

Diplomová práce



České
vysoké
učení technické
v Praze

F2

Fakulta strojní
Ústav řízení a ekonomiky podniku

Simulační studie systémů skladování

Viktor Šinágl

Vedoucí: doc. Ing. Jan Horejc, Ph.D., Ing. Pavel Scholz
Studijní program: Řízení průmyslových systémů
Červenec 2022

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Šinágl** Jméno: **Viktor** Osobní číslo: **473678**
Fakulta/ústav: **Fakulta strojní**
Zadávací katedra/ústav: **Ústav řízení a ekonomiky podniku**
Studijní program: **Řízení průmyslových systémů**
Specializace: **Bez specializace**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Simulační studie systémů skladování

Název diplomové práce anglicky:

Simulation Study of Warehouse Systems

Pokyny pro vypracování:

1. Úvod – zdůvodnění zadání, cíle a úkoly práce
2. Teoretická část – logistika a skladování, simulace, simulační nástroje a popis nástroje použitého v praktické části, simulační studie a její obsah
3. Praktická část – metodika řešení, charakteristika problému a možných variant skladování, tvorba koncepčního a simulačního modelu, simulace a zhodnocení výsledků
4. Závěry a doporučení

Seznam doporučené literatury:

1. GROS, Ivan a kol. Velká kniha logistiky. Vydání: první. Praha: Vysoká škola chemicko-technologická v Praze, 2016. 507 stran. ISBN 978-80-7080-952-5
2. MANLIG, František. Využití počítačové simulace výrobních systémů. 1. vyd. Liberec: Technická univerzita v Liberci, 2014. 99 s. ISBN 978-80-7494-162-7.
3. BANKS, Jerry. Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice. Atlanta (Georgia): John Wiley & Sons, c1998, 870 s. ISBN 978-0-471-13403-9.
4. LAW, Averil M. Simulation modeling and analysis. Fifth edition. New York: McGraw-Hill Education, [2015], ©2015. xviii, 776 stran. ISBN 978-0-07-340132-4.

Jméno a pracoviště vedoucí(ho) diplomové práce:

doc. Ing. Jan Horejc, Ph.D. ústav řízení a ekonomiky podniku FS

Jméno a pracoviště druhého(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Ing. Pavel Scholz ústav řízení a ekonomiky podniku FS

Datum zadání diplomové práce: **31.03.2022** Termín odevzdání diplomové práce: **22.07.2022**

Platnost zadání diplomové práce: **29.09.2023**

doc. Ing. Jan Horejc, Ph.D.
podpis vedoucí(ho) práce

Ing. Miroslav Žilka, Ph.D.
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Michael Valášek, DrSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Chtěl bych poděkovat mému vedoucímu Pavlu Scholzovi za ochotu, trpělivost a cenné rady poskytované při výběru tématu práce i při konzultacích. Chtěl bych také srdečně poděkovat mé rodině, přátelům a blízkým za podporu nejenom během psaní této práce, ale i během celého mého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 18. července 2022

.....
Bc. Viktor Šinágl

Abstrakt

Tato práce se zabývá simulační studií skladového systému AutoStore a porovnává tento typ skladu s běžným regálovým skladem. Simulační model byl vytvořen pomocí bezplatné zkušební verze simulačního software FlexSim Express 2021.

V teoretické části je rozebrána problematika simulací, simulačních modelů a simulační studie. Je zde také popsána problematika skladové logistiky a uvedeny základní druhy běžně používaných regálových skladů i moderních robotických skladových řešení.

Praktická část popisuje tvorbou simulační studie, jsou zde uvedeny základní data, ze kterých autor vycházel při tvorbě modelů. Podrobně jsou zde vysvětleny vytvořené simulační modely pro oba typy skladů a experimenty, které byly realizovány. Výsledky experimentů jsou poté diskutovány v závěru simulační studie.

Klíčová slova: simulační studie, simulace, AutoStore, FlexSim

Vedoucí: doc. Ing. Jan Horejc, Ph.D.,
Ing. Pavel Scholz
Ústav řízení a ekonomiky podniku,
Karlovo náměstí 13,
Praha 2

Abstract

This diploma thesis focuses on a simulation study of the AutoStore warehouse system and compares this type of warehouse with a conventional rack warehouse. The simulation model was created in a free trial version of the simulation software FlexSim Express 2021.

The theoretical part discusses the issues of simulations, simulation models and simulation study. It also discusses the basic problems of warehouse logistics and lists the basic types of commonly used rack warehouses and modern robotic warehouse solutions.

The practical part focuses on simulation study. There are presented basic data which author used in the model development. The simulation model and experiments that were implemented are explained in detail in practical part of thesis. The results of the experiments are then discussed in the final section of the simulation study.

Keywords: simulation study, simulation, AutoStore, FlexSim

Title translation: Simulation Study of Warehouse Systems

Obsah

Seznam zkratk	1		
Úvod	3		
1 Simulace a její problematika	5		
1.1 Historie simulací a současný stav	5		
1.2 Dělení simulací	8		
1.3 Diskrétní simulace	9		
1.3.1 Simulační čas v diskrétní simulaci	10		
1.3.2 Významné prvky diskrétní simulace	11		
1.4 Problém generování náhodných čísel	12		
1.4.1 Lineární kongruentní generátor	13		
1.5 Spojitá simulace	14		
1.6 Hybridní simulace	15		
1.7 Objektově orientovaná simulace	15		
1.7.1 Dědičnost	16		
1.7.2 Zapouzdření	16		
1.7.3 Polymorfismus	17		
1.8 Využití simulací v oblasti výroby a skladování	17		
1.8.1 Výhody a nevýhody použití průmyslových simulací	19		
2 Nástroje pro průmyslovou simulaci	23		
2.1 Základní vlastnosti simulačního SW	23		
2.1.1 Modelovací flexibilita	23		
2.1.2 Uživatelská přívětivost	24		
2.1.3 Tvorba výstupů	24		
2.1.4 Interoperabilita	24		
2.1.5 Deployment	24		
2.1.6 Otevřenost	24		
2.2 Druhy simulačních SW	24		
2.3 Popis vybraných simulačních nástrojů	26		
2.3.1 Arena	26		
2.3.2 Tecnomatix Plant Simulation	26		
2.3.3 Visual Components	26		
2.3.4 FlexSim simulation software	27		
3 Simulační studie	29		
3.1 Vymezení projektu	29		
3.1.1 Formulace projektu	29		
3.1.2 Definování projektu	29		
3.2 Tvorba a testování modelu	29		
3.2.1 Definice struktury systémů	32		
3.2.2 Sběr a analýza dat	32		
3.2.3 Tvorba modelu	33		
3.2.4 Verifikace	33		
3.2.5 Validace	33		
3.3 Provedení experimentů	34		
3.4 Vyhodnocení projektu	35		
4 Logistika a skladování	37		
4.1 Výrobní logistika	37		
4.2 Skladová logistika	38		
4.3 Druhy skladů	40		
4.3.1 Manuálně obsluhované sklady	40		
4.3.2 Automatizované skladovací systémy	41		
Praktická část			
5 Specifikace praktické části	51		
6 Popis použitého simulačního nástroje	53		
6.1 Uživatelské rozhraní a tvorba modelu	53		
6.2 Terminologie	53		
6.3 Popis základních použitých 3D objektů	56		
6.3.1 Source	56		
6.3.2 Processor	56		
6.3.3 Queue	56		
6.3.4 Task Executor	57		
6.3.5 Operator	57		
6.3.6 Floor Storage	58		
6.4 Popis základních použitých objektů process flow	58		
6.4.1 Source	58		
6.4.2 Assign Label	59		
6.4.3 Create Tokens	59		
6.4.4 Custom Code	59		
6.4.5 Decide	59		
6.4.6 Delay	60		
6.4.7 Task Sequences: Load, Unload, Travel	60		
6.4.8 Shared Assets: List, Push to List, Pull from List	60		
6.4.9 Warehousing: Find Item, Find Slot	60		

7 Vymezení projektu	61	Závěr	107
8 Koncepční model a sběr dat	63	Literatura	109
8.1 Definice simulovaného procesu . . .	63		
8.2 Rozměry a konfigurace skladů . . .	64		
8.3 Sběr dat	65	Přílohy	
9 Tvorba modelu	69	A Obsah CD	113
9.1 Výběr simulačního software	69		
9.2 Tvorba modelu AutoStore	69		
9.2.1 3D objekty pro Autostore . . .	69		
9.2.2 Process flow pro AutoStore . .	73		
9.3 Regálový sklad	80		
9.3.1 3D objekty pro regálový sklad	80		
9.3.2 Process flow	81		
9.4 Verifikace a validace	83		
9.4.1 Verifikace	83		
9.4.2 Validace	83		
10 Experimentální část	85		
10.1 Příprava dat pro simulaci	85		
10.2 AutoStore var.1 - výška 5 boxů, náhodné rozmístění	86		
10.3 AutoStore var. 2 - výška 5 boxů, rozmístění dle obrátkovosti	88		
10.4 AutoStore var.3 - výška 10 boxů, náhodné rozmístění	89		
10.5 AutoStore var.4 - výška 10 boxů, rozmístění dle obrátkovosti	91		
10.6 Regálový sklad, var. 5 - náhodné rozmístění	93		
10.7 Regálový sklad, var. 6 - rozmístění dle obrátkovosti	94		
10.8 AutoStore sklad, var. 7 - odbavení více požadavků najednou	96		
11 Vyhodnocení výsledků simulace	97		
11.1 Porovnání setříděného a nesetříděného AutoStore skladu . .	97		
11.2 Porovnání AutoStore skladu z hlediska výšky boxů	100		
11.3 Porovnání AutoStore skladu a běžně používaného regálového skladu	103		

Obrázky

1.1	Obecný průběh simulace [2]	5	4.3	Výrobní logistika v logistickém řetězci, zdroj: [20].	39
1.2	Počítač Pegasus, na kterém běžel Torcherův a Owenův program. Zdroj: wikipedia.org	6	4.4	Víceúrovňový regálový sklad, zdroj: [22]	41
1.3	Ukázka 2D uživatelského prostředí simulačního programu Arena. Zdroj: rockwellautomation.com	7	4.5	AS/RS - hlubinné zakládání, zdroj: [24]	42
1.4	Siemens Tecnomatix Plant Simulation, moderní simulační program. Zdroj: plant-simulation.de	8	4.6	Kompaktní AS/RS a jeho princip, zdroj: [24].	43
1.5	Grafické znázornění metody přírůstku po postupných událostech, vlastní zpracování.	10	4.7	Vlevo: horizontální karuselový systém, vpravo vertikální systém, zdroj: [24].	43
1.6	Stavový diagram pro diskrétní simulační běh, vlastní zpracování, zdroj: [5].	12	4.8	Vlevo: systém AVS/R od společnosti Vanderlande, vpravo robotické vozítko, zdroj: [24]	44
1.7	LCG pro správně a špatně zadané konstanty, vlastní zpracování v MS Excel.	14	4.9	Systém iBot od firmy Opex, zdroj:[24]	44
1.8	Vlnová rovnice, příklad spojitě simulace, vlastní zpracování v Octave	15	4.10	Sam Loydovo puzzle, zdroj:[24]	45
2.1	Ukázka simulace při použití herního enginu Unity a programovacího jazyka, zdroj [11] .	25	4.11	Sam Loydova úloha pro jednoduchý sklad, zdroj:[24]	45
2.2	Ukázka programového prostředí Siemens Plant Simulation, zdroj: [13]	27	4.12	Furmansův GridFlow skladovací systém, zdroj:[24]	45
2.3	Ukázka programového prostředí Visual Components, simulace robotického svařování karoserie auta, zdroj: [14].	27	4.13	RCSR systém od společnosti AutoStore, zdroj: autostoresystem.com	46
2.4	Model skladu v softwaru FlexSim, zdroj: vlastní práce autora	28	4.14	Robotické vozítko, také označované jako kiva robot, zdroj: [24]	47
3.1	Schéma simulačního projektu	30	5.1	Diagram plánu simulačního projektu	52
3.2	Problematika tvorby modelů v grafech, zdroj: [2]	31	6.1	Záložka model	54
3.3	Fáze tvorby počítačového modelu, zdroj: [16].	33	6.2	Ukázka nástroje Process Flow a jeho objektů	55
4.1	Obecné členění logistiky, zdroj: [19]	37	6.3	Source	56
4.2	Výrobní logistika v logistickém řetězci, zdroj: [19].	38	6.4	Processor	56
			6.5	Queue	57
			6.6	Task Executors, vlevo standardní FlexSim model, vpravo upravený design pro potřeby práce.	57
			6.7	Model operátora	57
			6.8	Floor Storage naplněná boxy	58
			6.9	Jednotlivé typy objektu source	58
			6.10	Objekt Assign Label	59
			6.11	Objekt Create Tokens	59
			6.12	Custom Code	59

6.13 Objekt Decide a jeho spojení s navazujícími objekty	59	9.10 Vytvořený model regálového skladu	80
6.14 Delay	60	9.11 Sekvenční diagram pro objekty regálového skladu	82
6.15 Load, Unload a Travel	60	9.12 generování boxů a vytvoření databáze (ItemListu)	82
6.16 Objekty List, Push to List a Pull from List	60	9.13 Process flow pro odbavení požadavku	84
6.17 Find Item a Find Slot	60	9.14 Compiler console a system console a ukázka chyb vzniklých při kompilaci kódu.	84
8.1 Mapa procesu vyskladnění zboží ze skladu AutoStore v notaci Proces Flow Chart	64	10.1 Pravděpodobnostní rozdělení pro velikost objednávky a hodnoty intervalů pro generování velikosti objednávky z pseudonáhodných čísel	86
8.2 Schéma skladu typu AutoStore	65	10.2 Pravděpodobností rozdělení obrátkovosti zboží dle úrovně boxu	88
8.3 Výkres regálu, velikosti udány v metrech. Regály jsou řazeny těsně vedle sebe.	66	10.3 Pravděpodobností rozdělení obrátkovosti zboží do 5 typů a odpovídající úrovně skladu	91
8.4 Schéma regálového skladu.	66	11.1 Krabicový graf pro setříděný a nesetříděný sklad o výšce 5-ti boxů a konfiguraci třech robotů.	99
8.5 Robotické vozítko R5 od společnosti AutoStore	67	11.2 Krabicový graf pro setříděný a nesetříděný sklad o výšce 10-ti boxů a konfiguraci pěti robotů.	99
9.1 Vytvořený model AutoStore v softwaru FlexSim	70	11.3 Porovnání průměrné doby odbavení požadavku AutoStore skladu o výšce 10-ti boxů s variantami regálových skladů. ...	105
9.2 Sekvenční diagram simulačního modelu AutoStore pro vybrané objekty	71		
9.3 Vlevo: přiřazení souřadnic boxu po vstupu do FloorStorage. Vpravo: nastavení Task Executors (AutoStore vozítek)	73		
9.4 Process Flow pro generování databáze boxů	74		
9.5 Process flow vyskladnění požadavku	75		
9.6 Důležité labely přiřazené v objektu info o itemu	76		
9.7 Vpravo: objekt Create Tokens. Vlevo: tabulka SizeofOrder která slouží pro přiřazení externě vygenerovaných (v MS EXCEL) hodnot z náhodných rozdělení pro velikost požadavku a typ požadovaného boxu.	77		
9.8 Process flow zaskladnění boxu zpět do skladu	79		
9.9 Nastavení objektu resource a priorizování tokenů z subprocessu zaskladnění	80		

Tabulky

1.1 Přehled analytických nástrojů a jejich použití. vlastní zpracování, zdroj [2]	19
7.1 Soupis relevantních metrik	62
8.1 Parametry vozítka R5	67
10.1 AutoStore - varianta č.1, 2 roboti	87
10.2 AutoStore - varianta č.1, 3 roboti	87
10.3 AutoStore - varianta č.2, 2 roboti	89
10.4 AutoStore - varianta č.2, 3 roboti	89
10.5 AutoStore - varianta č.3, 3 roboti	90
10.6 AutoStore - varianta č.3, 4 roboti	90
10.7 AutoStore - varianta č.3, 5 robotů	91
10.8 AutoStore - varianta č.4, 3 roboti	92
10.9 AutoStore - varianta č.4, 4 roboti	92
10.10 AutoStore - varianta č.4, 5 robotů	93
10.11 Regálový sklad se 3-mi operátory	93
10.12 Regálový sklad se 4-mi operátory	94
10.13 Regálový sklad řazený dle obrátkovosti se 3-mi operátory.	95
10.14 Regálový sklad řazený dle obrátkovosti se 4-mi operátory.	95
10.15 Setříděný AutoStore sklad o výšce 10-ti boxů a možnosti odbavovat až dva požadavky najednou, konfigurace se 4-mi operátory.	96
11.1 Významně změněné hodnoty mezi setříděným a neseříděným skladem pro sklad o počáteční výšce 5 boxů	97
11.2 Významně změněné hodnoty mezi setříděným a neseříděným skladem pro sklad o počáteční výšce 10 boxů	98
11.3 Významně odlišné parametry pro neseříděné sklady o různých výškách	101
11.4 Významně odlišné parametry pro setříděné sklady o různých výškách	101
11.5 Porovnání plochy skladu pro Autostore o výšce 5-ti a 10-ti boxů a regálového skladu	102
11.6 Porovnání setříděného skladu s možností odbavovat více požadavků najednou a skladu z varianty 4.	102
11.7 Porovnání průměrné doby odbavení požadavku AutoStore skladu o výšce 5-ti boxů s variantami regálových skladů.	103
11.8 Porovnání variant: AutoStore skladu o výšce 10-ti boxů s variantami regálových skladů.	104



Seznam zkratek

Symbol	Význam
AGV	Automated guided vehicle
AS/RS	Automated Storage and Retrieval System
AVS/R	Autonomous Vehicle based Storage and Retrieval System
BMPN	Business Process Modeling Notation
CAD	Computer Aided Design
GPSS	General Purpose Simulation Program
LCG	Linear Congruential Generator
PLC	Programmable Logic Controller
RCSR	Robot Based Compact Storage and Retrieval Systems
RMFS	Robotic Mobile Fulfillment Systems
SQL	Structured Query Language
SW	Software
UML	Unified Modeling Language
VR	Virtual Reality



Úvod

Trend automatizace a robotizace je dnes vidět téměř ve všech oblastech podnikání. a logistika a skladování není výjimkou. Stále více a více společností vynakládá značné finanční prostředky do investic v oblasti robotických a automatických skladů, avšak pro správné investiční rozhodnutí je třeba znát přínosy a omezení těchto skladů. Získání těchto poznatků velmi usnadňuje počítačová simulace, jelikož díky ní je možné odzkoušet skladový systém ve virtuálním prostředí a dostat výsledky, které se velmi blíží reálnému provozu, přičemž náročnost a náklady na tuto simulaci nemusí být nikterak vysoké. Tuto skutečnost demonstruje tato diplomová práce, která se zabývá simulační studií robotického skladu AutoStore a byla vytvořena s použitím bezplatné zkušební verze softwaru FlexSim Express 2021. Cílem této studie bylo získat důležité poznatky a parametry AutoStore skladu, porovnat je s parametry běžného regálového skladu a na základě výsledku stanovit obecná doporučení pro využívání skladů typu AutoStore.

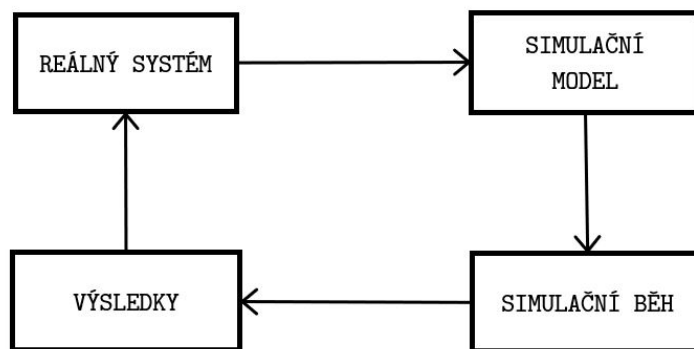
V teoretické části této diplomové práce jsou uvedeny poznatky, ze kterých autor vycházel při tvorbě simulační studie. Většina teoretické části je věnována problematice simulací, jsou zde zmíněny počátky a vývoj počítačové simulace a princip diskrétní simulace. Je zde vysvětlen obsah a charakter simulační studie a popsány softwarové nástroje, které se dají použít pro její řešení. Teoretická část se také věnuje logistice a popisuje jak běžné manuálně obsluhované sklady, tak moderní robotické skladové systémy.

