarts_cap) [
set pcolor 117.7
] [
set pcolor 68.8
]
]
set i i + 1
]
]
end

```

*Zdroj: Multiagentní simulační model maloobchodní jednotky*

*Příloha 5: Kompletní počítačový kód modelu pro případovou studii Supermarket*

```

; 1 tick = 1 minuta
; algoritmus pro vyhledávání cesty - wavefront, šíření vlny

breed [customers customer]
breed [checkouts checkout]
breed [employees employee]

globals [
carts_capacity ; počet nákupních vozíků k dispozici
cashout_capacity ; kapacita pokladny (položek/tick)
cashout_maxtime ; maximální doba provozu pokladny
bez přestávky
break-duration ; doba trvání přestávky
average-margin ; průměrná přírážka
input_price ; vstupní cena
pcarts ; patch, na kterém bude začínat fronta
voziků
pcarts_dir ; směr zásobníku vozíků (1 - doprava, -1 -
doleva)
pcarts_rows ; v kolika řadách budou vozíky umístěny
pcarts_cap ; počet vozíků na 1 patch
move_dir ; směr pohybu po prodejně (1 -
levotočivě, -1 - pravotočivě)
prob_departure ; pravděpodobnost odchodu v případě,
že není košík
cf ; frekvence příchodů zákazníků po minutách
hourly-arrivals ; příchody zákazníků po hodinách
purchase-size ; velikost nákupu
purchase-std-dev ; směrodatná odchylka velikostí nákupu
checkouts_to_be_open ; kolik má být otevřeno pokladen

labor-costs ; náklady na pracovníky doplňující zboží

color-floor
color-shelf1
color-shelf2
color-checkout
color-entrance-building
color-entrance-store

revenue
overheads
other-cost
]

customers-own [
amount_demanded ; definuje množství produktů, které
zákazník plánuje nakoupit -> doba strávená v obchodě,
hodnota nákupu
purchase_price ; aktuální cena nákupu
cart? ; má zákazník vozík?
attractors ; seznam atraktorů, generováno na základě
amount_demanded
path ; seznam patches, po kterých dojde k cíli
to_checkout? ; směřuje k pokladně?
finished? ; již zákazník nakoupil?
]

checkouts-own [
queue ; kolik zákazníků je ve frontě u pokladny?
state ; stav pokladny (0 - zavřena, 1 - otevřena, 2 -
končí)
oper_duration ; doba provozu
countdown_cust ; odpočet položek z aktuálního nákupu
]

employees-own [
working_hours
]

patches-own [
wavevalue ; hodnota pro algoritmus šíření vlny
goods_capacity ; maximální kapacita regálu - kolik ks zboží
zde může být umístěno?
goods_price ; průměrná cena za ks v regálu
goods_amount ; aktuální množství zboží, které je k
dispozici v regálu
]

to set_const
set color-floor 7.2
set color-shelf1 28.1
set color-shelf2 44.7
set color-checkout 24.9
set color-entrance-building 104.7
set color-entrance-store 62.7

set carts_capacity 5000 ; kolik vozíků je k dispozici
pro zákazníky?
set cashout_capacity 30 ; kolik položek za minutu v
průměru pokladna vyřídí?
set input_price 10 ; kolik je vstupní cena při
nákupu widgets?
set pcarts one-of patches with [ pcolor = 117.7 ] ; nastaví
podle barvy počáteční patch zásobníku vozíků
set pcarts_dir -1 ; nové vozíky doleva
set pcarts_cap 500 ; počet vozíků na 1 patch
set pcarts_rows 2 ; 2 řady vozíků
set move_dir 0 ; levotočivý pohyb po prodejně
set prob_departure 0.5 ; pravděpodobnost odchodu v
případě čekání na nákupní vozík
set cf (list) ; prázdný seznam příchodů
zákazníků po minutách podle nastavení

if (Návštěvnost = "Slabá") [set hourly-arrivals (list 140
144 162 144 135 162 171 189 198 171 126 86)]
if (Návštěvnost = "Střední") [set hourly-arrivals (list 155
160 180 160 150 180 190 210 220 190 140 95)]
if (Návštěvnost = "Silná") [set hourly-arrivals (list 171 176
198 176 165 198 209 231 242 209 154 105)]

if (Nákupní-chování = "Malé nákupy (13 položek)") [
set purchase-size 13
set purchase-std-dev 8
]
if (Nákupní-chování = "Střední nákupy (16 položek)") [
set purchase-size 16
set purchase-std-dev 10
]
if (Nákupní-chování = "Velké nákupy (19 položek)") [
set purchase-size 19
set purchase-std-dev 12
]

if (Nákupní-cena-zboží-pro-obchodní-jednotku = "Nízká") [
set average-margin 0.42
]
if (Nákupní-cena-zboží-pro-obchodní-jednotku = "Střední")
[
set average-margin 0.4
]
if (Nákupní-cena-zboží-pro-obchodní-jednotku =
"Vysoká") [
set average-margin 0.38
]
]

```