Praktická část se zabývá vlastní simulační studií. Jsou zde také vysvětleny základní pojmy a znalosti pro práci se zvoleným simulačním softwarem FlexSim Express 2021. Praktická část dále popisuje důležitá data, která autor musel zjistit pro vytvoření reality blížícího se modelu, včetně zvolených rozměrů skladů a popisu skladových procesů. Je zde podrobně vysvětlen princip a fungování simulačního modelu, včetně diskuze jeho omezení a odchylek od reálného systému. V závěru praktické části jsou pak popsány provedené experimenty a diskutovány výsledky a poznatky získané z těchto experimentů, včetně návazných doporučení.

Kapitola 1

Simulace a její problematika

Simulace označuje děj, při kterém dochází k imitaci funkce určitého reálného (plánovaného) systému, procesu či objektu, a používá se jako metoda pro řešení problémů reálného světa. Základem každé simulaci je simulační model, který vzniká zobrazením reálného systému a je jeho zjednodušením, proto je simulace pouze imitací chování reálného systému. Průběh takové simulace můžeme vidět na obrázku 1.1 [1].



Obrázek 1.1: Obecný průběh simulace [2]

Simulace je důležitým nástrojem zejména pro optimalizaci. Pomocí simulace jsme schopni daný systém, proces či objekt otestovat velmi rychle a v různých podmínkách a najít vhodné řešení, které vyhovuje danému zadání.

Simulace, která probíhá za použití počítače se označuje jako počítačová simulace.

1.1 Historie simulací a současný stav

Simulace jako takové se objevují již od počátků lidské civilizace. Člověk se vytvářením modelu snažil napodobovat a uvědomovat si systémové souvislosti předtím, než je převedl do praxe. Nejdříve se objevovaly fyzické modely jako například architektonické miniatury budov nebo anatomické modely pro zaškolení lékařů. S rozvojem vědy se poté simulace posunuly do abstraktní

roviny v rámci matematických simulačních modelů. Ve 20. století, s rozvojem informačních technologií a výpočetních možností, se matematické simulační modely stávají snáze realizovatelnějšími. To má za následek velký pokrok v oblasti simulací a jejich použití se rozšiřuje do spousty průmyslových a vědeckých oblastí.

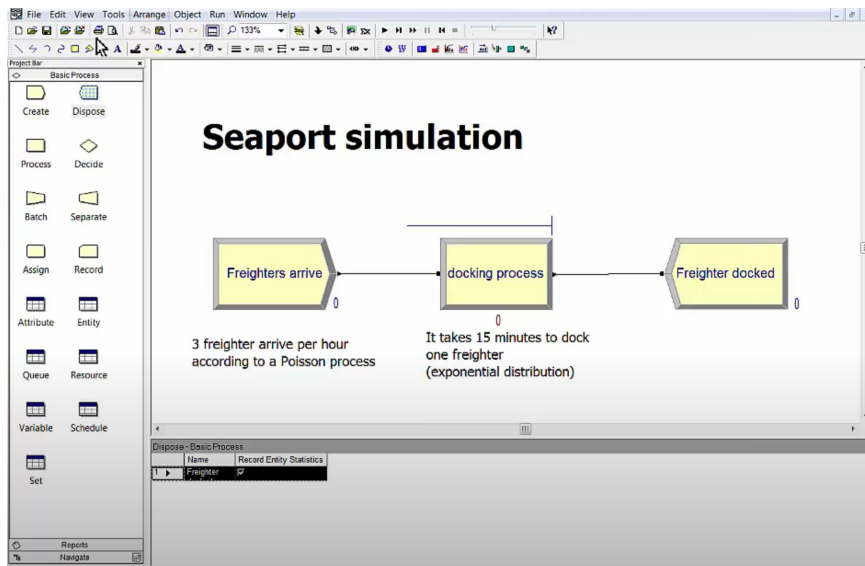
Diskrétní počítačové simulace, kterými se tato práce zabývá, mají počátek v roce 1960, kdy pánové Torcher a Owen vydali vědecký článek s názvem *The Automatic Programming of Simulations*. V tomto článku popsali algoritmus programu pro Ferrantiho počítač Pegasus (na obrázku 1.2), který se zabýval řešením různých možných situací v ocelárnách United Steel Companies. Tento program reprezentoval továrnu jako soubor strojů v různých stavech. Při změně stavu stroje (tzv. eventu) došlo k přepočítání proměnných a hodnot a zobrazení výstupů. Program tedy fungoval na principu diskrétní simulace, měl však značné výpočetní omezení, související s technologií jeho doby [3].



Obrázek 1.2: Počítač Pegasus, na kterém běžel Torcherův a Owenův program.
Zdroj: wikipedia.org

Další velký rozvoj v oblasti diskrétních simulací nastal až v roce 1981, kdy inženýr IBM Geoffrey Gordon uvedl svůj programovací jazyk GPSS (General Purpose Simulation System), který se zabýval řešením problémů diskrétní simulace, kde simulační čas postupoval po diskrétních krocích. V dalších letech pak dochází k rozvoji dalších simulačních jazyků, zejména jazyku SIMULA, což byl jeden z prvních objektově orientovaných programovacích jazyků vyvíjený Ole Johanem Dahlem a Kristenem Nygaardem na univerzitě v Oslu. Na obecné simulační jazyky pak navázali vlastní programy, které v sobě měly simulační jazyk již implementovaný a mimo to nabízeli nástroje pro tvorbu grafických výstupů (např. knihovny pro tvorbu grafů apod.). [3, 4]

V 80. letech se objevují první objektově orientované simulátory, jako např. software SIMAN/Cinema, který v sobě již obsahoval animační systém pro prezentaci výsledků. Tento program v sobě také kombinoval diskrétní a spojitou simulaci. Další velký rozvoj v rámci počítačové simulace se pak rozvíjel ruku v ruce s rozvojem informačních technologií. Začátkem 90.let se začínají objevovat první plně objektově orientované simulační softwary s uživatelským grafickým rozhraním. Uživatel tak mohl snáze vytvářet jednotlivé objekty simulace pouhým umístěním daného objektu (např. objekt reprezentující výrobní stroj) na pracovní plochu, namísto psaní vlastního kódu. Uživatel pouze měnil nastavení objektů a propojoval je mezi sebou. Samotné vytváření simulačního modelu se tak stalo přehlednějším a snadnějším, což byl velký posun v oblasti počítačových simulací. Mezi nejvýznamnější a nejpoužívanější programy této doby se řadil program Arena, který byl uveden na trh v roce 1993 [2, 3].

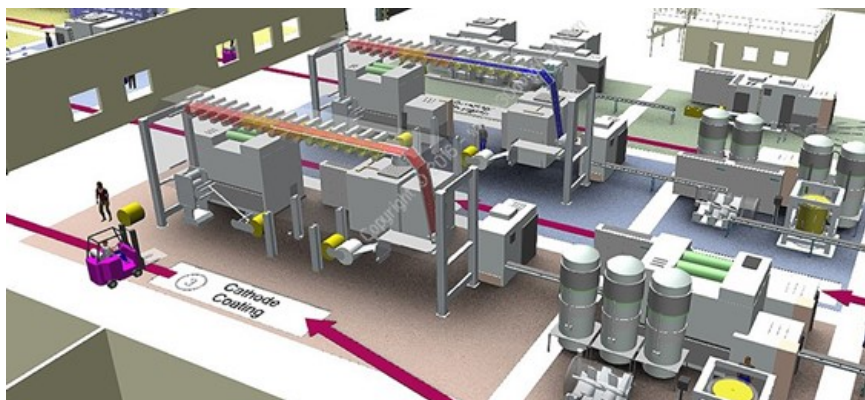


Obrázek 1.3: Ukázka 2D uživatelského prostředí simulačního programu Arena.
Zdroj: rockwellautomation.com

S dalším rozvojem výpočetní techniky se jednotlivé softwary stávají komplikovanější, na přelomu tisíciletí se objevují první 3D grafické uživatelské rozhraní a usnadňuje se práce se softwarem. Díky 3D grafickému rozhraní lze také plánovat výrobní linky i s ohledem na prostorová uspořádání, výstupy z modelu se stávají přehlednějšími. Po roce 2010 se objevují první softwary, které umožňují využití nástrojů virtuální reality [2].

Obecně se dá říci, že s rozvojem softwaru pro průmyslovou simulaci se snižuje doba tvorby modelu a nároky na uživatele. V začátku simulací byl simulační inženýr nucen vytvářet simulační model pomocí kódu a zamýšlet se nad mnoha detaily simulačního problému. To zabíralo velké množství času a zároveň požadovalo pracovníka velmi znalého problematiky. V dnešní době jsou simulační programy intuitivní a mají mnoho nástrojů, které uživateli usnadňují práci na simulačním modelu. Psaní vlastního kódu pro běžné a

jednodušší simulační problémy obvykle není nutné, jelikož jsou k dispozici objekty, které defacto vytvoří kód za uživatele. Simulace se v dnešní době stala snadno dostupným optimalizačním nástrojem pro všechny velikosti průmyslových podniků, který značně zvyšuje jejich konkurenceschopnost.



Obrázek 1.4: Siemens Tecnomatix Plant Simulation, moderní simulační program.
Zdroj: plant-simulation.de

1.2 Dělení simulací

Simulace je široký pojem, který se využívá téměř ve všech oborech, a proto autoři věnující se tomuto tématu klasifikují simulace různými způsoby, zejména podle oboru, kterému se sami věnují. Simulace tak můžeme rozdělit například na průmyslové simulace, matematické, biologické, simulace fyzikálních systémů (proudění kapalin, plynů) apod. Tato diplomová práce se ve svém textu dále věnuje převážně simulacím průmyslových procesů. Mimo dělení dle oboru lze simulace dále rozdělit dle způsobu, kterým je vytvořen simulační model. Většina simulací v dnešní době stojí na matematických modelech, kde je simulační model reprezentován matematickými funkcemi. Vzhledem k tomu, že matematický model je v podstatě základem většiny simulací, neoznačuje se v této práci simulace pouze jako „matematická simulace“, ale rozlišuje se podle dalších významných charakteristik simulačního modelu. Základní rozdělení simulací, se kterými se lze v literatuře setkat, jsou následující:

- **Fyzická simulace:** simulační model se skládá z objektů reálného světa, příkladem může být miniatura letounu ve větrném tunelu pro experimentální určení aerodynamických charakteristik [5].
- **Interaktivní simulace:** simulace, v níž je přítomen člověk, např. letecký simulátor pro trénink pilotů [1].
- **Diskrétní simulace:** simulace, kde se stavové proměnné v simulačním modelu mění skokově v nespojitých časových bodech, když nastane určitá událost. Příkladem takového jevu může být například pobočka na poště, kdy se jednotlivé přepážky obsluhují v závislosti na náhodném

příchodu zákazníků, tedy mění svůj stav (volno/obsazeno) pokaždé, když k přepážce vstoupí zákazník a nebo ji opustí [5].

- **Spojité simulace:** simulace, kde se stavové proměnné mění spojitě v čase, dle příslušné matematické funkce. Příkladem těchto simulací jsou nejčastěji simulace fyzikálních jevů, např. modely proudění kapalin apod. [1]
- **Stochastické simulace:** simulace, ve které jsou proměnné podmíněné pravděpodobnostním rozdělením s použitím pseudo-náhodných čísel. Charakteristickým rysem těchto simulací je, že v každém simulačním běhu dostaneme odlišný výsledek [1]. Diskrétní simulace jsou speciálním typem stochastické simulace.
- **Deterministická simulace:** je opakem stochastické simulace, proměnné jsou zde podmíněny deterministickým algoritmem. To znamená, že každý simulační běh který bude mít stejné počáteční podmínky, bude generovat stejný výsledek [5].
- **Hybridní simulace:** je simulace, při které dochází ke kombinaci vlastností diskretního a spojitěho modelu. Moderní počítačové softwary využívají právě tento typ simulací. Příkladem použití takové hybridní simulace může být např. modelování výroby nápojů. Výroba nápoje (míchání, zrání apod.) je spojitý proces, zatímco plnění lahví, ke kterému dochází při náhodném příchodu lahve, je diskretní děj [1].
- **Objektově orientovaná simulace:** je druhem počítačové simulace, kde je používán objektově orientovaný přístup v programování. Všechny moderní simulační programy (např. Tecnomatix Plant Simulation, FlexSim) stojí na objektově orientované simulaci [5].

Významnou kategorií simulací je pak počítačová simulace. Ta je realizovaná pomocí výpočetního stroje: počítače. Vzhledem k tomu že počítač je pouze nástrojem jak realizovat příslušnou simulaci, která je založena na matematických nebo fyzikálních vztazích, lze počítačovou simulaci dělit obdobnými způsoby, jako byly uvedeny výše (např. počítačová diskretní simulace atd.).

Jelikož diskretní a spojitá simulace (a jejich kombinace pomocí hybridní simulace) jsou základem pro většinu simulačních softwarů, budou v dalších podkapitolách vysvětleny podrobněji. Stejně tak jsou v dalších podkapitolách uvedeny druhy simulací, které jsou specifickými druhy počítačové simulace a mohou být poddruhy výše uvedených typů simulací, jako například objektově orientovaná simulace.

1.3 Diskrétní simulace

V diskretní simulaci daným modelem prochází určité modelové objekty, které mohou reprezentovat například výrobky, lidské pracovníky, zákazníky, materiál apod. Tyto objekty se označují jako **entity**. V rámci modelu existuje

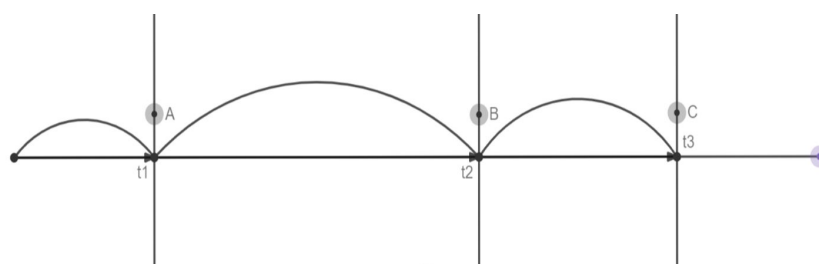
mnoho rozdílných entit a každá má své určité charakteristiky a atributy. Jednotlivé entity se mohou shlukovat do souborů, setů nebo řetězců. V průběhu simulačního děje se některé entity nachází v tzv. frontách, kde čekají na své zpracování. V diskrétní simulaci se objevují **aktivity**, které systém přesunou z jednoho stavu do druhého. Začátek a konec každé aktivity se nazývá **událostí**, přičemž stav diskrétního modelu zůstává konstantní mezi dvojicí po sobě jdoucích událostí [1, 6].

Je tedy patrné, že jednoduchá diskrétní simulace by mohla být realizována pouze s použitím kalkulačky, tužky a papíru. Princip diskrétní simulace je velmi jednoduchý. V simulacích reálných systémů je ale nutné pracovat s velkým množstvím dat a proměnných a provádět náročné výpočty, proto je nutné k diskrétní simulaci použít výpočetní techniku.

1.3.1 Simulační čas v diskrétní simulaci

V rámci simulace je nutné měřit simulační čas, abychom měli představu, jak se mění charakteristiky entit v rámci modelu v závislosti na vývoji simulace. Proměnnou, která určuje aktuální čas simulačního běhu, nazýváme **simulační hodiny** (simulation clock). Historicky se vyvinuly dva přístupy, jak počítat simulační čas: pomocí **metody stálého časového přírůstku (fixed increment time advance)** nebo metody **přírůstku času po postupných událostech (next event time advance)** [5].

Next event time advance spočívá v diskrétních časových skocích. Na začátku simulačního běhu je čas simulace nastaven na nulu a dojde k určení času výskytu budoucích událostí. Poté se simulační hodiny nastaví na hodnotu nejbližší časové události, dojde k vykonání příslušných simulačních operací, znovu se určí čas začátku budoucích simulačních událostí a dojde k nastavení simulačních hodin na další nejbližší událost. Takto se postupuje pořád dokola, dokud nenastanou předem nastavené podmínky ukončení cyklu. Časové skoky jsou pokaždé různé v závislosti na příslušných událostech, časový posun se tak děje po neregulárních časových krocích [5]. Grafické znázornění této metody je vidět na obrázku 1.5, kdy vodorovná osa reprezentuje časovou osu a oblouky představují skoky do příslušných časů t_1 , t_2 , t_3 událostí A, B, C.



Obrázek 1.5: Grafické znázornění metody přírůstku po postupných událostech, vlastní zpracování.

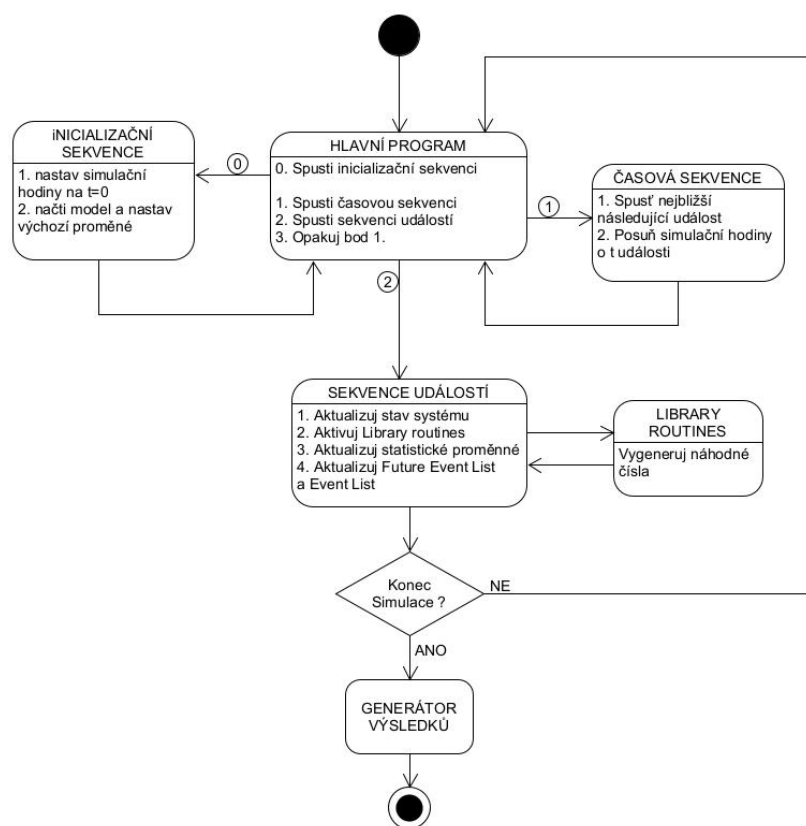
Metoda fixed increment time advance spočívá v přírůstku času po předem nastavených (většinou definovaných v rámci programu) časových přírůstcích. Časy jednotlivých událostí jsou pak elementární násobky těchto přírůstků [5].

■ 1.3.2 Významné prvky diskrétní simulace

Přestože je diskrétní simulace používána pro řešení spousty odlišných problémů, všechny diskrétní modely mají společně určité prvky. Tyto prvky jsou zobecněním logické struktury simulačních problémů. V oblasti programování pak tvoří jednotlivé třídy, objekty, metody či postupy, jak docílit požadovaného chování modelu [5]. Dle [5] jsou obecné prvky, které můžeme najít ve většině diskrétních modelech následující:

- **Stav systému (System State)** soubor systémových proměnných které charakterizují systém v daném čase.
- **Simulační hodiny (Simulation Clock)** proměnná, která udává čas simulace.
- **Seznam budoucích událostí (Future Event List)** seznam obsahující události, které ještě nenastaly, uspořádané v časovém pořadí jejich spuštění.
- **Seznam aktuálních událostí (Current Event List)** seznam událostí, které nastanou v příštím diskrétním simulačním průběhu. Událost se do seznamu aktuálních informací dostane ze seznamu budoucích událostí nebo se seznamu zpoždění (Delay List).
- **Seznam zpožděných událostí (Delay List)** seznam událostí, které byly vlivem jiných událostí zpožděny.
- **Statistické proměnné (Initialization Routine)** proměnné, které zůstávají po dobu simulačního běhu stejné (nemění se v simulačním čase).
- **Inicializační sekvence (Timing Routine)** subprogram, který spustí simulaci v čase 0.
- **Časová sekvence (Timing Routine)** subprogram, který má na starosti spouštění události ze seznamu aktuálních událostí.
- **Sekvence události (Event Routine)** spouští vlastní simulační děje v rámci události a mění hodnoty příslušných proměnných.
- **Library Routines** soubor subprogramů pro určení náhodných jevů v rámci simulačního modelu. Tyto náhodné jevy jsou zpravidla definovány příslušnými statistickými rozděleními, které modelují náhodné chování reálného systému. Jedná se zde vlastně o generování náhodných čísel.
- **Hlavní program (Main Program)** subprogram, který spouští jednotlivé sekvence (časovou, inicializační). Kontroluje také podmínky pro ukončení simulačního běhu a pokud tyto podmínky nastanou, simulaci ukončí.
- **Generátor výsledků (Report Generator)** vytváří požadované statistické výstupy ze statistických proměnných.

Logické souvislosti mezi jednotlivými prvky jsou vidět na stavovém diagramu pro diskretní simulaci na obrázku 1.6. Po spuštění softwarů se inicializuje hlavní program, což je defacto programové prostředí, které by mělo umožňovat i samotnou tvorbu modelu. V našem případě však uvažujeme pouze program pro vlastní simulační běh, kde je již model hotový. Celý simulační běh pak začíná akcí 0, kdy dojde ke spuštění inicializačního programu. Následně se aktivuje hlavní program, který spouští časovou sekvenci v kroku 1. Časová sekvence pak spouští simulační hodiny a přes hlavní program spouští příslušnou sekvenci událostí v kroku 2. Ta nastavuje lokální statistické proměnné, a pokud nedojde ke splnění koncové simulační podmínky, opět se opakují kroky 1 a 2.



Obrázek 1.6: Stavový diagram pro diskretní simulační běh, vlastní zpracování, zdroj: [5]

1.4 Problém generování náhodných čísel

Generování náhodných čísel je důležitou vlastností každé simulace a je třeba zdůraznit, že situace není tak jednoduchá, jak se na první pohled může zdát. V reálném světě můžeme náhodné číslo snadno generovat například hodem kostkou, mincí, speciálním mikro-elektronickým zařízením. U počítače je však

situace složitější, jelikož náhodná čísla jsou generována pomocí algoritmu. Pokud je ovšem náhodné číslo generováno pomocí algoritmu, nikdy nemůže být zcela náhodné, jelikož algoritmy jsou ze své podstaty deterministické. Jinými slovy: algoritmy pro generování náhodných čísel jsou funkce, jimiž se snažíme přiblížit očekávané náhodné hodnotě. Proto jsou náhodná čísla generovaná těmito algoritmy označovaná jako **pseudonáhodná** [7].

Algoritmů pro generování náhodných čísel v simulační historii vznikla celá řada. Nejznámější generátor pseudonáhodných čísel je **lineární kongruentní generátor** (*LCG – Linear Congruential Generator*) a jeho variace [7].

1.4.1 Lineární kongruentní generátor

U lineárního kongruentního generátoru se pseudonáhodná čísla generují podle následujícího vzorce:

$$x_{i+1} = (a \cdot x + b) \bmod m \quad (1.1)$$

kde a, b a m jsou přirozené konstanty, přičemž na jejich volbě závisí výsledná kvalita generátoru. *Mod* m je matematický operátor modulo. Ten udává celočíselný zbytek po dělení. Výsledek výrazu na levé straně je tedy celočíselný zbytek po dělení číslem m . Lineární kongruentní generátor má tyto charakteristiky: [8]

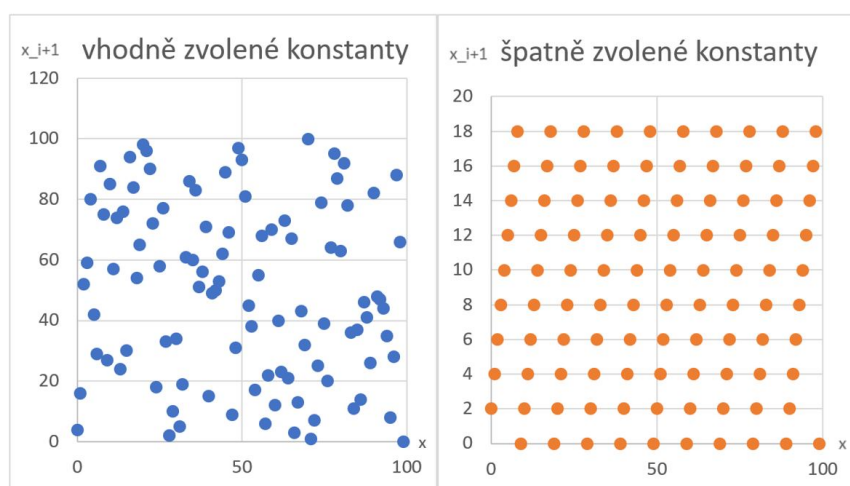
- Výsledkem je řada celých čísel o velikosti od 0 až do $m-1$.
- Výsledná řada je periodická, kdy k opakování dojde nejpozději po m vygenerovaných číslech (tzv. periodičita generátoru).
- Při nesprávně zvolených konstantách a , b a m leží vygenerované hodnoty v rovnoběžných přímkách.

Výsledky pseudonáhodného generátoru LCG při správně a nesprávně zvolených konstantách jsou vidět na obrázku 1.7. Na levém grafu bylo zvoleno: $a=3$, $b=4$, $m=101$ a na pravém grafu $a=1$, $b=2$, $m=20$. Je patrné, že čísla vygenerované pro konstanty pravého grafu se náhodnosti zcela jistě neblíží.

Pro lepší výsledky generátoru se doporučuje zvýšit periodu generátoru. Zvýšení periody se dá dosáhnout zvyšováním konstanty m (což způsobuje pro velká m snížení rychlosti generátoru), nebo spojením několika generátorů dohromady. Nejjednodušší takové spojení je prosté střídání dvou generátorů. Tato varianta zvyšuje maximální periodu na $m_1 \cdot m_2$ (pokud jsou m_1 a m_2 nesoudělná). Způsob spojení dvou lineárních kongruentních generátorů je možné vidět na následující rovnici:

$$w_i = \frac{x_i}{m_x} + \frac{y_i}{m_y} \bmod 1 \quad (1.2)$$

Operátor mod 1 vrací desetinnou část čísla daného součtem generátorů. Výsledkem jsou tedy racionální čísla v intervalu $\langle 0, 1 \rangle$ [8].



Obrázek 1.7: LCG pro správně a špatně zadané konstanty, vlastní zpracování v MS Excel

1.5 Spojitá simulace

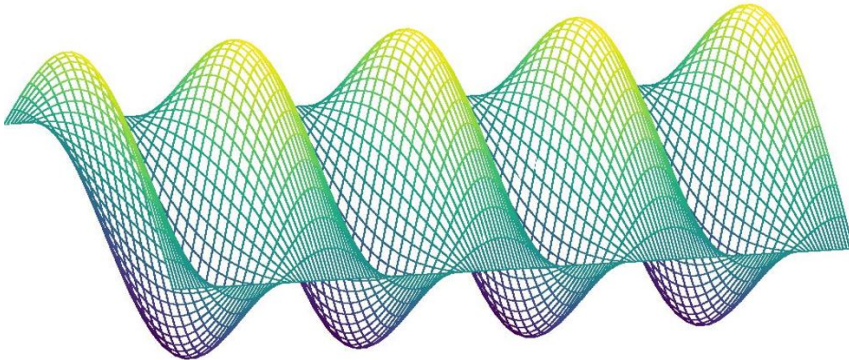
Ve spojitém simulačním modelu je stav systému reprezentován závislými proměnnými, které se spojitě mění v simulačním čase. Tyto závislé proměnné se také nazývají stavové proměnné, jelikož charakterizují stav simulovaného systému. Model spojitě simulace je pak vytvořen pomocí těchto stavových proměnných, které jsou popsány systémem matematických rovnic. Tento systémem má obvykle podobu jednoho z následujících třech typů [1]:

- systém rovnic vyjádřených explicitně: $y = f(x, t)$
- systém rovnic pro $n + 1$ člen: $y_{n+1} = ay_n + b$
- systém diferenciálních rovnic: $\frac{dy}{dt} = f(x, t)$

V simulačním systému spojitě simulace se vyskytuje nezávislá proměnná reprezentující čas, která je v rovnicích uvedených výše zastoupena písmenem t a n . Řešení simulace se získá zadáním hodnot proměnných v počátečním stavu. Tyto hodnoty se dále stávají výchozími hodnotami pro další výpočty hodnot stavových proměnných. Tento postup se označuje jako kontinuální simulační analýza a používá se v případě, že simulační model je tvořen systémem explicitních rovnic nebo rovnic pro $n+1$ člen. V případě na sobě vzájemně závislých rovnic, je simulační řešení dáno numerickým řešením soustavy rovnic v každém okamžiku simulace [1].

Názorným případem spojitě simulace jsou numerické simulace fyzikálních jevů, jako jsou například numerické metody pro výpočet proudění kapaliny, přenosu tepla apod. Stavové proměnné jsou v tomto případě zkoumané fyzikální veličiny (např. teplota) a rovnice tvoří nejčastěji soustavu diferenciálních rovnic. K realizaci těchto simulací se často používají matematické softwary jako Matlab nebo Octave. V rámci simulace průmyslových procesů jsou často

tyto simulace zastoupeny v kombinaci s diskrétní simulací a spolu tvoří tzv. hybridní nebo kombinovanou simulaci.



Obrázek 1.8: Vlnová rovnice, příklad spojitě simulace, vlastní zpracování v Octave

1.6 Hybridní simulace

Hybridní simulace je druh simulace, který v sobě kombinuje vlastnosti spojitě a diskrétní simulace. Simulační model je popsán kombinací výše uvedených charakteristik diskrétní a spojitě simulace, jako jsou entity, systémové proměnné, simulační hodiny, stavové proměnné apod. Chování modelu je charakterizováno výpočtem hodnot spojitých proměnných v krátkých časových okamžicích a výpočtem atributů jednotlivých entit v časech událostí. Mezi diskrétní simulací a spojitou se mohou objevit tři základní interakce. Za prvé, diskrétní změna hodnot proměnných může změnit hodnotu spojitě proměnné. Příkladem z oblasti průmyslové simulace může být například dokončení údržby stroje na plnění nápojů a následné okamžité zvýšení průtoku nápoje v rámci systému. Za druhé, spojitá stavová proměnná může po dosažení určitého prahu spustit diskrétní událost. Třetí možností je změna předpisu spojitě veličiny v případě výskytu specifické události. Příkladem může být například změna rychlosti (či zastavení) robota nebo AGV v případě, že se v blízkosti nachází člověk [1].

1.7 Objektově orientovaná simulace

Objektově orientovaná simulace je založena na objektově orientovaném přístupu v programování. K jednotlivým modelovým entitám přistupuje jako k objektům, které interagují mezi sebou v rámci simulace. Zástupce objektů, nesoucí podobné charakteristiky nebo vykonávající obdobnou funkci, se nazývá třída. Třída pak definuje atributy (vlastnosti) a metody (funkce), které jsou pro všechny objekty dané třídy stejné. Při programovém běhu pak konstruktor vytváří jednotlivé objekty (též nazývané instance) podle dané třídy [1].

Vlastnosti jednotlivých tříd můžeme chápat i v kontextu vlastností reálných objektů, autor této práce toto vysvětluje na následujícím teoretickém příkladu:

Uvažujme třídu `flow item`, která charakterizuje jakýkoliv hmotný objekt procházející výrobou (např. dřevěná deska). Tato deska bude mít základní atributy jako je šířka, tloušťka, délka, materiál a hmotnost. Hmotnost je pak typem vázaného atributu, která závisí na rozměrových a materiálových vlastnostech (lze ho dopočítat při znalostech jednotlivých rozměrů a hustoty materiálu). Je zřejmé, že výrobou může procházet více typů desek různých rozměrů, všechno to ovšem budou instance jednotlivých tříd, lišící se pouze různými hodnotami atributů. Obdobně se v objektově orientované simulaci vytváří třídy pro všechny entity daného modelu. Metoda je funkcí daného objektu. Uvažujme například instanci frézky, reprezentující konkrétní stroj ve výrobním podniku. Základní funkcí tohoto objektu je, že vyrábí dřevěné desky. Metodou tedy bude výroba dřevěných desek. Tato metoda při spuštění vytvoří instanci třídy `deska`. Každá metoda má parametry (argumenty funkce), které do metody vstupují. To může být v našem případě například materiál polotovaru nebo požadovaná délka. Metoda na základě těchto argumentů vytvoří dřevěnou desku dle požadované délky a výchozího materiálu.

Objektově orientovaný počítačový model je vytvářen pomocí objektově orientovaného jazyka, který umožňuje pracovat a vytvářet jednotlivé třídy a objekty (např. C++, Python, Java apod.). Mezi základní vlastnosti těchto objektů patří **dědičnost**, **polymorfismus** a **zapouzdření (encapsulation)** [1].

1.7.1 Dědičnost

Dědičnost je společnou vlastností všech tříd v objektově orientovaných jazycích. Tato vlastnost umožňuje zdědit vlastnosti dané třídy (atributy i metody) jinou, odvozenou třídou. Třída, ze které jsou ostatní třídy odvozené, se nazývá mateřská třída. Pokud budeme chtít například v simulačním softwaru vytvořit vlastní typ `flow itemu`, naše nově vytvořená třída zdědí základní atributy třídy definované v programu. V objektově orientovaných simulačních softwarech právě dědičnost značně usnadňuje vytváření modelu. Uživatel nemusí nastavovat atributy dané třídy od znovu, ale pomocí dědičnosti převezme základní atributy z mateřské třídy [1].

1.7.2 Zapouzdření

Zapouzdření umožňuje skrýt některé metody a atributy tak, aby zůstaly použitelné jen pro třídu zevnitř. Díky zapouzdření si objekt můžeme představit jako černou skříňku, které dáváme vstupní informace a ona je pak následně zpracovává. Neznáme však principy, kterými dané vstupní informace zpracovává, víme pouze to, jak se tento objekt navenek chová a jakým způsobem ho používat [1].

1.7.3 Polymorfismus

Polymorfismus je vlastností objektově orientovaných programovacích jazyků, který úzce souvisí se zapouzdřením. Polymorfismus požaduje, abychom mohli jedním a týmž způsobem komunikovat s různými objekty. Nevidíme-li do černé skříňky, nevidíme, co obsahuje a jak (pokud vůbec) se liší od jiných černých skříňek, je tedy jasné, že se všemi budeme komunikovat týmž způsobem. Naopak jestliže se všemi objekty komunikujeme stejně (polymorfně), není nutné, abychom pátrali po vnitřním obsahu. Důsledky polymorfismu pak umožňují používat jednotné rozhraní pro práci s různými typy objektů a přepsat dané metody u podtřídy tak, aby plnily náš požadovaný účel, měly stejný název a přesto neovlivnily chování metody v původní třídě [1].

Na základech objektově orientovaného přístupu dnes pracují všechny simulační softwary moderní doby, proto je základní pochopení těchto principů vhodné pro správné vytváření simulačních modelů v těchto programech. Znalosti objektově orientovaného programování pak lze využít i při vlastní tvorbě modelu, jelikož moderní programy mají své programové prostředí pro vlastní tvorbu kódu, což dovoluje simulovat náročnější a komplexnější procesy.

1.8 Využití simulací v oblasti výroby a skladování

V oblasti výroby a skladování se simulace používá pro řešení výrobních problémů a logistických problémů již mnoho let. Simulace v oblasti výroby a skladování je v podnicích používána, jelikož přináší následující pozitivní důsledky: [6, 1]

- Pomáhá zvyšovat konkurenceschopnost podniků
- Usnadňuje projektování automatizace a robotizace výrobních procesů
- Napomáhá optimalizovat a aplikovat metody štíhlé výroby a Just-In-Time.

Výhoda simulace v tomto oboru spočívá zejména ve velké komplexnosti výrobních a logistických systémů, která znemožňuje efektivní použití ostatních analytických nástrojů. Simulace je tak velmi dobrým nástrojem what-if analýzy (co se stane když, pozn.). Běžné otázky, na které simulace pomáhá najít odpověď jsou například: [1]

- Kolik lidí bude třeba zaměstnat pro včasné splnění objednávky ?
- Můžeme podnik přijmout novou zakázku, aniž by se zpomalila práce na aktuálních zakázkách ?
- Jak ověřit přínosy optimalizací a zdůvodnit jejich případnou investici a realizaci ?
- Jak nastavit pracoviště a řídit pracovníky tak, aby se snížila průměrná výrobní doba jednoho kusu ?

Simulace poskytuje metodu pro hledání odpovědní na tyto a mnohé další otázky. [1, 2]

Simulace mají v dnešní době velký význam i v rámci přechodu na průmysl 4.0. Přechod na kyberneticko-fyzikální systémy a s tím spojená automatizace a robotizace přináší pro podnik úspory ve výrobě a lepší flexibilitu, pojí se ovšem s velkými kapitálovými nároky na počáteční investici, zejména v oblasti výroby a skladování. Simulace je pak pro podniky velmi zajímavým nástrojem, jak případnou investici otestovat mimo reálný provoz s velmi malými náklady. Simulace se však uplatňuje i v (době psaní práce) aktuálním trendu digitálních dvojčat. Dnešní simulační softwary obsahují nástroje, které dokáží přenášet aktuální data z výroby do modelu. Pro řídicího pracovníka se poté stává příslušný model rychlým nástrojem what-if analýzy a umožňuje mu velmi flexibilně a pohotově řídit danou výrobu a ověřovat použití různých výrobních strategií.

Simulace dále umožňuje ve výrobních podnicích optimalizovat množstevní nároky na strojové vybavení a personál. Zkoumá různé dopadové scénáře investic a implementace nových strojů, redukci strojového parku či zaměstnanců, pomáhá analyzovat rizika a jejich dopady na výrobu v případě negativních scénářů (výpadek stroje či porucha). V době rozšíření nákazy COVID-19 simulace umožnila řešit otázky v případě onemocnění zaměstnanců ve velkém počtu, anebo naopak rychlou implementaci opatření a kvantifikaci jejich dopadu na výrobu (rozestupy apod.) Simulace také umožňuje vyřešit některé otázky související s logistikou, jako např. změna velikosti skladovacích prostor nebo optimalizace počtu manipulačních jednotek. Simulace je také nástrojem pro hodnocení výkonosti zejména v oblasti řešení úzkých míst ve výrobních systémech. Své uplatnění v rámci průmyslových systémů také hojně nachází v operativním řízení výroby, např. ověřením a optimalizací výrobních plánů, kontrolou řízení logistických systémů (dopravníky aj.) a je také efektivním nástrojem pro ověření plnění cílů výrobních strategií [1, 2].

Je ovšem nutné zdůraznit, že simulace jako taková vychází ze simulačního modelu a ten je pouze nahrazením systému reálného, nikdy není jeho věrnou kopií. Na simulaci je tak nutné nahlížet z tohoto úhlu pohledu a před samotnou tvorbu modelu pochopit co nejlépe souvislosti v simulovaném průmyslovém systému. Cíl simulačního inženýra by tak v této části měl být hlavně minimalizovat odlišnosti v reálném systému a v systému simulovaném a zároveň brát ohled na jednoduchost, přehlednost a náročnost na výpočetní čas v kontextu s dostupným výpočetním výkonem. Ideální model je tedy takový, který umožňuje přehlednou a flexibilní simulaci, která zároveň poskytuje relevantní výsledky.

Simulace nemusí být vhodným řešením v případě, že jsou dostupné jiné a levnější analytické nástroje. Průmyslové simulace pokrývají téměř většinu oblastí analytických nástrojů používaných v průmyslu. Někdy však jejich nasazení nemusí být efektivní a rychlé jako v případě použití jiných nástrojů. V tabulce 1.1 jsou vidět některé běžné problémy a nástroje, které se dají použít pro jejich řešení [2]. V následující podkapitole jsou poté shrnuty obecné výhody a nevýhody použití simulací v průmyslových systémech.

PROBLÉM	NÁSTROJE					
	Matematické programování	Síťová analýza	Teorie front	Teorie zásob	Rozmísťovací metody	Počítačová simulace
<i>Optimalizace výrobního programu</i>	X					X
<i>Optimalizace průběžné doby</i>		X	X			X
<i>Stanovení kapacit</i>	X	X	X			X
<i>Využití prostředků</i>	X		X			X
<i>Velikost zásob</i>	X		X	X		X
<i>Dispoziční uspořádání</i>	X	X	X		X	X

Tabulka 1.1: Přehled analytických nástrojů a jejich použití. vlastní zpracování, zdroj [2]

1.8.1 Výhody a nevýhody použití průmyslových simulací

Použití simulace nese vždy určitá úskalí a není samo-spásné ve všech situacích. Tak jako každý podnikový nástroj, i u simulací je třeba, aby se používal efektivně a v případech, kdy jeho použití dává pro podnik smysl. Je na vedoucích pracovnících řízení podniku, aby znali pozitiva a negativa simulací, zanalyzovali všechny faktory a vyhodnotili, zda použití průmyslové simulace dává v konkrétním případě smysl. Je nutné si uvědomit, že simulační softwary sebou přinášejí pořizovací náklady a náklady na zaškolení pracovníků. Použití softwarů může být pro nezaškolené pracovníky v začátku náročné a proto je třeba počítat i s časem, kterým nad vytváření modelu stráví. Druhou možností je pak outsorcovat simulační studie u poradenských společností, které se simulací přímo zabývají.