```

if (Podíl-amortizace-na-dlouhodobém-majetku = "Nízký
podíl (vysoké odpisy DM)") [
  set overheads 320000
]
if (Podíl-amortizace-na-dlouhodobém-majetku = "Střední
podíl (střední odpisy DM)") [
  set overheads 300000
]
if (Podíl-amortizace-na-dlouhodobém-majetku = "Vysoký
podíl (nízké odpisy DM)") [
  set overheads 280000
]

if (Výše-mezd = "Nízké") [
  set other-cost 8000
  set labor-costs 8000
]
if (Výše-mezd = "Střední") [
  set other-cost 10000
  set labor-costs 10000
]
if (Výše-mezd = "Vysoké") [
  set other-cost 12000
  set labor-costs 12000
]

set revenue 0

end

to setup
clear-all
import-pcolors "Layout supermarket.png"
set_const

ask patches [
  set goods-capacity 0
  set goods-price 0
  set goods-amount 0
]

; vygeneruj pokladny
ask patches with [pcolor = color-checkout] [
  sprout-checkouts 1 [
    hide-turtle
    set state 0
    set queue (list)
  ]
]

ask patches with [pcolor = color-shelf1 or pcolor = color-
shelf2] [
  set goods-capacity 5 + random 151
  set goods-price 10 + random 71
  set goods-amount goods-capacity
]

reset-ticks
end

to go
if hour > 20 and count customers = 0 [ stop ]
set_checkouts
goods_replenishment
arrivals
shopping
checking_out
departures
redraw_carts
tick
end

to goods_replenishment
let replenishment-times (list)
let i 1
repeat Frekvence-doplňování-zboží - 1 [
  set replenishment-times lput (i * round (720 / Frekvence-
doplňování-zboží)) replenishment-times
  set i i + 1
]
if member? ticks replenishment-times [
  ask patches with [pcolor = color-shelf1 or pcolor = color-
shelf2] [
    set goods-amount goods-capacity
  ]
  set other-cost other-cost + labor-costs
]
end

to arrivals
if minutes = 0 [ gen_CF ]
ask one-of patches with [pcolor = color-entrance-building] [
  sprout-customers item minutes cf [
    set shape "person"
    set size 1.5
    set color green
    set amount_demanded round (random-normal purchase-
size purchase-std-dev)
    if amount_demanded <= 0 [ set amount_demanded 1 ]
    set cart? false
    set to_checkout? false
    set finished? false
    set purchase-price 0

    ; nastavení atraktorů, 2 vždy ze základních oblastí, zbytek
z ostatních (vybírám se sousedící místo v uličce)
    ifelse amount_demanded > 2 [
      set attractors (patch-set (n-of 2 patches with [pcolor =
color-floor and [pcolor] of max-one-of neighbors4 [pcolor]
= color-shelf2]) (n-of (amount_demanded - 2) patches with
[pcolor = color-floor and [pcolor] of max-one-of neighbors4
[pcolor] = color-shelf1]))
    ] [
      set attractors (n-of amount_demanded patches with
[pcolor = color-floor and ([pcolor] of max-one-of
neighbors4 [pcolor] = color-shelf2 or [pcolor] of max-one-
of neighbors4 [pcolor] = color-shelf1))
    ]
    face min-one-of patches with [pcolor = color-entrance-
store] [distance myself]
  ]
]

ask customers with [ color = green ] [

  ifelse ([pcolor] of (patch-ahead 1) != color-entrance-store)
[
  forward min (list 5 (distance min-one-of patches with
[pcolor = color-entrance-store] [distance myself] - 1))
] [
  ifelse (carts_capacity - count customers with [cart?]) > 0 [
    set cart? true
    set color orange
    set shape "person cart"
    forward 1
    face patch-at move_dir 0
    if (is-patch? min-one-of attractors [distance myself]) [
      set path findpath patch-here (min-one-of attractors
[mod_distance myself self])
    ]
  ] [
    if random-float 1 < probab_departure [
      set color turquoise
    ]
  ]
]
]

```

## Přílohy

---

```
    set path min-one-of patches with [pcolor = color-entrance-building] [distance myself]
    face path
  ]
]
]
]
end
```

```
to shopping
ask customers with [ color = orange ] [
ifelse (not empty? path) [
; 1. krok / 1 tick
move-to first path
set path but-first path

; 2. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 3. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 4. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 5. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 6. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 7. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 8. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 9. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 10. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 11. krok / 1 tick
if (not empty? path) [
move-to first path
```

```
set path but-first path
]

; 12. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 13. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 14. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 15. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 16. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 17. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 18. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 19. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 20. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

if length path = 0 [
; vlož zboží do košíku a vydej se na další cestu!

if (not to_checkout?) [
let purchase-price-change 0
ask one-of neighbors4 with [pcolor = color-shelf1 or
pcolor = color-shelf2] [
if goods-amount > 0 [
set goods-amount goods-amount - 1
set purchase-price-change goods-price
]
]
set purchase-price purchase-price + purchase-price-change
]
]
```

```

    set attractors attractors with [ self != [patch-here] of
myself ]
    if (is-patch? min-one-of attractors [distance myself]) [
    set path findpath patch-here (min-one-of attractors
[mod_distance myself self])
    ]
    ]
    ] [
    ifelse (not to_checkout?) or (count checkouts-here with
[state != 1] = 1) [
    set to_checkout? false
    set path findpath patch-here choose_checkout
    if not empty? path [
    set to_checkout? true
    ]
    ] [
    set color red - 1
    ask one-of checkouts-here [ set queue (sentence ([queue]
of (one-of checkouts-here)) myself) ]
    ]
    ]
end

```