Jak už bylo řečeno výše, k simulaci existují alternativní analytické nástroje, a proto je třeba uvážit, kdy má simulace přínos a kdy je naopak nevhodné ji použít. **Přínosy** simulace a její **výhody** jsou shrnuty v následujících bodech: [1]

- Řešení komplexních a složitých systémů.
- Nástroj pro ověření jiných metod.
- Možnost sledovat chování systémů v zpomaleném či zrychleném čase.
- Kontrola experimentálních podmínek.
- Časová úspora při testování nových řešení (model lze po vytvoření snadno upravit a není třeba vytvářet nový).
- Opakované využití modelů.

- Náhled do systému při tvorbě modelu a lepší vhléd do systémových procesů.
- Podpora tvůrčí činnosti, podpora týmová práce.

Mezi **omezení a nevýhody** průmyslových simulací patří následující: [1, 5]

- Nároky na znalost simulačního softwaru a zkušenosti.
- Obtížná interpretace výsledků u některých úloh.
- Vysoká náročnost na čas a náklady.
- Potenciální nebezpečí chybného modelu a produkce nesprávných výsledků (pokud se neprovedla validace).
- Vyšší nároky na znalost informací o systému.
- Neoprávněné spoléhání na výsledky experimentů.
- Časová náročnost, záměna simulace za klasické analytické metody, které jsou daný problém schopny vyřešit přesněji, rychleji a s menšími náklady.

Pro správné nasazení simulací je třeba mít povědomí o určitých příkladech problémů, kdy je její použití vhodné a kdy je naopak nevhodné. Případy pro **vhodné** použití jsou uvedeny v následujících bodech: [9, 1]

- Potřeba experimentů a zkoumání interakcí uvnitř komplexních systémů.
- Zkoumání dopadů informačních a organizačních změn.
- Výrazná složitost a komplexnost systému - není vhodné jiné řešení úlohy.
- Ověření parametrů a chování nově projektovaného systému.
- Problémy vyžadující vizuální řešení či řešení pomocí virtuální reality.
- Použití pro praxi a vzdělávání.
- Realizace experimentů v praxi jen těžko proveditelných a ověřitelných.

Nevhodné situace a problémy pro řešení průmyslovou simulací jsou: [9]

- Řešení se dá najít jednodušší cestou, jednoduchými matematickými vztahy nebo obyčejným rozumem.
- Řešení je jednodušší, levnější nebo rychlejší najít ostatními běžnými analytickými nástroji.
- Řešení je jednodušší, levnější nebo rychlejší najít provedením experimentu v reálném systému.

- Náklady na simulaci převyšují úspory z ní plynoucí.
- Nedostatek pracovníků ochotných se simulací zabývat.
- Nerozumná očekávání zadavatele simulační studie (brzké termíny, přecenění výsledků).
- Nemožnost získat dostatek dat.
- Velmi komplexní systémy, které není možné řádně definovat, nebo systémy o takové složitosti, že chybí zaškolený pracovník, který by je zvládl pomocí simulace vyřešit.

Kapitola 2

Nástroje pro průmyslovou simulaci

Nástrojem pro průmyslovou simulaci v dnešní době obvykle rozumíme simulačním softwarem, jelikož průmyslová simulace se dnes řeší za použití počítače. Před začátkem simulační studie je třeba pečlivě zvážit, který dostupný nástroj použít. Pracovníci v oblasti vedení podniku by pak měli také zvážit, jak velké přínosy simulace pro daný podnik má, jaká je charakteristika simulací v podniku (jednorázová simulace nebo bude simulace využívána opakovaně?) a podle toho zvolit příslušný nástroj. Běžné simulační problémy lze řešit v jednodušších, ale levnějších softwarech, popřípadě v demo licencích. Pro řešení složitých problémů je třeba zvolit komplexní a dražší software.

Existují také softwary, které v sobě obsahují balíčky pro všeobecné řešení simulačních problémů, nejenom problémů průmyslových systémů, anebo už sama podstata softwaru umožňuje širší použití. Ve druhém případě ale může být software limitovaný v určitých funkcích, nebo je tvorba simulačního modelu komplikovanější a časově náročnější než při použití specializovaného nástroje.

2.1 Základní vlastnosti simulačního SW

Při výběru simulačního software je vhodné zaměřit se na následující vlastnosti a důsledně je vyhodnotit.

2.1.1 Modelovací flexibilita

Dle [5] je základní a nejdůležitější vlastností, kterou by měl mít každý simulační software, *modelovací flexibilita*. Modelovací flexibilita představuje schopnost namodelovat model takovým způsobem, aby jeho operační postupy mohly dosáhnout jakéhokoliv stupně komplexnosti. Mezi prvky, které ovlivňují modelovací flexibilitu, patří například schopnost využití logických a matematických funkcí, schopnost definování a změna atributů jednotlivých entit a proměnných systému k jejich využití v simulačním procesu. Dále pak možnost ukládání modelů do knihoven a jejich následná modifikace pro jiná využití [5].

■ 2.1.2 Uživatelská přívětivost

Uživatelská přívětivost je další významnou vlastností simulačního SW. Lepší uživatelskou přívětivost významně ovlivňuje grafická reprezentace programu. Dalšími faktory, které mají vliv na uživatelskou přívětivost je pak dostupnost dokumentace (manuálů, videí apod.) a jejich srozumitelnost a také komunita. Komunitou se rozumí jak komunita vlastních vývojářů, kteří jsou ochotni odpovědět na otázky ohledně SW, tak i komunita uživatelů. Mnoho problémů a návodů, jak vyřešit daný problém lze najít v komunitních fórech příslušných SW, což velmi usnadňuje práci uživatelům daného SW.

■ 2.1.3 Tvorba výstupů

Výstupy simulačního software se zobrazují především v grafické podobě. Schopnost tvorby přehledných grafů je tak významnou vlastností ovlivňující kvalitu simulačních nástrojů.

■ 2.1.4 Interoperabilita

Interoperabilita představuje možnost propojení SW s dalšími nástroji prostřednictvím komunikačních protokolů [10].

■ 2.1.5 Deployment

Deployment vyjadřuje schopnost zpracování simulačních dat. Nejběžnější formy zpracování dat jsou [10]:

1. Lokální výpočetní stanice
2. Cloud computing

■ 2.1.6 Otevřenost

Otevřenost představuje možnost přidávání dalších funkcionalit do softwaru. To se realizuje buď prostřednictvím balíčku, nebo přímou formu (přidáním scriptu do programu apod.) [10].

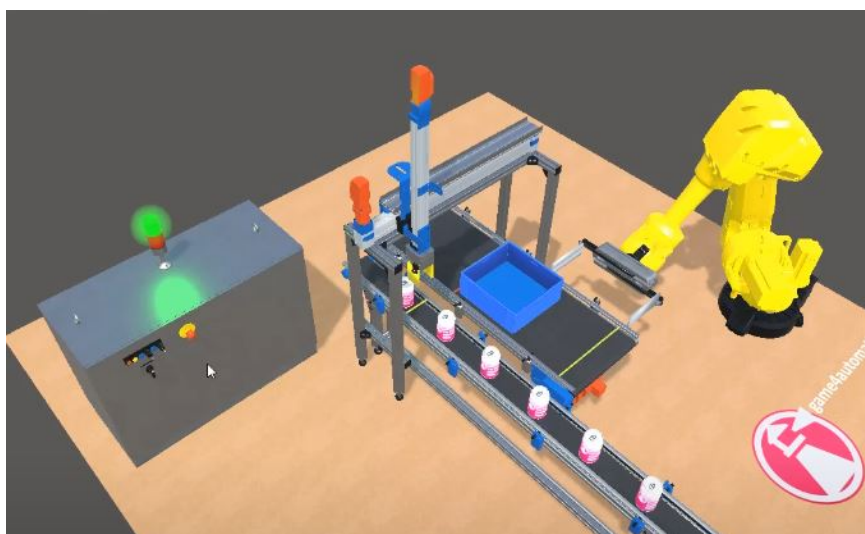
■ 2.2 Druhy simulačních SW

Pro výběr správného simulačního software je třeba ujasnit si, jaké náležitosti a výstupy od simulace čekáme. V dnešní době existují v podstatě tři obecné základní typy nástrojů počítačové simulace.

Simulace lze provádět přímo ve specializovaném simulačním software, který je obvykle zaměřen na simulaci průmyslových procesů a tvorbu digitálních tovarů. Typickými příklady použití takového specializovaného SW je například modelování výrobních linek, modelování automatických skladů apod. Tyto softwary jsou obvykle robustní, se spoustou knihoven a nástrojů, které charakterizují základní entity tvořící výrobní a logistické systémy (dopravníky, stroje

atd.). Tyto druhy simulačních SW mají obvykle dražší licenční podmínky a lze je použít převážně pro průmyslové problémy. Modelování specifických situací je zde obvykle náročnější, i když ne nemožné. Na druhou stranu jsou snazší na tvorbu simulačních modelů, jednodušší na znalosti a učení. Velkou výhodou je pak obvykle robustní uživatelská komunita, kde lze najít řešení pro širokou škálu problémů objevující se v problematice tvorby modelu. Mezi typické zástupce patří například simulační software Arena, FlexSim, Tecnomatix Plant Simulation nebo Visual Components.

Druhá možnost jak řešit simulační problém, je pomocí herních grafických prostředí (např. Unity), se specializovanými balíčky a knihovnami zaměřenými na průmyslové simulace, a běžně dostupných objektově orientovaných programovacích jazyků. Toto řešení umožňuje vytvářet velmi zajímavé a komplexní simulace zejména menších průmyslových systémů, jednotlivých procesů, robotických systémů a nebo tam, kde uživatel chce využít možnosti virtuální reality. Pořizovací náklady na tento druh SW jsou také velmi nízké. Nevýhoda tohoto řešení je, že model musí vytvářet lidé, kteří mají dobré znalosti v oblasti tvorby animací a grafiky v grafických enginech, a také dobré znalosti programování. Doba řešení u běžných typů úloh průmyslové simulace je také výrazně delší než u použití specializovaných simulačních SW [10, 11].



Obrázek 2.1: Ukázka simulace při použití herního enginu Unity a programovacího jazyka, zdroj [11]

Třetí možností je vytvoření vlastního simulačního modelu a simulačního běhu v programovacím jazyku nebo programovacím prostředí pro řešení matematických úloh (např. Matlab) nebo tabulkových procesorů. Simulace je hardwarově přívětivá, rychlá z hlediska procesní času a nenákladná. Nevýhodou této simulace je velmi složité a časově náročné simulování komplexních systémů. Nepřítomnost grafického prostředí také napomáhá k horšímu odstraňování chyb při tvorbě modelu. Mnoho chyb a špatně nastavených modelových procesů lze odhalit při samotné vizuální kontrole v programu (např. jeden objekt prochází druhým, aniž by respektoval jeho fyzikální hranice apod.),

ale odhalit je přímo z kódu nebo z výsledků simulace je obtížnější [5, 10].

■ 2.3 Popis vybraných simulačních nástrojů

V této podkapitole jsou popsány některé specializované simulační software, které jsou používány pro řešení obdobných problémů, jako je problém řešený v praktické části této práce.

■ 2.3.1 Arena

Jedná se o obecný simulační programovací jazyk integrovaný do vývojového prostředí. Tento software má grafickou a animační nadstavbu pro 3D a 2D vizualizaci modelu a zobrazení simulační animace. Model se vytváří pomocí prvků vývojových diagramů, reprezentující danou třídu nebo objekt. Pospojováním prvků (vytvářením interakcí) pak vzniká samotný simulační model. Uživatel nemusí psát vlastní kód, ten je již obsažený v objektech, které uživatel umísťuje do modelu. Podobný přístup se používá například v softwaru Matlab-Simulink [7, 12].

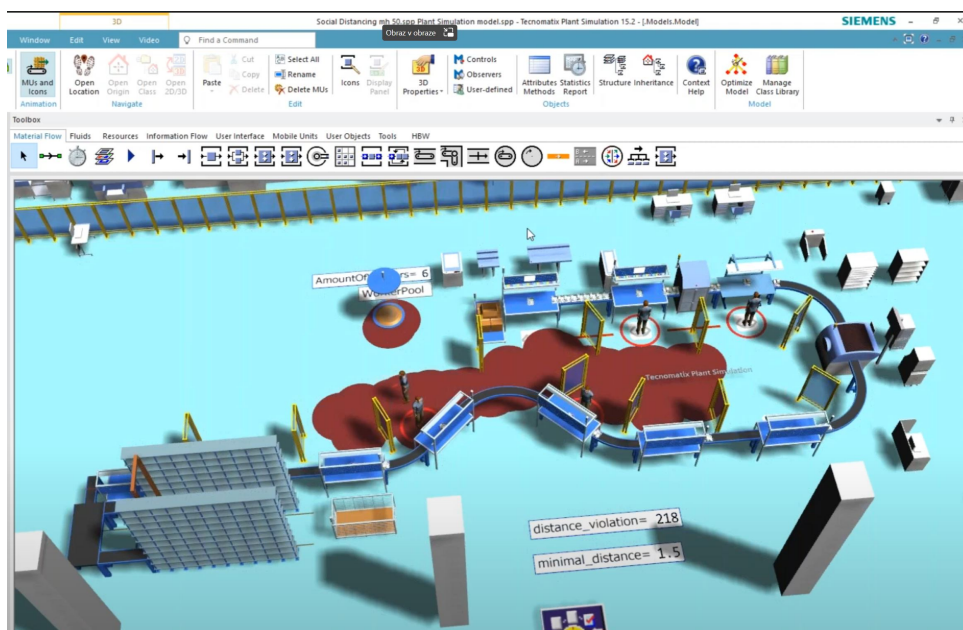
Arena je také dostupná ve zkušební verzi, která je limitovaná počtem použitých objektů.

■ 2.3.2 Tecnomatix Plant Simulation

Tecnomatix Plant Simulation je software od společnosti Siemens. Slouží k tvorbě průmyslových simulací, modelování procesů, konfigurací layoutů a digitálních dvojčat. Plant Simulation je kompatibilní s CAD systémy, nabízí možnost modelování ve 2D i 3D grafickém prostředí. Grafická podoba tohoto prostředí je zobrazena na obrázku 2.2. 3D objekty v sobě již mají zabudované základní animace, a tak animace pro běžné pohyby není nutné vytvářet. Plant Simulation umožňuje také k modelu připojit PLC a tím získat aktuální informace z provozu. Plant simulation je tak velmi dobrým nástrojem pro tvorbu digitálních dvojčat. Výrobce také poskytuje poměrně širokou škálu knihoven a elektronický katalog, kde je možné stahovat již hotové modely nebo příslušné komponenty [13].

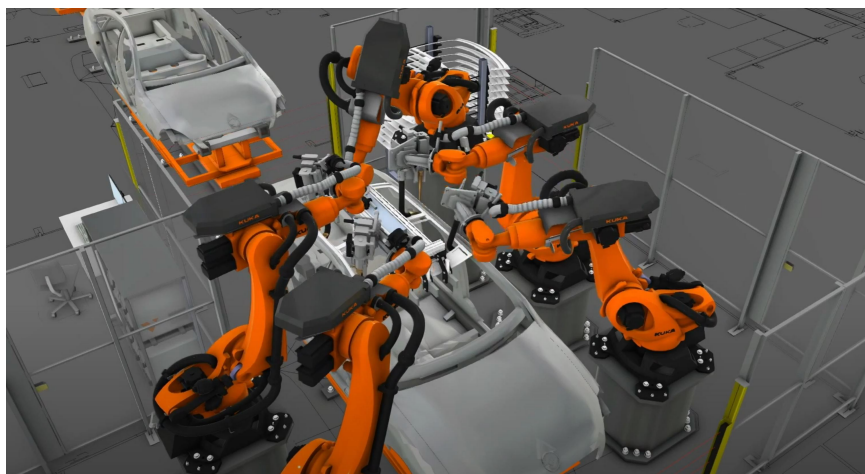
■ 2.3.3 Visual Components

Visual Components je další simulační nástroj umožňující tvorbu 3D modelů, který je, v době psaní této práce, poměrně nový (první stable release byla vydána v roce 2016). Visual Components je dostupný ve 3 dílčích licencích. Základní licence softwaru, tzn. Essentials, obsahuje základní nástroje pro tvorbu modelů, základní balíčky pro programování robotů, import dat z CAD softwarů a také zobrazení modelu do virtuální reality. Dražší verze Professional obsahuje vše co základní verze Essentials a navíc nástroje pro tvorbu vlastních 3D objektů a úpravu importovaných CAD modelů. Nejpokročilejší verze Premium obsahuje pokročilejší knihovny pro programování a navigaci robotů,



Obrázek 2.2: Ukázka programového prostředí Siemens Plant Simulation, zdroj: [13]

interaktivní VR model, konektivitu PLC nebo online prezentace modelů ve virtuální realitě. Visual Components poskytuje 30denní zkušební licenci verze Essentials pro studenty [14, 13]. Grafický výstup softwaru Visual Components je možné vidět na obrázku 2.3.



Obrázek 2.3: Ukázka programového prostředí Visual Components, simulace robotického svařování karoserie auta, zdroj: [14]

2.3.4 FlexSim simulation software

3D simulační software od společnosti FlexSim Software Products, který slouží k modelování výrobních systémů, manipulaci s materiálem, skladování a

celou řadou dalších podnikových procesů. FlexSim obsahuje také modul FlexSim Healthcare Simulation, který usnadňuje modelování procesů v oblasti zdravotnictví [15].

FlexSim vytváří model na 2 základních úrovních, 3D objektové úrovni a tzv. process flow. Ve 3D objektové úrovni uživatel propojuje jednotlivé fyzické reprezentace objektů (jako jsou stroje, dopravníky, sklady, AGV apod.). V process flow uživatel vytváří blokové schéma logických vazeb (obdobně jako v simulačním softwaru Arena), které reprezentují jednotlivé činnosti, procesy nebo děje. Process flow tak například umožňuje vytvořit speciální strategii skladování pro 3D objekt reprezentující vysokozdvižný vozík. FlexSim obsahuje základní katalog objektů jak pro objektovou úroveň, tak pro process flow [15].

Velkou výhodou FlexSimu je jeho široká uživatelská online komunita, kde je možné dohledat mnoho řešených problémů. K FlexSimu existuje na internetu také velké množství video návodů a tutoriálů, což významně zlepšuje jeho uživatelskou přívětivost. FlexSim na svých stránkách umožňuje bezplatně stáhnout časově neomezenou demo licenci, která je omezena počtem třiceti 3D objektů a třiceti-pěti objekty process flow. I s touto volně dostupnou licencí však lze vyřešit mnoho běžných podnikových problémů, což dokazuje praktická část této diplomové práce, kde byl vytvořen model dvou skladových systémů za použití demo licence.

Příklad modelu vytvořeného pomocí softwaru FlexSim je možné vidět na obrázku 2.4 Podrobná charakteristika a základní práce s tímto softwarem je uvedena v praktické části této diplomové práce.



Obrázek 2.4: Model skladu v softwaru FlexSim, zdroj: vlastní práce autora

Kapitola 3

Simulační studie

Simulační studie je studie vzniklá za účelem specifikace a realizace simulačního projektu. Simulační studii lze dle [2] rozdělit na čtyři základní části: **vymezení projektu**, **tvorba a testování modelu**, **realizace experimentů** a **vyhodnocení projektu**. Tyto části jsou podrobně vysvětleny v následujících podkapitolách. Simulační studie je důležitou součástí celého simulačního projektu. Schéma postupu při realizaci simulačního projektu je možné vidět na obrázku 3.1.

3.1 Vymezení projektu

Tato část simulační studie je pro přehlednost rozdělena do dalších dvou částí: **formulace projektu** a **definování projektu**.

3.1.1 Formulace projektu

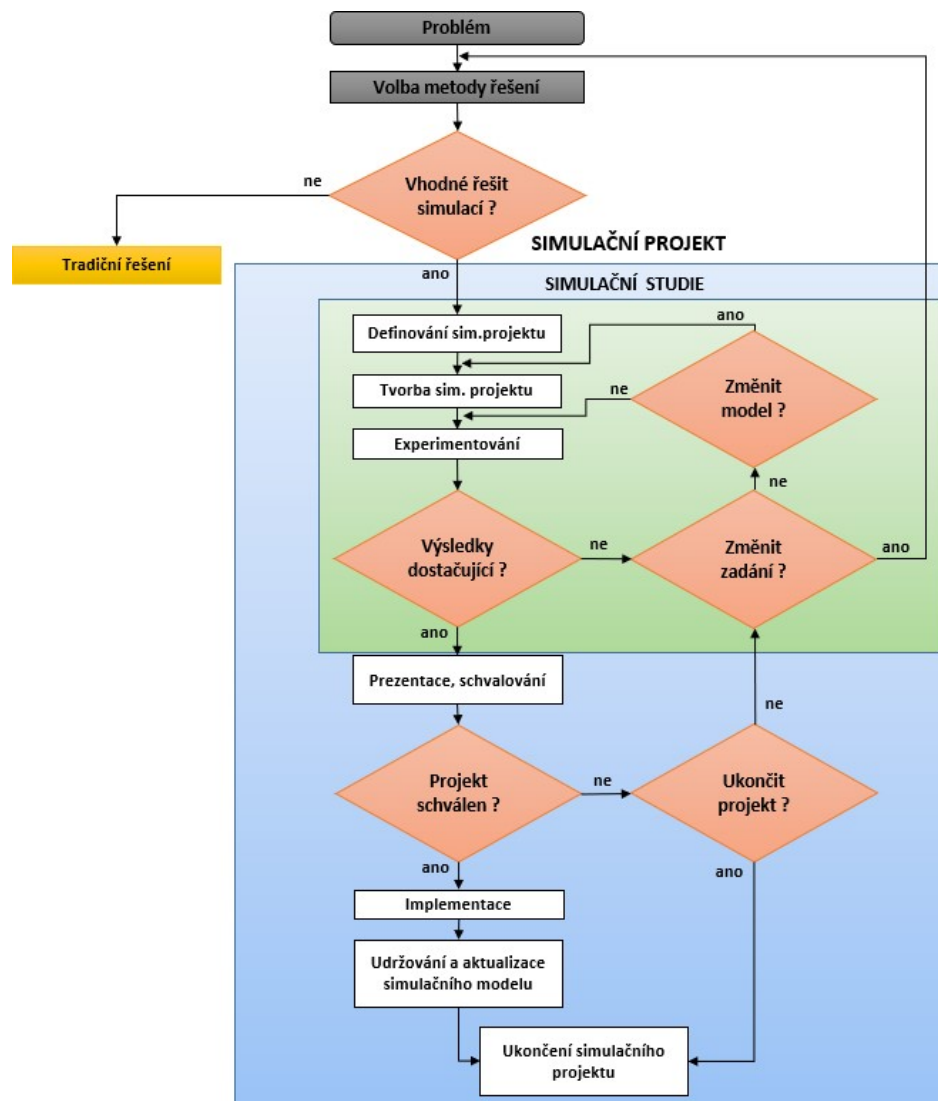
V této fázi simulační studie je nutné si zkoumaný problém nastudovat, rozebrat a konzultovat se zadavatelem studie. Je třeba ujasnit si požadavky zadavatele a vyjasnit zadavateli přednosti a omezení požití simulací a dle toho následně rozhodnout, zda je daný problém efektivní řešit pomocí simulační studie a jejích nástrojů, nebo zvolit jinou metodu řešení daného problému. Typické problémy, které je vhodné řešit pomocí simulací byli podrobně rozebrány v kapitole 2.8 *Využití simulací v oblasti výroby a manipulace s materiálem* [2].

3.1.2 Definování projektu

V případě, že je daný problém vhodný řešit pomocí simulací, lze přistoupit k dalšímu bodu, kterým je definice zadaného projektu. V tomto bodě jde vlastně o přípravu obchodní nabídky, definici cílů simulační studie, stanovení harmonogramu a plánu celého projektu a řešených scénářů [2].

3.2 Tvorba a testování modelu

Simulační model je základní stavební kámen simulace. Samotná simulace je pak pouze zobrazením chování modelu a jeho reakcí na vstupní veličiny. Simulační



Obrázek 3.1: Schéma simulačního projektu

model je zobecněným zobrazením reálného světa (přes relaci podobnosti). Před tvorbou modelu obvykle analyzujeme reálný systém, určujeme vazby mezi jednotlivými prvky a poté je převádíme do modelu. Vynecháváme však vazby, které nemají vliv, nebo mají zanedbatelný vliv na zkoumané chování systému. Tímto model zjednodušujeme, výsledný model se od reálného systému pak vždy liší v určitých nepřesnostech, které by neměly překročit stanové meze. Náhodné jevy pak modelujeme pomocí příslušných statistických rozdělení. Model můžeme charakterizovat následujícím zobrazením: [5]

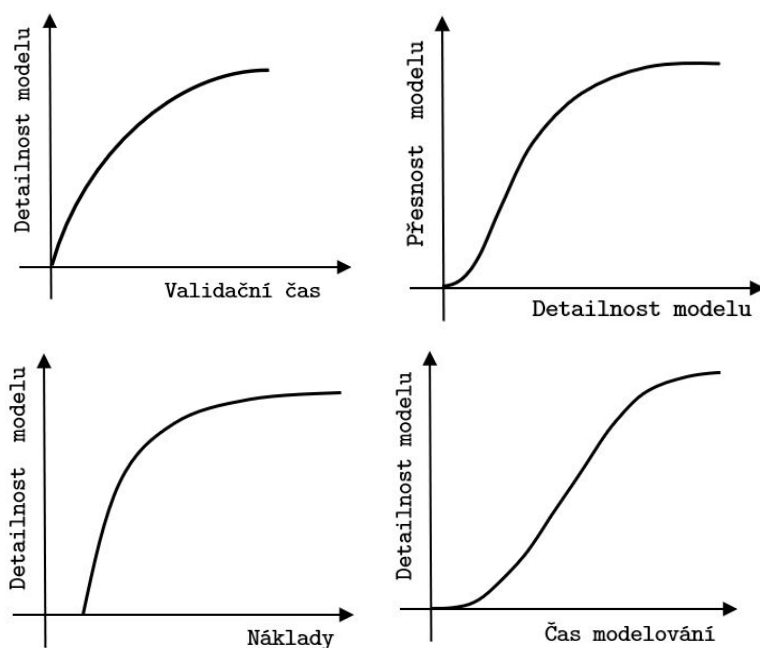
$$S_p \rightarrow M = (S_m, R_p) \quad (3.1)$$

kde

- S_p představuje reálný systém (modelovaný originál)

- M je model systému
- S_m je vytvořený modelový systém
- R_p je relace podobnosti S_m a S_p

Tvorba modelů pak není ničím jiným, než hledáním příslušné relace podobnosti. Můžeme tedy hovořit o tom, že model je do jisté míry detailní. Tomuto se také říká *úroveň abstrakce* modelu. Čím více je model abstraktní, tím méně časově náročnější je samotná tvorba a tím kratší je výsledný čas na validaci modelu (validace modelu je proces ověření, kdy ověřujeme správnost našeho modelu a shodu s reálným systémem). Naopak, čím detailnější model budeme chtít vytvořit, tím náročnější bude samotná tvorba a i validace zabere mnohem více času. Přesnost (věrohodnost) modelu se zároveň od určité míry detailnosti výrazně nezvyšuje či se dokonce nemění. S rozsáhlostí a detailností modelu však nesouvisí jen validační čas, ale také čas modelování a náklady s jeho tvorbou spojené. Všechny výše zmíněné závislosti je možné vidět na grafu 3.2:



Obrázek 3.2: Problematika tvorby modelů v grafech, zdroj: [2]

Z průběhu nákladů, přesnosti, času modelování i validace je tedy patrné, že je nutné přistoupit na určitou míru kompromisu. Obecnou zásadou při tvorbě modelu je, že je lepší detaily přidávat později, tedy v počátku vytvořit základní kostru modelu, která pak bude postupně zpřesňována. [1, 2].

Tvorba simulačního modelu by měla vycházet z dobré znalosti problému a zkoumaného systému. V rámci modelování je třeba úzké spolupráce se zadavatelem simulační studie [17]. Etapy tvorby modelu můžeme shrnout do těchto částí:

3.2.1 Definice struktury systémů

Definování struktury je významná část procesu modelování, kterou není radno podcenit. Je důležité si vytvořit schéma struktury a příslušných vazeb ještě před tím, než přikročíme k samotné práci se SW. Správná výchozí analýza nám může ušetřit mnoho práce při samotném programování modelu [2]. Každý průmyslový systém lze abstrahovat na tři základní jednotky: **Entity**, **zdroje** a **aktivity**. [6]

Entity představují jakýkoliv dynamický objekt který vstupuje do systému, spotřebává zdroje či je dočasně obsazuje a následně model zase opouští, Příkladem takové entity je například zákazník nebo výrobek. [6]

Zdroje jsou spotřebovány či obsazovány entitami. Jejich základním atributem jsou kapacity a v případě vytíženosti či nedostupnosti zdroje se entity následně řadí do front. Příkladem takového zdroje může být např. výrobní stroj. [6].

Aktivity představují jednotlivé procesy které vytváří novou hodnotu v systému. Aktivity vykonávají entity za pomoci zdrojů. [6]. Aktivity lze pro přehlednost znázornit procesní mapou, BPMN (Business Process Modeling Notation) a nebo diagramem aktivit pomocí UML (Unified Modeling Language).

Výsledkem této fáze modelu je tzv. **koncepční model**, který by se dal přirovnat k procesní mapě nebo algoritmickému zobrazení daného reálného systému.

3.2.2 Sběr a analýza dat

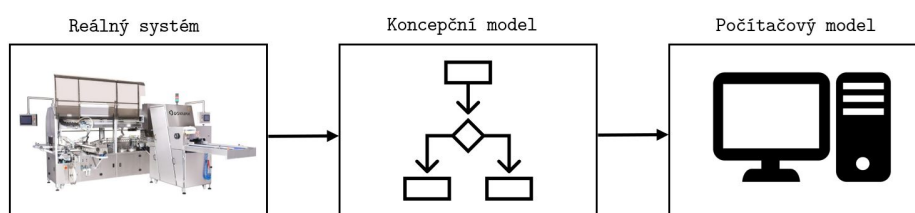
V této části dochází ke sběru informací o simulovaném systému, určují se vstupní proměnné, parametry jednotlivých entit, zdrojů apod. Těmito informacemi se doplňuje a zpřesňuje koncepční model. Významnou složkou je také analýza těchto dat, jelikož v praxi má většina dat stochastický charakter (náhodný). Deterministické parametry (jako např. pojezdová rychlost AGV) se dají snadno určit, ovšem pro určení stochastických parametrů je zapotřebí statistické analýzy. Tyto stochastické parametry se obvykle modelují za použití statistických rozdělení. Některá základní rozdělení jsou popsána v bodech níže [2].

- **Rovnoměrné rozdělení** - časté použití v případě neúplnosti informací o problému, zadává se maximální a minimální hodnota.
- **Trojúhelníkové rozdělení** - taktéž v případě neúplnosti informací, vstupní parametry: minimální hodnota, maximální hodnota, modus.
- **Binomické rozdělení** - v případech, kdy známe pravděpodobnost výskytu určitého jevu (např. procento vadných výrobků ve výrobní dávce).
- **Poissonovo rozdělení** - v případech, kdy známe průměrný výskyt událostí v určitém časovém intervalu (např. průměrný počet poruch za den).

- **Normální rozdělení** - obvykle v případech symetrických odchylek od průměru, nejpoužívanější typ rozdělení.
- **Weibullovo rozdělení** - při modelování délky trvání určitého jevu (např. trvání poruchy).

3.2.3 Tvorba modelu

V této fázi dochází k tvorbě modelu v simulačním programu. Jedná se v podstatě o transformaci koncepčního modelu do programovacího jazyka.



Obrázek 3.3: Fáze tvorby počítačového modelu, zdroj: [16]

3.2.4 Verifikace

V této fázi dochází k ověření funkčnosti a přesnosti daného simulačního modelu. Kvalita výsledků je závislá na přesnosti a adekvátnosti koncepčního modelu a také na programu, který ho vytvořil. Model pak považujeme za správný, pokud při průběhu simulace jeho chování odpovídá představám. Při verifikaci obvykle ověříme správnost [17, 16]:

- Logické a obsahové stránky modelu
- Převodu modelu do programu
- Formální správnost programu v simulačním jazyce

Některé simulační software disponují nástroji, které usnadňují verifikaci modelu. Jedná se například o program, kontrolující správnost kódu při jeho kompilaci, nebo program pro debugování kódu.

3.2.5 Validace

Ve fázi validace dochází k ověření, zda model správně reprezentuje plánovaný systém. Validační techniky jsou buď subjektivní nebo objektivní. Mezi subjektivní techniky se řadí: [2, 16]

- Analýza sensitivity
- Test extrémních podmínek
- Test konzistence

- Face validation

Objektivní techniky jsou: [2, 16]

- Validace transformace vstupů na výstupy
- Validace s využitím historických vstupních dat

3.3 Provedení experimentů

Experimenty v simulační studii pak rozumíme provedení vlastního simulačního běhu a zhodnocení jejich výsledků. V rámci experimentální části se obvykle také provádí optimalizace simulačního modelu.

Experimenty by také měli poskytovat statisticky relevantní výsledky. Pro získání použitelných výsledků je dle [17] vhodné stanovit:

- Délku simulačního chodu
- Počet experimentů
- Přesnost a spolehlivost výsledků experimentů

Podle délky trvání rozdělujeme simulace dle [6] na *simulace s konečným horizontem* a *simulace dlouhodobého chování*. Simulace s konečným horizontem předpokládá, že je známo pravidlo ukončení simulace. To může být například předem určeným časem (doba uzavření pobočky) nebo počtem průchozích entit (například vyrobení 50 kusů výrobku). V případě simulace dlouhodobého chování neexistuje žádná ukončovací událost, jedná se o analýzu běžného provozu. V případě simulace dlouhodobého chování je nutné odstranit problém vlivu počátečního stavu pomocí tzv. zahřívací doby. Zahřívací doba slouží k odstranění statistických nepřesností daných inicializací startovacích procesů (například naplnění skladu výrobky apod.). Zahřívací dobu lze určit graficky například pomocí sledování klouzavých průměrů vstupních hodnot [6, 2].

Obecné pravidlo pro délku simulačního chodu v případě simulace s konečným horizontem zní, že sledovaná událost by se měla v systému vyskytnout nejméně 10-20 krát [2].

Po získání výsledků simulace je obvyklé daný model optimalizovat. Optimalizaci lze provádět provádět "ruční cestou" (nastavení parametrů modelu dle citu a znalostí simulačního inženýra) nebo pomocí optimalizačních metod. Mezi nejběžnější používané optimalizační metody patří [2, 18]:

1. Heuristické algoritmy
2. Meta-heuristické algoritmy
3. Genetické algoritmy
4. Gradientní postupy
5. Lineární programování

6. Vícekriteriální optimalizace

7. Znalostní systémy

Optimalizačních metod existuje celá řada, přehled většiny používaných optimalizačních metod včetně jejich popisu lze najít v [18].

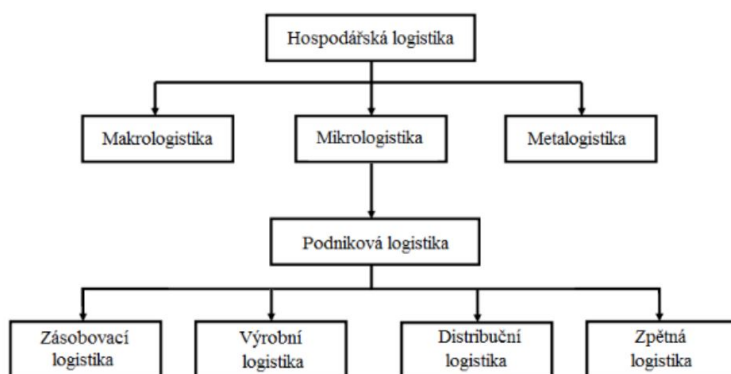
3.4 Vyhodnocení projektu

K této části projektu patří zejména kompletace projektové dokumentace, vytvoření závěrečné zprávy pro odběratele simulační studie a prezentace výsledků. Po simulační studii obvykle (v případě úspěšné simulační studie) nastává fáze zavádění. Je třeba, aby realizační tým byl ve fázi vyhodnocení projektu důsledně obeznámen se smyslem simulační studie a navrhovanými opatřeními. Ideální varianta je, pokud jsou tvůrci simulační studie zároveň i členové realizačního týmu.

Kapitola 4

Logistika a skladování

Logistika skladování je rozsáhlá vědní disciplína, jenž se zabývá řízením podniku v oblasti dodavatelského řetězce a materiálového toku. Logistiku lze obecně rozdělit na tři základní segmenty: makro-logistiku, mikro-logistiku a meta-logistiku. Podniková logistika včetně výrobní logistiky jsou pak podmnožinou mikro-logistiky [19]. Rozdělení logistiky na jednotlivé segmenty můžeme vidět na obrázku 4.1.

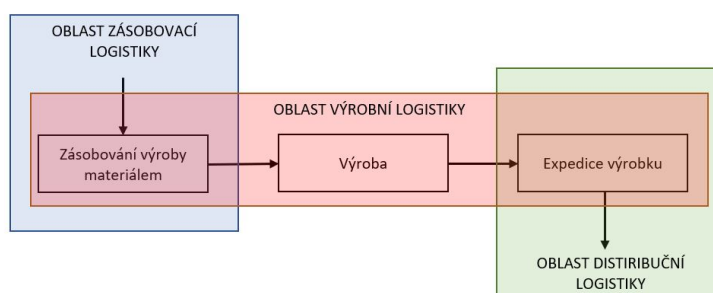


Obrázek 4.1: Obecné členění logistiky, zdroj: [19]

4.1 Výrobní logistika

Výrobní logistiku lze charakterizovat jako přepravní a skladovací činnosti, které spojují ve výrobním procesu jednotlivé výrobní kroky a organizační činnosti. Výrobní logistika se tedy zabývá způsoby manipulace s materiálem ve výrobě, mezi výrobou a skladem a mezi výrobou a expedicí. Výrobní logistika tedy úzce navazuje na logistiku zásobovací a distribuční, jak je možné vidět na obrázku 4.2 [19, 20].

Hlavním cílem výrobní logistiky je pak zajistit pravidelný přísun materiálu k zabezpečení výrobního procesu a zajištění plynulého odsunu hotových výrobků zboží do skladu a jejich následnou expedici [19]. Mezi hlavní činnosti výrobní logistiky pak patří:



Obrázek 4.2: Výrobní logistika v logistickém řetězci, zdroj: [19]

- plánování předvýrobního skladování materiálů a polotovárů
- manipulace s materiálem v různých stupních fáze výroby
- mezioperační a operační doprava
- mezioperační skladování
- manipulace při montáži celků
- manipulace s hotovými výrobky
- zajištění příznivých pracovních podmínek pro zaměstnance

4.2 Skladová logistika

Skladová logistika je konkrétní aplikací výrobní logistiky pro potřeby provozu. Je zaměřena nejen na fyzickou manipulaci a skladování materiálu, ale také na integraci a koordinaci plánovacích, řídicích, kontrolních činností s cílem snížit náklady, zvýšit výkon a kvalitu všech činností spojených s materiálovým tokem.

Základní logistickou jednotkou jsou sklady. Sklad je jakýkoliv objekt, budova či prostor určený k ukládání předmětů. Jejich úkolem je obvykle zásobit podnik materiálem pro kontinuální výrobní proces, zabezpečování výroby v podobě rezervních zásob, uskladnění výrobního sortimentu nebo sortimentu zboží nebo mohou výrobky skladovat pro jakostní zrání výrobků (např. zrání alkoholických nápojů v sudech) [19, 20]. Sklady lze obecně rozdělit na: [20, 19]

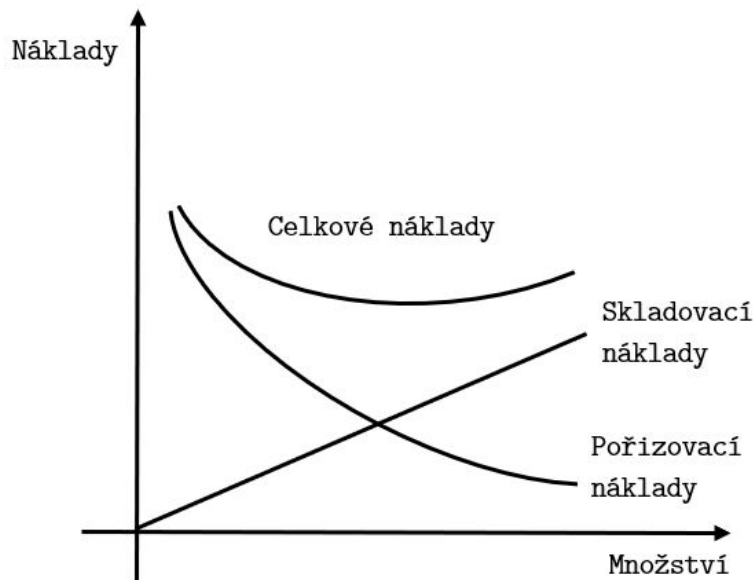
1. Podle funkce skladovacího systému

- a. **Sklady materiálu** - vstupní sklady určené k udržování zásob výrobních materiálů.
- b. **Mezisklady** určené ke skladování mezi různými částmi výrobního procesu.