```

to checking_out
ask checkouts with [ state != 0 ] [
if length queue > 0 [
if countdown_cust = 0 [ set countdown_cust
[amount_demanded] of item 0 queue]
set countdown_cust countdown_cust - cashout_capacity
if countdown_cust <= 0 [
ask item 0 queue [
set revenue revenue + purchase-price
set finished? true
set color turquoise
move-to patch-at 0 -1
set path min-one-of patches with [pcolor = color-
entrance-building] [distance myself]
face path
]
set queue but-first queue
set countdown_cust 0
]
]
]
end

```

```

to departures
ask customers with [ color = turquoise ] [
ifelse ([pcolor] of (patch-ahead 1) != color-entrance-
building) [
forward min (list 5 (distance min-one-of patches with
[pcolor = color-entrance-building] [distance myself] - 1))
] [
die
]
]
end

```

```

to set_checkouts
set checkouts_to_be_open Pokladny-v-provozu
ask checkouts with [state = 2 and length queue = 0] [
set state 0
]
if checkouts_to_be_open != count checkouts with [state =
1] [
ifelse checkouts_to_be_open > count checkouts with [state
= 1] [
ask n-of (checkouts_to_be_open - count checkouts with
[state = 1]) checkouts with [state = 0] [
set state 1
]
]
]

```

```

] [
ask n-of (count checkouts with [state = 1] -
checkouts_to_be_open) checkouts with [state = 1] [
set state 2
]
]
]
ask checkouts [
if state = 0 [set pcolor red]
if state = 1 [set pcolor lime]
if state = 2 [set pcolor orange - 1]
]
set other-cost other-cost + count checkouts with [state = 1
or state = 2] * 2.5
end

```

```

; =====
; POMOCNÉ PROCEDURY A FUNKCE
; =====

```

```

to-report hour
report 8 + floor (ticks / 60)
end

```

```

to-report minutes
report remainder ticks 60
end

```

```

to-report formathm
let hm ""

```

```

ifelse hour < 10 [
set hm (word "0" hour ":")
] [
set hm word hour ":"
]

```

```

ifelse minutes < 10 [
set hm (word hm "0" minutes)
] [
set hm word hm minutes
]

```

```

report hm
end

```

```

to gen_CF
let minute-arrivals 0
if hour < 20 [set minute-arrivals floor ((item (hour - 8)
hourly-arrivals) / 60)]
set cf (list)
repeat 60 [
set cf sentence cf minute-arrivals
]

```

```

if hour < 20 [
let random-minute 0
repeat item (hour - 8) hourly-arrivals - sum cf [
set random-minute (random 60)
set cf replace-item random-minute cf (1 + item random-
minute cf)
]
]
]
end

```

```

to-report genwaves [level finish2]
let found false
ask patches with [wavevalue = level] [
ask neighbors4 with [wavevalue = -1] [
if finish2 = self [
set found true
]
]
if wavevalue = -1 [

```

```

ifelse (pcolor = color-floor) [
  set wavevalue level + 1
] [
  set wavevalue -2
]
]
]
]
report found
end

to-report findpath [start finish]
let rpath finish
let pos start

if-else (is-patch? start and is-patch? finish) [
  ask patches with [wavevalue != -1] [
    set wavevalue -1
  ]
  ask finish [
    set wavevalue 0
  ]
  let i 0
  let t false
  while [not t] [
    set t genwaves i (start)
    set i i + 1
  ]
  while [pos != finish] [
    ask pos [ set pos min-one-of neighbors4 with [wavevalue
    >= 0] [wavevalue]]
    set rpath sentence rpath pos
  ]
  report but-first rpath
] [
  report (list)
]
end

to-report choose_checkout
; v tomto případě si zákazník vybírá nejkratší frontu
if-else (count checkouts with [state = 1] > 0) [
  report [patch-here] of min-one-of (checkouts with [state =
  1]) [length queue]
] [
  report nobody
]
end

]
end

to-report mod_distance [cust patch1]
let dir_weight 2
let dist 0

ask patch1 [
  set dist (distance cust)
]

ask cust [
  set dir_weight ifelse-value (xcor > [pxcor] of patch1)
  [dir_weight] [1]
]

report dir_weight * dist
end

to-report variable-cost
report revenue / (1 + average-margin)
end

to redraw_carts
let i 0
let j 0
ask pcarts [
  repeat ceiling (carts_capacity / pcarts_cap) [
    if i * pcarts_cap >= carts_capacity / pcarts_rows [
      set i 0
      set j j - 1
    ]
    ask patch-at (pcarts_dir * i) j [
      ifelse ceiling ((carts_capacity - count customers with
      [cart?]) / pcarts_cap) > i - j * ceiling (carts_capacity /
      pcarts_rows / pcarts_cap) [
        set pcolor 117.7
      ] [
        set pcolor 68.8
      ]
    ]
    set i i + 1
  ]
]
end

```

*Zdroj: Multiagentní simulační model maloobchodní jednotky*

## Příloha 6: Kompletní počítačový kód modelu pro případovou studii Diskont