- c. **Skлады hotových výrobků** - odbytové sklady určené k vyrovnávání časových rozdílů mezi výrobními a odbytovými procesy nebo čistě odbytové sklady sloužící pro skladování zboží v případě nevýrobních podniků.
 - d. **Skладиště odpadu** z výrobního procesu.
2. **Podle konstrukce** (uzavřené, kryté, otevřené, etážové ...)
 3. **Podle stupně mechanizace a automatizace** (automatizované sklady, plně automatizované, vysoce automatizované, mechanizované)

Požadavky na sklad a jeho konstrukci jsou pak obvykle dány jeho typem a funkcí. Sklad je třeba při návrhu dimenzovat vždy i s ohledem na předpokládaný budoucí vývoj a požadavky. Výstavba skladu totiž představuje často velké finanční investice, nebylo by tak příliš vhodné, aby ještě před uplynutím odpisové doby podnik musel kvůli nedostatečným skladovacím prostorům realizovat sklad nový. Navíc v průmyslu je obecně patrná tendence, že počet položek v sortimentu spíše roste než klesá, kvůli obecnému cíli prodeje - nabízet kompletní sortiment výrobků [21].

Skладovací náklady mají charakter nákladů variabilních a jejich vývoj v závislosti na množství sleduje přímku přímé úměry. Pořizovací náklady pak ze své podstaty s objednaným množstvím klesají a jejich křivka má tvar lomené funkce. Průběh těchto nákladů je možné vidět na obrázku 4.3. Je vidět, že funkce celkových nákladů má globální minimum. Nalezením tohoto minima pak dostáváme **optimální nákupní množství**.



Obrázek 4.3: Výrobní logistika v logistickém řetězci, zdroj: [20]

4.3 Druhy skladů

Tato sekce se věnuje dostupným typům skladů, zejména pak skladů vstupních (sklady materiálu) a výstupních (odbytové sklady). Vzhledem k aktuálnímu trendu automatizace a robotizace je pak širěji rozebraná část věnující se problematice automatických skladových systémů.

V následujících bodech jsou popsány v praxi běžně používané typy skladů:

4.3.1 Manuálně obsluhované sklady

Manuálně obsluhované sklady jsou běžné typy skladů, kde vyskladňování a zaskladňování předmětu z a do skladu provádí pracovník. Tyto sklady mohou mít různý stupeň mechanizace (např. usnadnění manipulace pomocí vysokozdvížného vozíku).

Pevné regálové sklady

Pevné regály jsou nejběžnějšími typy regálů v menších skladech. Mají robustnější konstrukci a jsou navrženy tak, že se nepředpokládá manipulace s jednotlivými regály [22].

Pojízdné regálové sklady

Pojízdné regály jsou určeny spíše pro menší velikosti skladů. Regály (nebo skříně) jsou namontovány na kolejnicový systém zapsuštěný do podlahy. Pro vyskladnění předmětu je tedy obvykle nutné pohnout s jedním nebo více regály, aby se mezi nimi uvolnil prostor a pracovník mohl projít na místo uložení. Tyto regály tak obvykle nejsou vhodné pro velké plochy skladů ani pro těžké předměty. Jejich výhodou oproti pevným regálům je v úspore skladovací plochy. Hodí se spíše pro menší provozy s vysokým nárokem na úsporu prostoru [22].

Pokud jsou ovšem i ve větším výrobním podniku velké nároky na úsporu prostoru, je možno zkonstruovat i poměrně robustní řešení velkých pojízdných regálů, které umožňuje použít paletové systémy pro těžké předměty a vysokozdvížné vozíky.

Paletové sklady

Paletové sklady využívají palety jako úložnou jednotku pro jednotlivé skladovací jednotky. Skladovací jednotky (např. balíčky) se skládají na paletu a paleta se pak uskladňuje do regálového skladu. Paletové systémy vyžadují obvykle k manipulaci s paletami vysokozdvížné vozíky, jedná se většinou tedy o mechanizované sklady. Palety se vyrábějí z plastu, překližky nebo dřeva. Paletové systémy pak mohou využívat buď pevných nebo pojízdných regálů. [23]

■ Víceúrovňové regálové sklady

Víceúrovňové regálové sklady se obvykle používají pro velké zásoby zboží s menšími rozměry. Víceúrovňový regál je charakteristický svojí patrovou strukturou, kde stojí dvě a více pater na sobě. Mezi jednotlivými podlažními jsou obvykle schodiště pro pohyb zaměstnanců nebo pojezdové plošiny pro pohyb ještěrek [22].



Obrázek 4.4: Víceúrovňový regálový sklad, zdroj: [22]

■ 4.3.2 Automatizované skladovací systémy

V dnešní době, kdy se stále klade větší důraz na rozvoj Průmyslu 4.0, automatizaci a robotizaci, přibývá stále větší počet skladovacích řešení, které pro realizaci materiálového toku nepotřebují člověka. Tyto skladové systémy jsou novou, nastupující generací, jelikož zejména ve větších provozech přináší značné úspory a zároveň pomáhají řešit problém nedostatku pracovní síly v oblasti skladování. Například norská společnost Elektroimportøren uvádí, že investice do automatizovaného skladu od společnosti AutoStore se vrátila již po dvou letech. Výhoda těchto řešení však není jen v rychlejším naskladnění, vyskladnění či redukci zaměstnanců, ale i v úspoře skladovacího prostoru.

V dnešní době existuje takovýchto automatizovaných skladovacích řešení již celá řada, pro podnik tak nemusí být snadné zvolit variantu, která co nejlépe odráží jeho logistické potřeby. Přehled některých významných automatizačních řešení je následující:

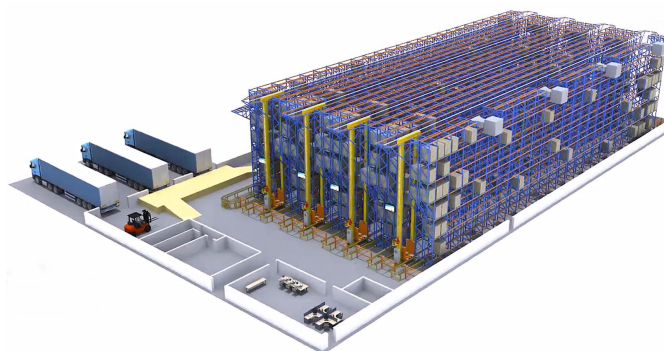
■ Sklad automatizovaných zakladačů (AS/RS)

AS/RS je zkratka z anglického Automated Storage and Retrieval System. Jedná se o typ skladu, kdy zakládání i vyskladnění požadovaného předmětu probíhá automaticky pomocí robotického ramene. Pro AS/RS skladovací

system je typické, že se skládá z více robotických jednotek, které obstarávají pohyb v rovině yz a xy . Pohyb v rovině xz pak obstarává soustava dopravníků (tento pohyb představuje v podstatě dopravení zboží ke stanovišti příslušného zakladače). Tyto sklade lze dle [24] na:

■ Systém hlubinného zakládání

Tento systém se obvykle sestává ze 2 základních struktur. Dopravníkové struktury, která pomocí dopravníků přepravuje zboží od přejímky k příslušnému zakladači, a systému zakladačů. Tyto zakladače obvykle obsluhují příslušnou dvojici regálů. Po tom, co dopravník přepraví zboží k zakladači, zakladač zboží naloží a pohybuje se v rovině yz na příslušné volné místo. Obdobným principem funguje i vyskladňování produktu ze skladu. Schéma tohoto skladu je vidět na obrázku 4.5.



Obrázek 4.5: AS/RS - hlubinné zakládání, zdroj: [24]

■ Kompaktní úložný systém

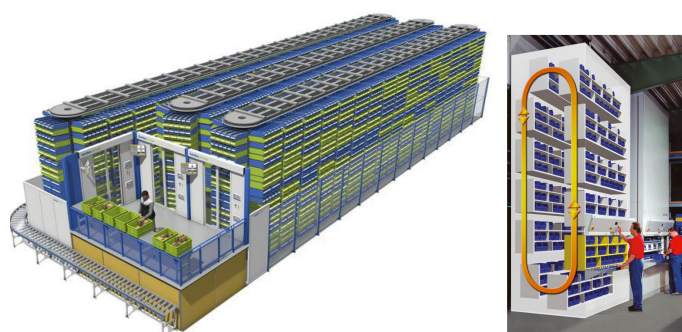
Je velmi podobný předchozímu AS/RS systému jen s tím rozdílem, že zde nejsou volné uličky mezi jednotlivými regály. Celý sklad tedy tvoří kompaktní krychli. Pohyb výrobku v jednotlivých rovinách neprovádí zakladač a soustava dopravníků jako v předešlém případě, ale zakladač a dopravníková linka umístěná v regálových patrech. To znamená, že každé patro skladu je tvořeno dvojicemi dopravníkových linek ve tvaru O (tyto dvojice jsou spojené a umožňují přemístění hledaného zboží k ramenu jeřábu). Jeden pár dopravníku bývá obvykle poháněný do mírného sklonu, druhý pár je gravitační dopravník. Pro přehlednost je tento princip zachycen na obrázku 4.6.

■ Karuselové AS/RS systémy

U tohoto systému jsou všechny regály spojeny k sobě a svou rotací pak vytvářejí uzavřenou smyčku. Rotace je buďto horizontální nebo vertikální. Podle rotace pak rozlišujeme horizontální karuselové systémy nebo vertikální karuselové systémy. Konstrukce u každého tohoto systému je rozdílná, viz obrázek 4.7.



Obrázek 4.6: Kompaktní AS/RS a jeho princip, zdroj: [24]

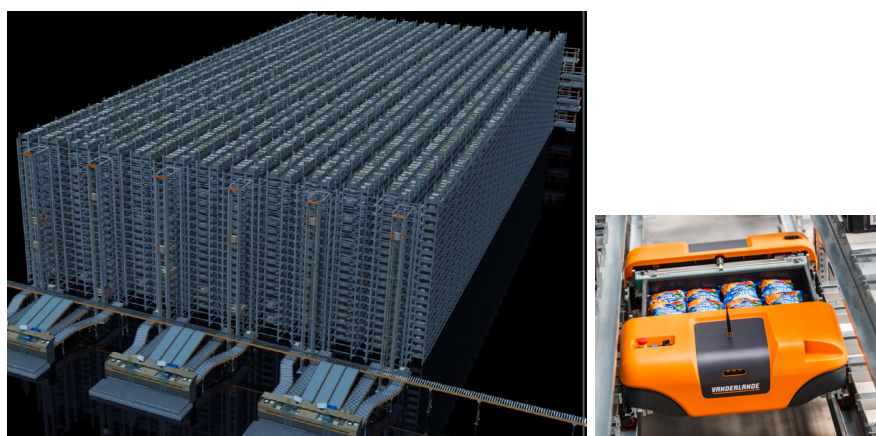


Obrázek 4.7: Vlevo: horizontální karuselový systém, vpravo vertikální systém, zdroj: [24]

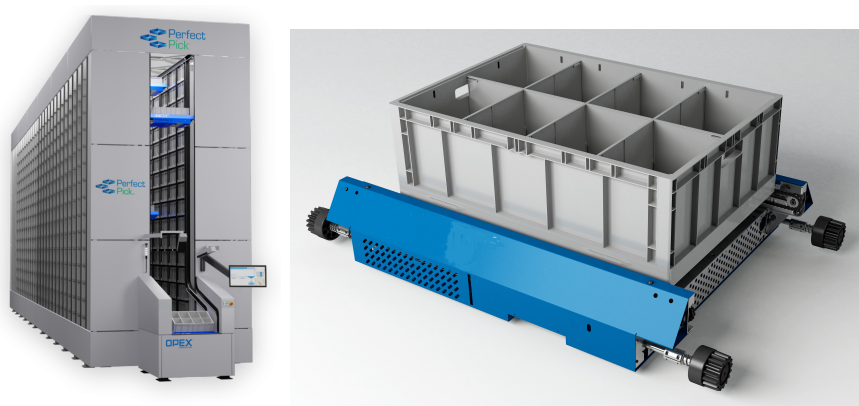
■ Systém AVS/R

Systém AVS/R, z anglického Autonomous Vehicle-based Storage and Retrieval System, využívá k naskladnění a vyskladnění autonomních vozíků. Mezi jednotlivými regály jsou umístěny uličky s kolejnicemi, po kterých jezdí autonomní vozítka. Tyto vozítka nakládají a vykládají zboží pomocí výsuvného ramene. Jelikož jsou vozítka schopna pohybu pouze v dané úrovni skladu (nemohou cestovat nahoru po souřadnici z), jsou po straně skladu umístěny výtahy (u AS/RS to byly zakladače, což je další odlišnost těchto systémů), které umožňují pohyb zboží z horních pater do vstupního/výstupního portu. Výhodou tohoto řešení oproti AS/RS je, že není nutné instalovat konstrukci zakladačů v každé uličce skladu jako u AS/RS, jelikož vozítka se mohou pohybovat po celé rovině xy dané úrovně, což urychluje dobu vyskladnění zboží. Vozový park robotů lze také přizpůsobit podle potřeby vytíženosti skladu. Realizací a vývojem tohoto typu skladovacích systémů se zabývá firma Vanderlande, a tento systém je implementován např. v Istanbulsém letišti.

Speciálním typem AVS/R systému je pak **AVS/R vertikálně - horizontální systém**. Tento systém využívá robotů, kteří se pohybují ve všech rovinách, jsou tedy schopni i vertikálního pohybu ("šplhají nahoru po regálech"). V klasickém AVS/R systému vozítka nejsou schopna vertikálního



Obrázek 4.8: Vlevo: systém AVS/R od společnosti Vanderlande, vpravo robotické vozítko, zdroj: [24]



Obrázek 4.9: Systém iBot od firmy Opex, zdroj:[24]

pohybu a je třeba mít v každé úrovni příslušný počet vozítek. Velmi zajímavým řešením je pak iBot od firmy OPEX. Tento systém se skládá z bloků, které tvoří dvojice regálů s uličkou uprostřed, mezi kterými se ve vertikální i horizontálním směru pohybují roboti, kteří zaskladňují a vyskladňují zboží. Schéma je vidět na obrázku 4.9.

■ Shuttle/Robot-Based Compact Storage systémy

Charakteristika těchto skladů je realizace pohybu materiálu pomocí AGV nebo robotických systému a dále pak snaha o úsporu co největšího místa, tedy dosažení co největšího skladovacího poměru. Tyto skladovací systémy obvykle ukládají předměty do velmi husté krychlové sítě. Podle konkrétní realizace skladovací aktivity rozdělujeme tyto systémy na:

■ Puzzle-Based Compact Storage systémy

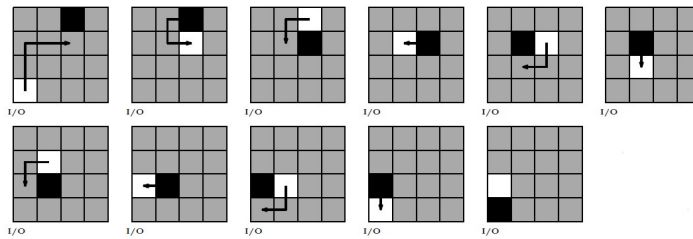
Tento systém vychází z matematické úlohy nazývané jako Sam Loydovo puzzle. Jedná se o algoritmus pro řešení hry, kde v hracím poli skládajícím se z čtvercových obsazených bloků a čtvercových volných bloků

se přemísťují jednotlivé náhodné rozmístěné čtverce do přesně daného pořadí. Schéma hry je znázorněno na obrázku 4.10 [24].

1	3	2	4
12	14	15	5
13		6	11
9	10	8	7

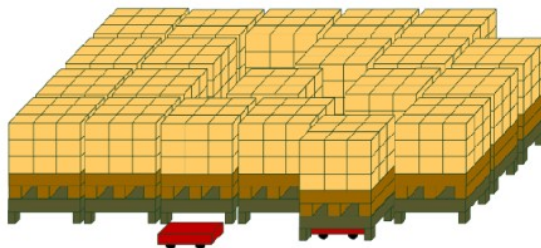
Obrázek 4.10: Sam Loydovo puzzle, zdroj:[24]

Sam Loydova úloha aplikovaná na skladovací systém pak může vypadat způsobem zobrazeným na obrázku 4.11, kde v úložišti o velikosti 4x4 s jedním I/O portem (pozn. I/O značí input-output, tedy vstup-výstup) o skladovací hustotě $(n - 1)/n$, kde n označuje celkový počet skladovacích buněk, je třeba vyskladnit obsah náhodně zvolené buňky (na obrázku černé čtvercové pole) [24].



Obrázek 4.11: Sam Loydova úloha pro jednoduchý sklad, zdroj:[24]

Na Loydovo puzzle navázal Kai Furmans, který v roce 2011 ve svém článku "*Future of material handling: modular, flexible and efficient*" představil koncept tzv. GridFlow systémů. Tento systém spočívá ve speciálním paletovém uspořádání skladu, kde pod paletami jezdí AGV, které realizují pohyb palety. Paleta představuje obsazený čtverec v Loydově úloze. Schéma tohoto systému je možné vidět na obrázku 4.12 [24].

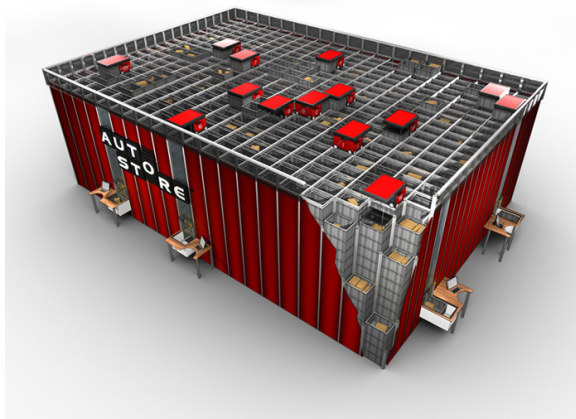


Obrázek 4.12: Furmansův GridFlow skladovací systém, zdroj:[24]

V současnosti však nefunguje žádný sklad na principu Loydovy úlohy a GridFlow systému. Některé společnosti však již podobný systém vyvíjejí, a proto můžeme očekávat, že skladovací systém na principu GridFlow bude v blízké budoucnosti v praxi použit.

■ RCSR systémy (Robot-Based Compact Storage and Retrieval Systems)

Tyto sklady jsou tvořeny velmi hustou krychlovou sítí skladovacích pozic s mřížovou konstrukcí ve své horní části. V této konstrukci obvykle jezdí roboti, kteří vyzvedávají jednotlivé boxy ze skladovacích pozic. V boxech jsou pak uskladněny jednotlivé předměty. Robot vyzvedne vždy příslušný horní box, a pokud se požadovaný předmět nenachází v horním boxu, přemístí ho o pozici vedle a tento proces opakuje tak dlouho, dokud nevyzvedne požadovaný box. Tento box pak následně odveze na příslušnou I/O (*Input/Output*) pozici. Volný robot pak obvykle zaskladní přeházené boxy zpátky na místo. Typickým příkladem tohoto principu skladování je norský systém AutoStore, nebo systém britského online potravinového eshopu Ocado. Oba dva tyto systémy pracují na principu RCSR, liší se však použitými algoritmy. Jednotlivé roboty jsou pak řízeny centrálně ze specializovaného počítače, se kterým komunikují pomocí bezdrátové sítě. Centrální počítač vykonává tzv. Air Traffic Control (Řízení letového provozu) a koordinuje dráhy robotů tak, aby se nesrazily. Do jednotlivých boxů jsou ukládány předměty o podobné obrátkovosti. Tyto boxy jsou pak obvykle umístované v horních polohách skladiště, tím dochází k rychlejšímu odbavování požadavků. Algoritmy řídicí RCSR systémy jsou velmi pokročilé a složité a tvoří významné know-how společností, které tyto sklady vyvíjejí.