```

; 1 tick = 1 minuta
; algoritmus pro vyhledávání cesty - wavefront, šíření vlny

breed [customers customer]
breed [checkouts checkout]
breed [employees employee]

globals [
carts_capacity ; počet nákupních vozíků k dispozici
cashout_capacity ; kapacita pokladny (položek/tick)
cashout_maxtime ; maximální doba provozu pokladny
bez přestávky
break-duration ; doba trvání přestávky
average-margin ; průměrná přírážka
input_price ; vstupní cena
pcarts ; patch, na kterém bude začínat fronta
vozíků
pcarts_dir ; směr zásobníku vozíků (1 - doprava, -1 -
doleva)
pcarts_rows ; v kolika řadách budou vozíky umístěny
pcarts_cap ; počet vozíků na 1 patch
move_dir ; směr pohybu po prodejně (1 -
levotočivě, -1 - pravotočivě)
prob_departure ; pravděpodobnost odchodu v případě,
že není košík
cf ; frekvence příchodů zákazníků po minutách
hourly-arrivals ; příchody zákazníků po hodinách
purchase-size ; velikost nákupu
purchase-std-dev ; směrodatná odchylka velikosti nákupu
checkouts_to_be_open ; kolik má být otevřeno pokladen

labor-costs ; náklady na pracovníky doplňující zboží

color-floor
color-shelf1
color-shelf2
color-checkout
color-entrance-building
color-entrance-store

revenue
overheads
other-cost
]

customers-own [
amount_demanded ; definuje množství produktů, které
zákazník plánuje nakoupit -> doba strávená v obchodě,
hodnota nákupu
purchase_price ; aktuální cena nákupu
cart? ; má zákazník vozík?
attractors ; seznam atraktorů, generováno na základě
amount_demanded
path ; seznam patches, po kterých dojde k cíli
to_checkout? ; směřuje k pokladně?
finished? ; již zákazník nakoupil?
]

checkouts-own [
queue ; kolik zákazníků je ve frontě u pokladny?
state ; stav pokladny (0 - zavřena, 1 - otevřena, 2 -
končí)
oper_duration ; doba provozu
countdown_cust ; odpočet položek z aktuálního nákupu
]

employees-own [
working_hours
]

patches-own [
wavevalue ; hodnota pro algoritmus šíření vlny
goods_capacity ; maximální kapacita regálu - kolik ks zboží
zde může být umístěno?
goods_price ; průměrná cena za ks v regálu
goods_amount ; aktuální množství zboží, které je k
dispozici v regálu
]

to set_const
set color-floor 7.2
set color-shelf1 28.1
set color-shelf2 44.7
set color-checkout 24.9
set color-entrance-building 104.7
set color-entrance-store 62.7

set carts_capacity 5000 ; kolik vozíků je k dispozici
pro zákazníky?
set cashout_capacity 30 ; kolik položek za minutu v
průměru pokladna vyřídí?
set input_price 10 ; kolik je vstupní cena při
nákupu widgets?
set pcarts one-of patches with [ pcolor = 117.7 ] ; nastaví
podle barvy počáteční patch zásobníku vozíků
set pcarts_dir -1 ; nové vozíky doleva
set pcarts_cap 500 ; počet vozíků na 1 patch
set pcarts_rows 2 ; 2 řady vozíků
set move_dir 0 ; levotočivý pohyb po prodejně
set prob_departure 0.5 ; pravděpodobnost odchodu v
případě čekání na nákupní vozík
set cf (list) ; prázdný seznam příchodů
zákazníků po minutách podle nastavení

if (Návštěvnost = "Slabá") [set hourly-arrivals (list 131
140 153 144 171 189 198 189 207 171 135 90)]
if (Návštěvnost = "Střední") [set hourly-arrivals (list 145
155 170 160 190 210 220 210 230 190 150 100)]
if (Návštěvnost = "Silná") [set hourly-arrivals (list 160 171
187 176 209 231 242 231 253 209 165 110)]

if (Nákupní-chování = "Malé nákupy (12 položek)") [
set purchase-size 12
set purchase-std-dev 10
]
if (Nákupní-chování = "Střední nákupy (15 položek)") [
set purchase-size 15
set purchase-std-dev 13
]
if (Nákupní-chování = "Velké nákupy (18 položek)") [
set purchase-size 18
set purchase-std-dev 16
]

if (Nákupní-cena-zboží-pro-obchodní-jednotku = "Nízká") [
set average-margin 0.32
]
if (Nákupní-cena-zboží-pro-obchodní-jednotku = "Střední") [
set average-margin 0.3
]
if (Nákupní-cena-zboží-pro-obchodní-jednotku =
"Vysoká") [
set average-margin 0.28
]
]

```





## Přílohy

---

```
    set path min-one-of patches with [pcolor = color-
entrance-building] [distance myself]
    face path
  ]
]
]
]
end
```

```
to shopping
ask customers with [ color = orange ] [
ifelse (not empty? path) [
; 1. krok / 1 tick
move-to first path
set path but-first path

; 2. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 3. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 4. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 5. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 6. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 7. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 8. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 9. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 10. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 11. krok / 1 tick
if (not empty? path) [
move-to first path
```

```
set path but-first path
]

; 12. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 13. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 14. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 15. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 16. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 17. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 18. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 19. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 20. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

if length path = 0 [
; vlož zboží do košíku a vydej se na další cestu!

if (not to_checkout?) [
let purchase-price-change 0
ask one-of neighbors4 with [pcolor = color-shelf1 or
pcolor = color-shelf2] [
if goods-amount > 0 [
set goods-amount goods-amount - 1
set purchase-price-change goods-price
]
]
set purchase-price purchase-price + purchase-price-
change
]
]
```

```

    set attractors attractors with [ self != [patch-here] of
myself ]
    if (is-patch? min-one-of attractors [distance myself]) [
    set path findpath patch-here (min-one-of attractors
[mod_distance myself self])
    ]
    ] [
ifelse (not to_checkout?) or (count checkouts-here with
[state != 1] = 1) [
set to_checkout? false
set path findpath patch-here choose_checkout
if not empty? path [
set to_checkout? true
]
] [
set color red - 1
ask one-of checkouts-here [ set queue (sentence ((queue)
of (one-of checkouts-here)) myself) ]
]
]
end

to checking_out
ask checkouts with [ state != 0 ] [
if length queue > 0 [
if countdown_cust = 0 [ set countdown_cust
[amount_demanded] of item 0 queue]
set countdown_cust countdown_cust - cashout_capacity
if countdown_cust <= 0 [
ask item 0 queue [
set revenue revenue + purchase-price
set finished? true
set color turquoise
move-to patch-at 0 -1
set path min-one-of patches with [pcolor = color-
entrance-building] [distance myself]
face path
]
set queue but-first queue
set countdown_cust 0
]
]
]
end

to departures
ask customers with [ color = turquoise ] [
ifelse ([pcolor] of (patch-ahead 1) != color-entrance-
building) [
forward min (list 5 (distance min-one-of patches with
[pcolor = color-entrance-building] [distance myself] - 1))
] [
die
]
]
end

to set_checkouts
set checkouts_to_be_open Pokladny-v-provozu
ask checkouts with [state = 2 and length queue = 0] [
set state 0
]
if checkouts_to_be_open != count checkouts with [state =
1] [
ifelse checkouts_to_be_open > count checkouts with [state
= 1] [
ask n-of (checkouts_to_be_open - count checkouts with
[state = 1]) checkouts with [state = 0] [
set state 1
]
]
]

]] [
ask n-of (count checkouts with [state = 1] -
checkouts_to_be_open) checkouts with [state = 1] [
set state 2
]
]
]
ask checkouts [
if state = 0 [set pcolor red]
if state = 1 [set pcolor lime]
if state = 2 [set pcolor orange - 1]
]
set other-cost other-cost + count checkouts with [state = 1
or state = 2] * 2.5
end

;=====
; POMOCNÉ PROCEDURY A FUNKCE
;=====

to-report hour
report 8 + floor (ticks / 60)
end

to-report minutes
report remainder ticks 60
end

to-report formathm
let hm ""

ifelse hour < 10 [
set hm (word "0" hour ":")
] [
set hm word hour ":"
]

ifelse minutes < 10 [
set hm (word hm "0" minutes)
] [
set hm word hm minutes
]

report hm
end

to gen_CF
let minute-arrivals 0
if hour < 20 [set minute-arrivals floor ((item (hour - 8)
hourly-arrivals) / 60)]
set cf (list)
repeat 60 [
set cf sentence cf minute-arrivals
]

if hour < 20 [
let random-minute 0
repeat item (hour - 8) hourly-arrivals - sum cf [
set random-minute (random 60)
set cf replace-item random-minute cf (1 + item random-
minute cf)
]
]
]
end

to-report genwaves [level finish2]
let found false
ask patches with [wavevalue = level] [
ask neighbors4 with [wavevalue = -1] [
if finish2 = self [
set found true
]
]
if wavevalue = -1 [

```

```

ifelse (pcolor = color-floor) [
  set wavevalue level + 1
] [
  set wavevalue -2
]
]
]
]
report found
end

to-report findpath [start finish]
let rpath finish
let pos start

if-else (is-patch? start and is-patch? finish) [
  ask patches with [wavevalue != -1] [
    set wavevalue -1
  ]
  ask finish [
    set wavevalue 0
  ]
  let i 0
  let t false
  while [not t] [
    set t genwaves i (start)
    set i i + 1
  ]
  while [pos != finish] [
    ask pos [ set pos min-one-of neighbors4 with [wavevalue
    >= 0] [wavevalue]]
    set rpath sentence rpath pos
  ]
  report but-first rpath
] [
  report (list)
]
end

to-report choose_checkout
; v tomto případě si zákazník vybírá nejkratší frontu
if-else (count checkouts with [state = 1] > 0) [
  report [patch-here] of min-one-of (checkouts with [state =
  1]) [length queue]
] [
  report nobody
]
end

]
end

to-report mod_distance [cust patch1]
let dir_weight 2
let dist 0

ask patch1 [
  set dist (distance cust)
]

ask cust [
  set dir_weight ifelse-value (xcor > [pxcor] of patch1)
  [dir_weight] [1]
]

report dir_weight * dist
end

to-report variable-cost
report revenue / (1 + average-margin)
end

to redraw_carts
let i 0
let j 0
ask pcarts [
  repeat ceiling (carts_capacity / pcarts_cap) [
    if i * pcarts_cap >= carts_capacity / pcarts_rows [
      set i 0
      set j j - 1
    ]
    ask patch-at (pcarts_dir * i) j [
      ifelse ceiling ((carts_capacity - count customers with
      [cart?]) / pcarts_cap) > i - j * ceiling (carts_capacity /
      pcarts_rows / pcarts_cap) [
        set pcolor 117.7
      ] [
        set pcolor 68.8
      ]
    ]
    set i i + 1
  ]
]
end

```

*Zdroj: Multiagentní simulační model maloobchodní jednotky*

## Příloha 7: Kompletní počítačový kód modelu pro případovou studii Convenience store