Obrázek 4.13: RCSR systém od společnosti AutoStore, zdroj: autostoresystem.com

■ RMFS systém (Robotic Mobile Fulfillment Systems)

Tento systém je svým chováním podobným GridFlow systému, avšak neřadí se do compact storage systému z důvodu, že nedosahuje tak velkého kompresního skladovacího poměru. Tento systém byl představen společností Amazon a používá se typicky pro případy, kdy je nutné skladovat velké portfolio drobných výrobků. Systém tvoří speciálně ergonomicky navržené odbavovací stanice, robotické vozítka a regály. V tomto systému dochází k manipulaci s celým regálem. Robotická vozítka najíždí pod příslušné regály, nadzvednou je a odvezou k příslušné pickovací stanici. Tento systém je velmi snadný na případné rozšíření a také velmi flexibilní (lze snadno měnit počty regálů i robotů). To je obvykle velmi výhodné pro internetové prodejce, kteří mají často volatelnou poptávku po svých produktech. Jednotlivá vozítka jsou obdobně jako u RCSR systému řízeny pomocí centrálního počítače, se kterým komunikují. V dnešní době však existují již decentralizované RMFS systémy, kde spolu jednotliví roboti komunikují navzájem [24].



Obrázek 4.14: Robotické vozítka, také označované jako kiva robot, zdroj: [24]



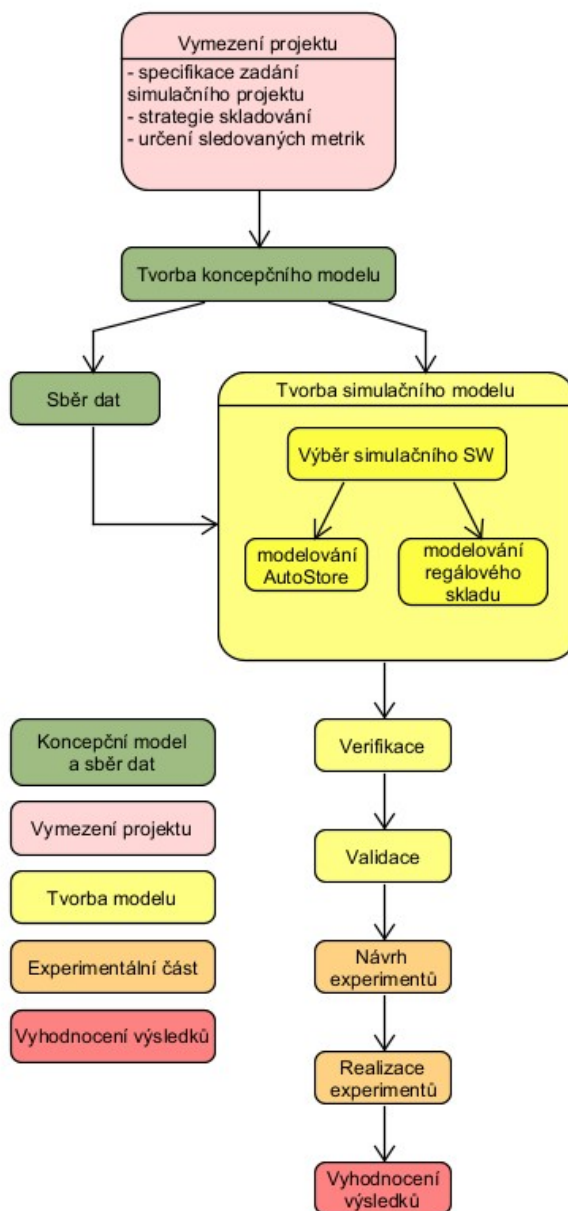
Praktická část

Kapitola 5

Specifikace praktické části

Praktická část této práce se týká simulační studie robotického skladu typu RCSR (Robot Base Compact Storage), konkrétně skladovacího robotického systému od společnosti AutoStore. V rámci simulační studie je provedena analýza fungování robotického systému, a dále je rovněž tento systém porovnán s běžným typem regálového skladu.

V praktické části je nejprve popsán použitý simulační nástroj: FlexSim Express 2021. Mimo tento simulační software autor použil pro generování náhodných čísel také tabulkový procesor MS Excel. V dalších kapitolách je pak řešena simulační studie dle struktury, která byla uvedena v teoretické části v kapitole 3. Postup řešení praktické části tak respektuje postup řešení simulační studie. Tento postup je možné vidět na obrázku 5.1.



Obrázek 5.1: Diagram plánu simulačního projektu

Kapitola 6

Popis použitého simulačního nástroje

Pro realizaci simulace zvolil autor této práce simulační software FlexSim, konkrétně jeho volně dostupnou zkušební licenci FlexSim Express ve verzi 2021, která je omezena maximálním počtem 30ti 3D objektů a 35ti objektů process-flow. Tato zkušební verze je volně dostupná z oficiálních stránek FlexSimu <https://account.FlexSim.com/FlexSim-express/>.

Základní popis tohoto simulačního software byl již popsán v kapitole věnující se simulačním nástrojům. Zde budou podrobně vysvětleny termíny, prvky a nástroje simulačního softwaru, které byly použity k tvorbě simulačního modelu a simulaci.

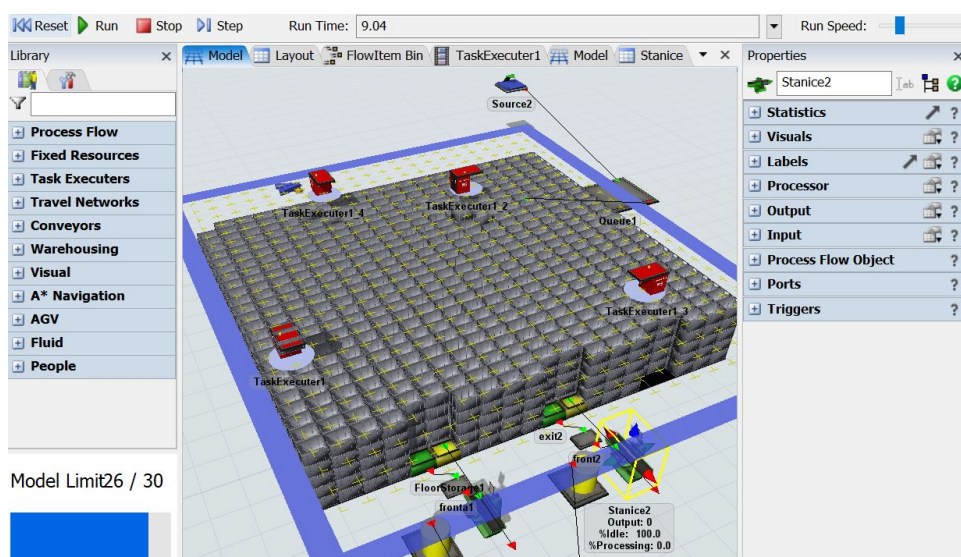
6.1 Uživatelské rozhraní a tvorba modelu

Po spuštění softwaru FlexSim se spustí základní uživatelské rozhraní. To obsahuje záložku Model, která představuje 3D grafickou reprezentaci modelu. Uživatel přemísťuje jednotlivé objekty na 3D pracovní plochu, kde je pak spojuje **connections (vazbami)**. Schéma záložky model můžeme vidět na obrázku 6.1. Na levé straně je vidět seznam základních typů objektů, uprostřed se nachází 3D grafická reprezentace modelu a napravo je pak záložka s podrobnostmi o vybraném objektu.

Vazby jsou ve FlexSimu trojího druhu: Connect Objects, Connect Center Ports a Extended Connect. Nejdůležitější a ve většině případů nejpoužívanějšími jsou pouze první dva druhy spojení. **Connect Objects** slouží pro spojování jednotlivých objektů (např. stroj s dopravníkem, nebo stroj s frontou). **Connect Center Ports** slouží obvykle pro připojení task executors (např. AGV nebo pracovníků) k ostatním objektům. Při propojení dojde k vytvoření reference mezi objekty, čímž jeden objekt získá informace o tom druhém. Výrobní stroj (processor) tak například pozná, jestli je připojený operátor k dispozici.

6.2 Terminologie

K pochopení základů práce se softwarem FlexSim je třeba vysvětlit některé základní termíny, se kterými FlexSim pracuje:



Obrázek 6.1: Záložka model

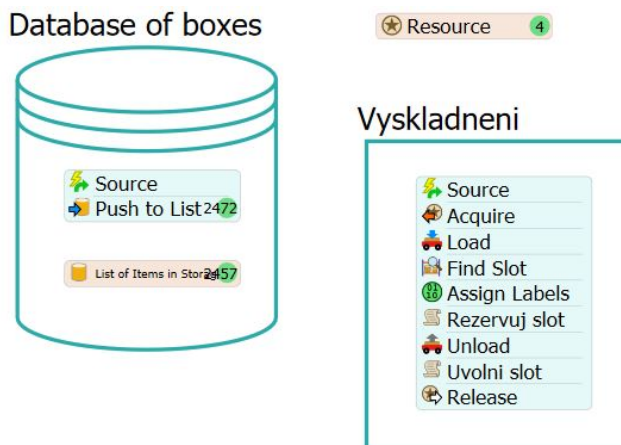
■ Objects (objekty)

Objekty fungují jako základní jednotky programového modelu a jsou reprezentovány svojí 3D grafickou podobou. Objekty se ve FlexSimu dělí na několik základních typů, v této práci uvádíme pouze objekty použité při modelování:

- *Flow items* - objekty tvořící tok entit uvnitř modelu (mohou charakterizovat jak materiálový tak informační tok). Příkladem může být AutoStore box, který se v průběhu skladovacího procesu opětovně vyskládňuje a zaskladňuje.
- *Fixed resources* - jedná se o objekty, které jsou statické a mají svoji specifickou funkci. V kontextu simulační terminologie se tak jedná o zdroje (nejčastěji stroje, pracoviště apod.). Nejběžnějšími typy fixed resources v simulačním prostředí FlexSim je queue (fronta), source (zdroje), sink nebo processor, viz. popis použitých objektů v simulačním modelu.
- *Task executors* - task executors jsou pohybliví vykonavatelé příslušných aktivit. Může se jednat například o pracovníky, AGV, roboty apod.
- *Warehousing* - jedná se o objekty, které reprezentují prvky skladů. Každý tento objekt má svou databázi skladových míst, tzv. **slotů**. Jednotlivým slotům lze pomocí nástroje **Paint Slot Labels** přiřadit tzv. **Labely** (vysvětleno níže).
- *A* navigator* - jedná se o objekty, které souvisí s navigací a modelováním chování objektů typu task executors. Slouží pro zamezení kolíží mezi task executors i mezi task executors a ostatními požadovanými objekty.

■ Process Flow Objects (objekty procesního toku)

Jedná se o objekty, které nejsou součástí 3D modelu a nemají svoji grafickou 3D reprezentaci. Tyto objekty modelují základní prvky všech aktivit, operací a procesů. Skládáním těchto prvků můžeme vytvořit libovolně složitý proces. Process Flow se vytváří ve vlastní záložce Process Flow. Objekty prochází entita nazvaná Token (obdoba flow item), která při průchodu objektem process flow spouští jeho funkce. Tokem tokenů těmito objekty se může řídit libovolně složitý proces. Příklad takového Process Flow je možné vidět na obrázku 6.2.



Obrázek 6.2: Ukázka nástroje Process Flow a jeho objektů

■ Labels (Štítky)

Neboli štítky jsou proměnné, které lze přiřadit ke každému modelovému objektu i objektu process flow. Slouží k uchování informací o daném objektu (např. o jaký typ výrobku se jedná, jeho souřadnice apod.).

■ Lists (Listy)

Listy jsou v podstatě seznamy položek. Obvykle slouží jako databáze pro uchování informací. Příkladem takového listu je např. seznam položek ve skladu. Pro práci s listy slouží v process flow tzv. Shared Assets.

■ Tables (Tabulky)

Tabulky slouží pro uchování informací, ty mohou být využity jednak pro dané procesy (generování výrobků podle tabulky, přiřazení labelu dle tabulky apod.) nebo zde mohou být zaznamenávány informace o procesu. S informacemi obsaženými v tabulkách pak lze v modelu dále pracovat.

6.3 Popis základních použitých 3D objektů

V následujících bodech jsou uvedeny 3D objekty, které byly použity při tvorbě simulačních modelů.

6.3.1 Source

Source je generátor flow itemů. V nastavení tohoto objektu lze zvolit, který typ flow itemu se vytvoří a kdy dojde k jeho vytvoření. Source umožňuje generovat flow itemy v daném čase dle zadaných hodnot (lze nastavit např. na pevnou hodnotu nebo generovat dle statistického rozdělení) nebo dle exportované tabulky ve formátu xls. nebo csv. Grafická 3D podoba objektu source je zobrazena na obrázku 6.3.



Obrázek 6.3: Source

6.3.2 Processor

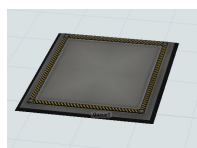
Processor (na obrázku 6.4) reprezentuje nejčastěji stroj nebo určitý proces, kterým prochází flow item. V nastavení processoru nastavujeme nejčastěji dobu trvání procesu a limit počtu najednou zpracovávaných flow itemů. Lze nastavit také různé triggerery (spouštěcí události), které přidělí flow itemu label nebo změní jeho vizuální podobu.



Obrázek 6.4: Processor

6.3.3 Queue

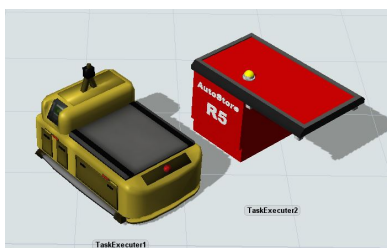
Queue neboli fronta reprezentuje mezisklad. Je to dočasné skladiště flow itemů, obvykle umístované před procesor. Základní nastavované parametry fronty jsou její typ (FIFO - First In First Out, LIFO apod.) a velikost. Grafickou podobu queue je možné vidět na obrázku 6.5.



Obrázek 6.5: Queue

6.3.4 Task Executor

Task Executor (na obrázku 6.6) představuje třídu operátorů úloh, které mají na starost manipulaci s flow itemy. Charakteristické atributy tohoto objektu jsou rychlost, zrychlení, kapacita, rychlost nakládání a vykládání. Pro Task Executors lze definovat dráhy, po kterých se mají pohybovat. Je možné je také přiřadit k A*navigátoru, který pak zamezuje kolizím mezi jednotlivými Task Executors i ostatními přiřazenými objekty.



Obrázek 6.6: Task Executors, vlevo standardní FlexSim model, vpravo upravený design pro potřeby práce

6.3.5 Operator

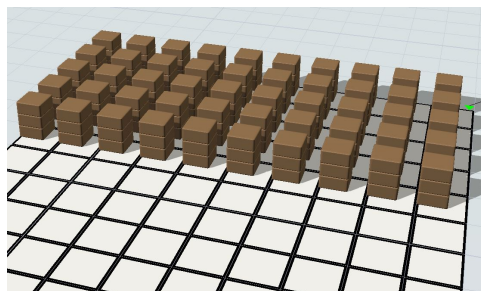
Operator (zobrazen na obr. 6.7) je speciálním druhem objektu Task Executor, který představuje lidského pracovníka. Ve FlexSimu jsou nastaveny základní parametry a kinematiky pohybů pro pracovníky, včetně několika druhů jejich vizuálů. U pracovníka jsou základní charakteristiky obdobné jako u Task Executor.



Obrázek 6.7: Model operátora

6.3.6 Floor Storage

Je objekt typu Warehousing (skladování). Floor Storage umožňuje skladovat objekty typu flow item. Charakteristiky Floor Storage jsou: počet řádků, počet sloupců a jejich rozměry, počet objektů v jednom slotu a strategie naskladňování předmětů. Kapacita Floor Storage je definována počtem řádků a sloupců a rozměry: výška objektů skládaných na sebe se nastavuje zadáním velikosti *z-ové souřadnice*. Floor Storage je i s naskladněnými boxy možné vidět na obrázku 6.8.



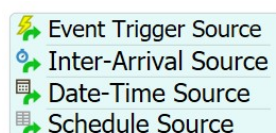
Obrázek 6.8: Floor Storage naplněná boxy

6.4 Popis základních použitých objektů process flow

6.4.1 Source

Source je zdrojem tokenů v proces flow. Ve FlexSimu můžeme najít čtyři typy objektů source, jejich grafická reprezentace je vidět na obrázku 6.9.

- Event-triggered source: generuje token na základě události v modelu (například příchod flow itemu do queue).
- Inter-Arrival source: generuje tokeny na základě manuálně zadaného vstupu (například na základě statistického rozdělení, pevně zvoleného času apod.).
- Date-Time source: generuje token dle zadaného data a času.
- Schedule Source: generuje token dle tabulky.



Obrázek 6.9: Jednotlivé typy objektu source

6.4.2 Assign Label

Tento objekt umožňuje vytvořit a přiřadit procházejícímu tokenu label.



Obrázek 6.10: Objekt Assign Label

6.4.3 Create Tokens

Objekt Create Tokens vytváří při průchodu tokenu další tokeny ve vybraném objektu process flow. Vytvořené tokeny mohou být Child Tokens, Siblings Tokens nebo Independent Tokens. Při tvorbě modelu byly použity Child Tokens, které jsou navazány na token, který je vytvořil a v případě změny labelu tohoto tokenu se tato změna propíše i do vytvořených tokenů. Tyto tokeny zároveň přebírají všechny labely jejich mateřského tokenu.



Obrázek 6.11: Objekt Create Tokens

6.4.4 Custom Code

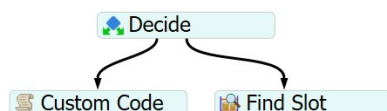
Custom umožňuje vytvořit libovolný kód v programovacím jazyce flexscript, nebo použít některé předvytvořené kódy, například pro zápis hodnot do tabulky.



Obrázek 6.12: Custom Code

6.4.5 Decide

Tento objekt umožňuje v process flow vytvořit rozhodovací podmínku. Nastavení podmínek je poměrně široké, autor pro tvorbu modelu ale používal výhradně tzv. Conditional Decide, které token pouští do příslušných konektorů na základě splnění dané podmínky.



Obrázek 6.13: Objekt Decide a jeho spojení s navazujícími objekty

6.4.6 Delay

Delay, neboli zpoždění, umožňuje do modelu přidat časové zpoždění před dalším průchodem tokenu skrze další objekty Process Flow.



Obrázek 6.14: Delay

6.4.7 Task Sequences: Load, Unload, Travel

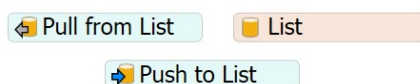
Objekty Load a Unload v modelu spouští proces vyložení a naložení pro definovaného task executora. Objekt Travel pak umožňuje pohybovat task executorem na definovanou pozici (může být zadána i v podobě souřadnic).



Obrázek 6.15: Load, Unload a Travel

6.4.8 Shared Assets: List, Push to List, Pull from List

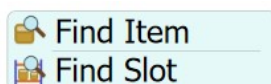
Tyto objekty umožňují přesunout token do listu (vytvoří se zde jeho instance) a nebo token z listu naopak "vytáhnout". Vytažení pak může probíhat na základě dotazu (query) zapsaném v databázovém jazyce SQL.



Obrázek 6.16: Objekty List, Push to List a Pull from List

6.4.9 Warehousing: Find Item, Find Slot

Objekt Find Item umožňuje najít objekt, který je naskladněn ve 3D objektu typu storage (např. floor storage, rack). Find Slot naopak objektu přiřazuje příslušnou skladovací pozici (slot) ve skladu. Příslušný Item nebo Slot lze najít dle dotazu zapsaném v jazyce SQL obdobně jako v Shared Assets



Obrázek 6.17: Find Item a Find Slot

Kapitola 7

Vymezení projektu

Zadaní simulačního problému je následující: vytvořit modely pro oba skladové systémy (regálový sklad a Autostore sklad) v softwaru FlexSim a porovnat jejich vlastnosti a charakteristiky pomocí simulační studie. Pro lepší srovnatelnost obou skladů se uvažují následující charakteristiky obou skladů:

1. Jedná se o výdejní sklad, kde se velikost požadavku na vyskladnění (objednávky) pohybuje mezi jedním až šesti druhy zboží. Pravděpodobnost počtu zboží v požadavku je následující:
 - a. Pravděpodobnost 15%, že bude jeden druh zboží v objednávce.
 - b. Pravděpodobnost 25%, že budou dva druhy zboží v objednávce.
 - c. Pravděpodobnost 30%, že budou tři druhy zboží v objednávce.
 - d. Pravděpodobnost 20%, že budou čtyři druhy zboží v objednávce.
 - e. Pravděpodobnost 10%, že bude pět druhů zboží v objednávce.
2. Základní skladovací jednotkou je Autostore box ve velikosti 425 mm. Vnější rozměry ve formátu **šířka, délka, výška** jsou 449 mm, 603 mm, 425 mm.
3. Kapacita skladu je 2470 boxů, každý box obsahuje zboží jednoho druhu.
4. Příchod objednávky do systému je dán normálním rozdělením o střední hodnotě $\mu = 45s$ a rozptylu $\sigma = 15s$.