```

; 1 tick = 1 minuta
; algoritmus pro vyhledávání cesty - wavefront, šíření vlny

breed [customers customer]
breed [checkouts checkout]
breed [employees employee]

globals [
  carts_capacity ; počet nákupních vozíků k dispozici
  cashout_capacity ; kapacita pokladny (položek/tick)
  cashout_maxtime ; maximální doba provozu pokladny
  bez přestávky
  break-duration ; doba trvání přestávky
  average-margin ; průměrná přírážka
  input_price ; vstupní cena
  pcarts ; patch, na kterém bude začínat fronta vozíků
  pcarts_dir ; směr zásobníku vozíků (1 - doprava, -1 - doleva)
  pcarts_rows ; v kolika řadách budou vozíky umístěny
  pcarts_cap ; počet vozíků na 1 patch
  move_dir ; směr pohybu po prodejně (1 - levotočivě, -1 - pravotočivě)
  prob_departure ; pravděpodobnost odchodu v případě, že není košík
  cf ; frekvence příchodů zákazníků po minutách
  hourly-arrivals ; příchody zákazníků po hodinách
  purchase-size ; velikost nákupu
  purchase-std-dev ; směrodatná odchylka velikostí nákupů
  checkouts_to_be_open ; kolik má být otevřeno pokladen

  labor-costs ; náklady na pracovníky doplňující zboží

  color-floor
  color-shelf1
  color-shelf2
  color-checkout
  color-entrance-building
  color-entrance-store

  revenue
  overheads
  other-cost
]

customers-own [
  amount_demanded ; definuje množství produktů, které zákazník plánuje nakoupit -> doba strávená v obchodě, hodnota nákupu
  purchase_price ; aktuální cena nákupu
  cart? ; má zákazník vozík?
  attractors ; seznam atraktorů, generováno na základě amount_demanded
  path ; seznam patches, po kterých dojde k cíli
  to_checkout? ; směřuje k pokladně?
  finished? ; již zákazník nakoupil?
]

checkouts-own [
  queue ; kolik zákazníků je ve frontě u pokladny?
  state ; stav pokladny (0 - zavřena, 1 - otevřena, 2 - končí)
  oper_duration ; doba provozu
  countdown_cust ; odpočet položek z aktuálního nákupu
]

employees-own [
  working_hours
]

patches-own [
  wavevalue ; hodnota pro algoritmus šíření vlny
  goods_capacity ; maximální kapacita regálu - kolik ks zboží zde může být umístěno?
  goods_price ; průměrná cena za ks v regálu
  goods_amount ; aktuální množství zboží, které je k dispozici v regálu
]

to set_const
  set color-floor 7.2
  set color-shelf1 28.1
  set color-shelf2 44.7
  set color-checkout 24.9
  set color-entrance-building 104.7
  set color-entrance-store 62.7

  set carts_capacity 5000 ; kolik vozíků je k dispozici pro zákazníky?
  set cashout_capacity 20 ; kolik položek za minutu v průměru pokladna vyřídí?
  set input_price 10 ; kolik je vstupní cena při nákupu widgets?
  set pcarts one-of patches with [ pcolor = 117.7 ] ; nastaví podle barvy počáteční patch zásobníku vozíků
  set pcarts_dir -1 ; nové vozíky doleva
  set pcarts_cap 500 ; počet vozíků na 1 patch
  set pcarts_rows 2 ; 2 řady vozíků
  set move_dir 0 ; levotočivý pohyb po prodejně
  set prob_departure 0.5 ; pravděpodobnost odchodu v případě čekání na nákupní vozík
  set cf (list) ; prázdný seznam příchodů zákazníků po minutách podle nastavení

  if (Návštěvnost = "Slabá") [set hourly-arrivals (list 32 45 41 32 36 41 36 27 32 27 24 20)]
  if (Návštěvnost = "Střední") [set hourly-arrivals (list 35 50 45 35 40 45 40 30 35 30 27 22)]
  if (Návštěvnost = "Silná") [set hourly-arrivals (list 39 55 50 39 44 50 44 33 39 33 30 24)]

  if (Nákupní-chování = "Malé nákupy (4 položek)") [
    set purchase-size 4
    set purchase-std-dev 2
  ]
  if (Nákupní-chování = "Střední nákupy (5 položek)") [
    set purchase-size 5
    set purchase-std-dev 3
  ]
  if (Nákupní-chování = "Velké nákupy (6 položek)") [
    set purchase-size 6
    set purchase-std-dev 4
  ]

  if (Nákupní-cena-zboží-pro-obchodní-jednotku = "Nízká") [
    set average-margin 0.17
  ]
  if (Nákupní-cena-zboží-pro-obchodní-jednotku = "Střední") [
    set average-margin 0.15
  ]
  if (Nákupní-cena-zboží-pro-obchodní-jednotku = "Vysoká") [
    set average-margin 0.13
  ]
]

```

## Přílohy

---

```
if (Podíl-amortizace-na-dlouhodobém-majetku = "Nízký
podíl (vysoké odpisy DM)") [
  set overheads 3500
]
if (Podíl-amortizace-na-dlouhodobém-majetku = "Střední
podíl (střední odpisy DM)") [
  set overheads 3000
]
if (Podíl-amortizace-na-dlouhodobém-majetku = "Vysoký
podíl (nízké odpisy DM)") [
  set overheads 2500
]

if (Výše-mezd = "Nízké") [
  set other-cost 1800
  set labor-costs 1800
]
if (Výše-mezd = "Střední") [
  set other-cost 2000
  set labor-costs 2000
]
if (Výše-mezd = "Vysoké") [
  set other-cost 2200
  set labor-costs 2200
]

set revenue 0

end

to setup
clear-all
import-pcolors "Layout convenience store.png"
set_const

ask patches [
  set goods-capacity 0
  set goods-price 0
  set goods-amount 0
]

; vygeneruj pokladny
ask patches with [pcolor = color-checkout] [
  sprout-checkouts 1 [
    hide-turtle
    set state 0
    set queue (list)
  ]
]

ask patches with [pcolor = color-shelf1 or pcolor = color-
shelf2] [
  set goods-capacity 5 + random 151
  set goods-price 10 + random 71
  set goods-amount goods-capacity
]

reset-ticks
end

to go
if hour > 20 and count customers = 0 [ stop ]
set_checkouts
goods_replenishment
arrivals
shopping
checking_out
departures
redraw_carts

tick
end

to goods_replenishment
let replenishment-times (list)
let i 1
repeat Frekvence-doplňování-zboží - 1 [
  set replenishment-times lput (i * round (720 / Frekvence-
doplňování-zboží)) replenishment-times
  set i i + 1
]
if member? ticks replenishment-times [
  ask patches with [pcolor = color-shelf1 or pcolor = color-
shelf2] [
    set goods-amount goods-capacity
  ]
  set other-cost other-cost + labor-costs
]
end

to arrivals
if minutes = 0 [ gen_CF ]
ask one-of patches with [pcolor = color-entrance-building] [
  sprout-customers item minutes cf [
    set shape "person"
    set size 1.5
    set color green
    set amount_demanded round (random-normal purchase-
size purchase-std-dev)
    if amount_demanded <= 0 [ set amount_demanded 1 ]
    set cart? false
    set to_checkout? false
    set finished? false
    set purchase-price 0

; nastavení atraktorů, 2 vždy ze základních oblastí, zbytek
z ostatních (vybírám se sousedící místo v uličce)
ifelse amount_demanded > 2 [
  set attractors (patch-set (n-of 2 patches with [pcolor =
color-floor and [pcolor] of max-one-of neighbors4 [pcolor]
= color-shelf2]) (n-of (amount_demanded - 2) patches with
[pcolor = color-floor and [pcolor] of max-one-of neighbors4
[pcolor] = color-shelf1]))
][
  set attractors (n-of amount_demanded patches with
[pcolor = color-floor and ([pcolor] of max-one-of
neighbors4 [pcolor] = color-shelf2 or [pcolor] of max-one-
of neighbors4 [pcolor] = color-shelf1))
]
  face min-one-of patches with [pcolor = color-entrance-
store] [distance myself]
]
]

ask customers with [ color = green ] [

ifelse ([pcolor] of (patch-ahead 1) != color-entrance-store)
[
  forward min (list 5 (distance min-one-of patches with
[pcolor = color-entrance-store] [distance myself] - 1))
][
  ifelse (carts_capacity - count customers with [cart?]) > 0 [
    set cart? true
    set color orange
    set shape "person cart"
    forward 1
    face patch-at move_dir 0
    if (is-patch? min-one-of attractors [distance myself]) [
      set path findpath patch-here (min-one-of attractors
[mod_distance myself self])
    ]
  ][
    if random-float 1 < probab_departure [
```

```

    set color turquoise
    set path min-one-of patches with [pcolor = color-
entrance-building] [distance myself]
    face path
  ]
]
]
]
end

to shopping
ask customers with [ color = orange ] [
ifelse (not empty? path) [
; 1. krok / 1 tick
move-to first path
set path but-first path

; 2. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 3. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 4. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 5. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 6. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 7. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 8. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 9. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 10. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 11. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 12. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 13. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 14. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 15. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 16. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 17. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 18. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 19. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

; 20. krok / 1 tick
if (not empty? path) [
move-to first path
set path but-first path
]

if length path = 0 [
; vlož zboží do košíku a vydej se na další cestu!

if (not to_checkout?) [
let purchase-price-change 0
ask one-of neighbors4 with [pcolor = color-shelf1 or
pcolor = color-shelf2] [
if goods-amount > 0 [
set goods-amount goods-amount - 1
set purchase-price-change goods-price
]
]
set purchase-price purchase-price + purchase-price-
change

```

## Přílohy

---

```
]
    set attractors attractors with [ self != [patch-here] of
myself ]
    if (is-patch? min-one-of attractors [distance myself]) [
    set path findpath patch-here (min-one-of attractors
[mod_distance myself self])
    ]
    ] [
    ifelse (not to_checkout?) or (count checkouts-here with
[state != 1] = 1) [
    set to_checkout? false
    set path findpath patch-here choose_checkout
    if not empty? path [
    set to_checkout? true
    ]
    ] [
    set color red - 1
    ask one-of checkouts-here [ set queue (sentence ([queue]
of (one-of checkouts-here)) myself)]
    ]
    ]
end

to checking_out
ask checkouts with [ state != 0 ] [
if length queue > 0 [
if countdown_cust = 0 [ set countdown_cust
[amount_demanded] of item 0 queue]
set countdown_cust countdown_cust - cashout_capacity
if countdown_cust <= 0 [
ask item 0 queue [
set revenue revenue + purchase-price
set finished? true
set color turquoise
move-to patch-at 0 -1
set path min-one-of patches with [pcolor = color-
entrance-building] [distance myself]
face path
]
set queue but-first queue
set countdown_cust 0
]
]
]
end

to departures
ask customers with [ color = turquoise ] [
ifelse ([pcolor] of (patch-ahead 1) != color-entrance-
building) [
forward min (list 5 (distance min-one-of patches with
[pcolor = color-entrance-building] [distance myself] - 1))
] [
die
]
]
end

to set_checkouts
set checkouts_to_be_open Pokladny-v-provozu
ask checkouts with [state = 2 and length queue = 0] [
set state 0
]
if checkouts_to_be_open != count checkouts with [state =
1] [
ifelse checkouts_to_be_open > count checkouts with [state
= 1] [
ask n-of (checkouts_to_be_open - count checkouts with
[state = 1]) checkouts with [state = 0] [
set state 1
]
]
]
] [
ask n-of (count checkouts with [state = 1] -
checkouts_to_be_open) checkouts with [state = 1] [
set state 2
]
]
]
ask checkouts [
if state = 0 [set pcolor red]
if state = 1 [set pcolor lime]
if state = 2 [set pcolor orange - 1]
]
set other-cost other-cost + count checkouts with [state = 1
or state = 2] * 2.5
end

; =====
; POMOCNÉ PROCEDURY A FUNKCE
; =====

to-report hour
report 8 + floor (ticks / 60)
end

to-report minutes
report remainder ticks 60
end

to-report formathm
let hm ""

ifelse hour < 10 [
set hm (word "0" hour ":")
] [
set hm word hour ":"
]

ifelse minutes < 10 [
set hm (word hm "0" minutes)
] [
set hm word hm minutes
]

report hm
end

to gen_CF
let minute-arrivals 0
if hour < 20 [set minute-arrivals floor ((item (hour - 8)
hourly-arrivals) / 60)]
set cf (list)
repeat 60 [
set cf sentence cf minute-arrivals
]

if hour < 20 [
let random-minute 0
repeat item (hour - 8) hourly-arrivals - sum cf [
set random-minute (random 60)
set cf replace-item random-minute cf (1 + item random-
minute cf)
]
]
end

to-report genwaves [level finish2]
let found false
ask patches with [wavevalue = level] [
ask neighbors4 with [wavevalue = -1] [
if finish2 = self [
set found true
]
]
]
]
```

```

if wavevalue = -1 [
  ifelse (pcolor = color-floor) [
    set wavevalue level + 1
  ] [
    set wavevalue -2
  ]
]
]
]
]
report found
end

to-report findpath [start finish]
let rpath finish
let pos start

if-else (is-patch? start and is-patch? finish) [
  ask patches with [wavevalue != -1] [
    set wavevalue -1
  ]
  ask finish [
    set wavevalue 0
  ]
  let i 0
  let t false
  while [not t] [
    set t genwaves i (start)
    set i i + 1
  ]
  while [pos != finish] [
    ask pos [ set pos min-one-of neighbors4 with [wavevalue
    >= 0] [wavevalue]]
    set rpath sentence rpath pos
  ]
  report but-first rpath
] [
  report (list)
]
end

to-report choose_checkout
; v tomto případě si zákazník vybírá nejkratší frontu
if-else (count checkouts with [state = 1] > 0) [
  report [patch-here] of min-one-of (checkouts with [state =
  1]) [length queue]
] [
  report nobody
]
end

to-report mod_distance [cust patch1]
let dir_weight 2
let dist 0

ask patch1 [
  set dist (distance cust)
]

ask cust [
  set dir_weight ifelse-value (xcor > [pxcor] of patch1)
  [dir_weight] [1]
]

report dir_weight * dist
end

to-report variable-cost
report revenue / (1 + average-margin)
end

to redraw_carts
let i 0
let j 0
ask pcarts [
  repeat ceiling (carts_capacity / pcarts_cap) [
    if i * pcarts_cap >= carts_capacity / pcarts_rows [
      set i 0
      set j j - 1
    ]
    ask patch-at (pcarts_dir * i) j [
      ifelse ceiling ((carts_capacity - count customers with
      [cart?]) / pcarts_cap) > i - j * ceiling (carts_capacity /
      pcarts_rows / pcarts_cap) [
        set pcolor 117.7
      ] [
        set pcolor 68.8
      ]
    ]
    set i i + 1
  ]
]
end

```

*Zdroj: Multiagentní simulační model maloobchodní jednotky*



*Příloha 8: Multiagentní simulační modely maloobchodní jednotky na přiloženém paměťovém médiu*

Paměťové médium je vlepeno na zadních deskách práce. Obsahuje instalační soubor programu NetLogo verze 5.0.3, který je potřebný k otevření simulačních modelů. Dále médium obsahuje původní multiagentní model a tři modely vytvořené na jeho základě pro případové studie.