Pro věrohodné porovnání obou skladů budou následující parametry uvažovány stejné pro oba dva případy:

1. velikosti skladu
2. pravděpodobnosti velikosti požadavku
3. velikosti úložných jednotek (AutoStore boxů)
4. pravděpodobnostní rozdělení výskytu požadavků

Autor se rozhodl uvažovat dvě základní strategie skladování:

1. Náhodné rozmístění úložných jednotek (boxů)

2. Uložení dle obrátkovosti zboží

Druhá strategie předpokládá uložení boxů se zbožím dle jejich obrátkovosti. Obrátkovost charakterizuje míru poptávky po konkrétním typu zboží. Celkově bylo určeno pět "obrátkovostních" tříd (1, 2, 3, 4 a 5, kdy po zboží typu 5 je poptávka nejvyšší). V AutoStore modelu bude skupina 5 umísťována na vrchní pozice skladu, skupina 4 na sloty pod nimi a tak dále. V regálovém skladu bude skupina 5 umísťována na pozici co nejbližší k výdejním pracovištím.

Pro porovnání výsledků byly vybrány simulační metriky uvedené v tabulce 7.1.

METRIKY SIMULAČNÍ STUDIE		
č.	Metrika	Jednotka
1	Počet odbavených požadavků za jednotku času	<i>požadavek \ čas</i>
2	Čas potřebný na odbavení určitého počtu požadavků (zakázek)	<i>minut</i>
3	Využitá plocha	m^2
4	Vytížení zdrojů (pracovníci/roboti/pracoviště)	%
5	Průměrný čas požadavku na pracovišti	<i>minut</i>
6	Vzdálenosti ujeté roboty a nachozené pracovníky	<i>metr</i>

Tabulka 7.1: Soupis relevantních metrik

Kapitola 8

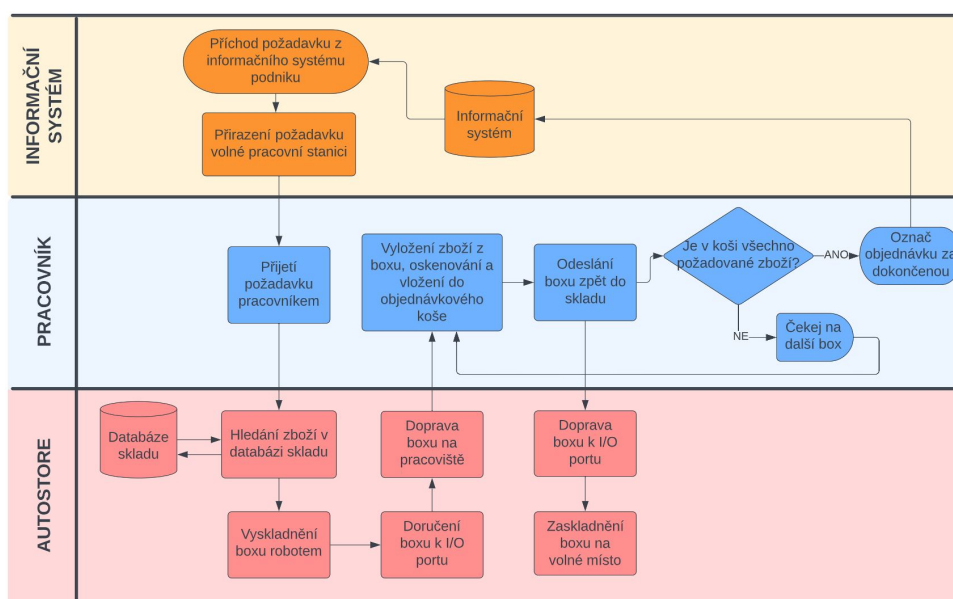
Koncepční model a sběr dat

V této kapitole jsou popsány základní vlastnosti koncepčního modelu a data, které bylo nutné zjistit pro tvorbu reality blízcího se modelu. Kapitola je strukturována do třech podkapitol: definice simulovaného procesu, rozměry a konfigurace skladů a sběr dat. Definice simulovaného procesu popisuje princip skladu AutoStore a regálového skladu. Sekce rozměry a konfigurace skladů se zabývá prostorovým uspořádáním skladů, včetně jejich rozměrových charakteristik. V sekci sběru dat jsou pak uvedeny údaje, které pro tvorbu modelu bylo nutné dohledat či určit.

8.1 Definice simulovaného procesu

Samotný proces začíná vznikem požadavku, ten reprezentuje např. objednávku zákazníka, požadavek na vyskladnění z předešlého procesu výroby, požadavek na vyskladnění objednávky pro kompletační pracoviště apod. Tento požadavek vzniká v podnikovém informačním systému a dále pak putuje do simulovaného skladu. Ve skladu je pak nalezeno požadované zboží (může být i více druhů) a dále dojde k jeho vložení do objednávkového koše. Následně je zboží odesláno na další pracoviště (např. pracoviště balení) Tato studie se zabývá simulací procesu výdeje zboží ze skladu. Tento proces popíšeme zvlášť pro jednotlivé typy skladů:

AutoStore: Systém AutoStore se skládá ze dvou základních částí: odbavovací pracoviště (nebo také odbavovací stanice) a robotizovaný sklad. Odbavovací pracoviště obsahuje počítač, kde pracovníkovi přijde požadavek na kompletaci objednávky. Pracovník požadavek přijme a roboti začnou vyskladňovat boxy s požadovaným zbožím, odvezou je k I/O portu, odkud se pak box dostane na odbavovací stanici . Zde pracovník vyjme z boxu požadovaný předmět, oskenuje jeho QR-kód a vloží jej do objednávkového koše. Boxy, ze kterých pracovník vyndal požadovaný předmět(y), pak putují zpět do I/O portu a jsou roboty naskladněny zpět do skladu. Tento proces se opakuje do té doby, dokud pracovník nevloží všechny požadované předměty do objednávkového koše. Po kompletaci box odešle a proces končí, mapu celého procesu můžeme vidět na obrázku 8.1.



Obrázek 8.1: Mapa procesu vyskladnění zboží ze skladu AutoStore v notaci Proces Flow Chart

Regálový sklad: u regálového skladu je z hlediska procesu situace obdobná jako u AutoStore systému, jediným rozdílem je, že box s požadovaným zbožím zde nevyhledávají roboti, ale pracovníci. Pracovník po příchodu požadavku putuje s objednávkovým košem do skladu, kde postupně nabírá požadované zboží z příslušných boxů. Po kompletaci se pak vrací zpět na své pracoviště, kde předměty oskenuje, přendá do objednávkového koše a odešle.

8.2 Rozměry a konfigurace skladů

Sklad AutoStore již byl popsán v teoretické části v kapitole 3, proto zde nebude již znovu vysvětlován jeho princip. Pro koncepční model je ale třeba specifikovat některé významné náležitosti tohoto skladu.

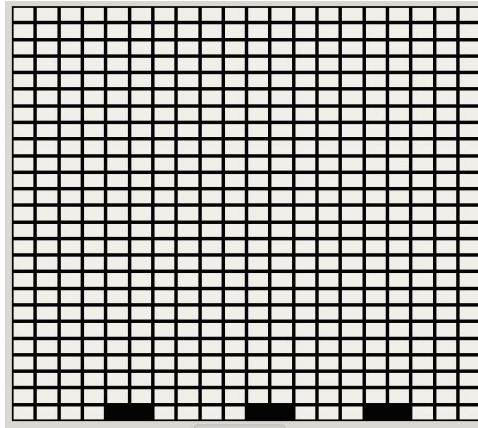
Pro uskladnění 2470 boxů se autor rozhodl zvolit AutoStore sklad o velikosti:

1. 25x20 skladovacích pozic pro sklad o počáteční výšce pěti boxů
2. 16x16 skladovacích pozic pro sklad o počáteční výšce deseti boxů

Je nutné ale část skladovacích pozic vyhradit I/O (*Input/Output*) portům. Pro jedno I/O pracoviště byly rezervovány celkem 2 pozice, jedna pozice představuje výtah pro input pozici, druhý pro output pozici. Celkově bylo rezervováno 6 skladovacích pozic (slotů), je tedy možné použít až tři pracoviště obsluhy. Schéma AutoStore skladu je vidět na obrázku 8.2, kde nevyplněné obdélníky představují sloty a černě začerněné obdélníky jsou sloty rezervované výdejními stanicemi.

V input pozici robot vyloží požadovaný box, výtah box odveze do spodní části skladu, kde je pak dopraven dopravníkem na pracoviště obsluhy (dále v

práci nazývané jako odbavovací stanice nebo stanice pro odbavení požadavků), kde jsou z boxu vyndány potřebné předměty. Poté je druhým dopravníkem dopraven na output pozici, kde box výtah vyzdvihne nahoru do horní části skladu, kde ho převezme volný robot a zaskladní ho zpět do skladu.



Obrázek 8.2: Schéma skladu typu AutoStore

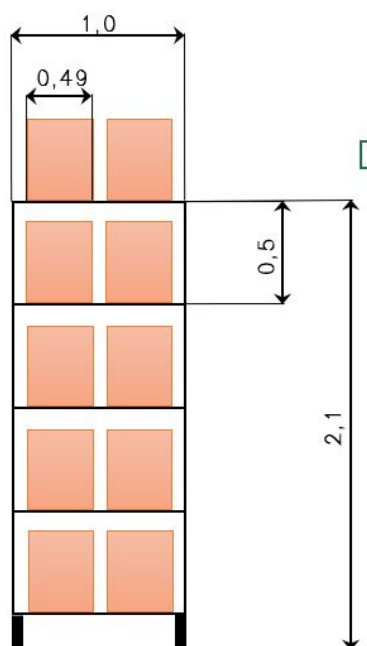
U regálového skladu je situace jednodušší, zde byl vytvořen takový počet regálů o takové délce, aby splňoval zadaný počet boxů. Bylo rozhodnuto použít 10 regálů o délce 25 metrů. Parametry jednoho regálu jsou vidět na obrázku 8.3. Napříč regály byly vytvořeny průchody pro snazší pohyb zaměstnanců mezi jednotlivými uličkami. Schéma regálového skladu je pak možné vidět na obrázku 8.4.

8.3 Sběr dat

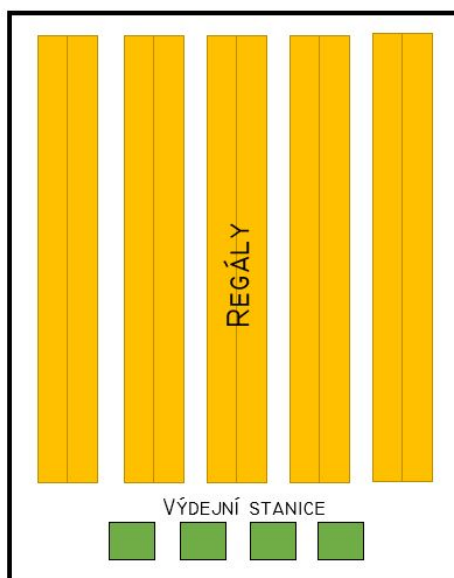
Pro správný a realitě se blížící model bylo třeba vyhledat několik důležitých údajů. Důležité hodnoty, které bylo třeba zjistit, jsou:

1. Rychlosti a zrychlení AutoStore vozíčků.
2. Rychlosti a zrychlení pracovníků.
3. Rychlosti vykládání a nakládání AutoStore vozíčků.
4. Průměrná doba naložení předmětu z boxu (pro pracovníka i AutoStore vozítko).
5. Průměrná doba vyložení předmětu z boxu, naskenování a vložení do objednávkového koše.
6. Rychlost AutoStore výtahů.

Rychlosti jednotlivých pracovníků jsou již integrovány v simulačním softwaru FlexSim Express. Všechny potřebné informace pro sklad AutoStore se daly najít na webových stránkách výrobce. Výrobce udává 3 různé typy robotických



Obrázek 8.3: Výkres regálu, velikosti udány v metrech. Regály jsou řazeny těsně vedle sebe.



Obrázek 8.4: Schéma regálového skladu.

vozítek, pro tuto práci bylo zvoleno vozítko modelového typu R5 (na obrázku 8.5). Charakteristiky tohoto vozítka jsou pak shrnuty v tabulce 8.1.

Pro určení údajů 4. a 5. (vyložení a naložení předmětu do a z boxu) autor provedl vlastní měření. Průměrná doba naložení předmětu do boxu byla určena jako 7,1s. Průměrná doba vyložení předmětu z boxu, skenování a



Obrázek 8.5: Robotické vozítko R5 od společnosti AutoStore

Parametr	Jednotka
Délka	<i>963 mm</i>
Šířka	<i>700 mm</i>
Výška	<i>545 mm</i>
Zrychlení	$0.8 \text{ m} \cdot \text{s}^{-2}$
Rychlost	<i>3.1 m/s</i>
Rychlost zdvihu	<i>1.6 m/s</i>

Tabulka 8.1: Parametry vozítka R5

vložení do objednávkového koše pak byla určena jako 10,0 s.

Kapitola 9

Tvorba modelu

9.1 Výběr simulačního software

Autor této simulační studie použil jako simulační software program FlexSim 2021 v licenci Express (zkušební licence omezená počtem objektů). Popis tohoto softwaru byl již uveden v kapitolách 2.3.4 a 6. Tento simulační software byl vybrán z toho důvodu, že zkušební licence je časově neomezená a autor se s tímto softwarem setkal již ve výuce. Jedná se zároveň o jeden z nejpoužívanějších nástrojů pro tvorbu simulací a velkou výhodou je také jeho uživatelská komunita, kde lze získat velmi rychle odpověď na daný problém.

9.2 Tvorba modelu AutoStore

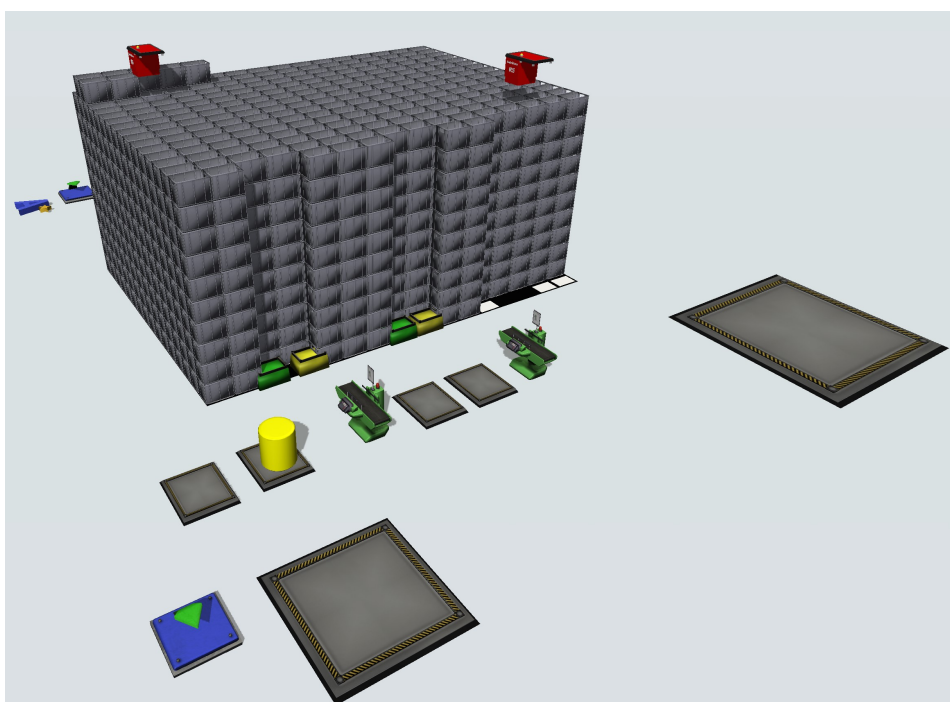
Při tvorbě modelu můžeme vyjít z definice procesu v podkapitole 8.1 a obrázku 8.1. Simulační model se skládá ze 3D objektů a Process Flow. Vytvořený model AutoStore je zobrazen na obrázku 9.1 .

9.2.1 3D objekty pro Autostore

Model tvoří několik významných 3D objektů nebo skupin 3D objektů, které spolu vzájemně komunikují. Jedná se o:

- generování požadavků (source)
- fronta požadavků (queue)
- staniceX_požadavky (queue)
- sklad (skupina objektů)
- staniceX (processor)
- odbavené požadavky (queue)
- A*StarNavigator (zamezuje kolizím mezi roboty, viz 6.3)

Vzájemné interakce mezi objekty je pak možné vidět na UML sekvenčním diagramu (obrázek 9.2).



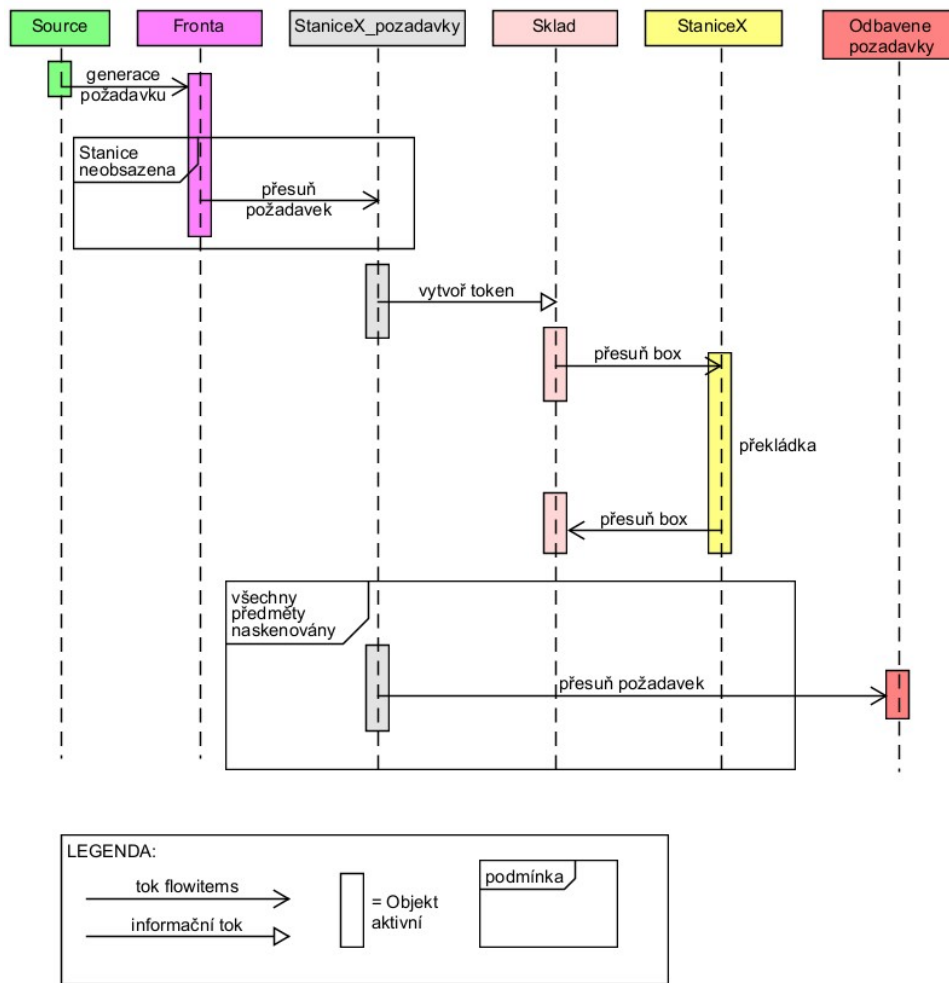
Obrázek 9.1: Vytvořený model AutoStore v softwaru FlexSim

■ Generování požadavků

Objekt **Source** (viz. podkapitola 6.3) generuje fyzický objekt (pro tento případ byl zvolen žlutý válec) dle příslušného statistického rozdělení. Žlutý válec reprezentuje požadavek, který přišel z informačního systému. Požadavky jsou pak v tomto objektu generovány ze sešitu programu MS Excel, kde se generují náhodná čísla a z nich pak příslušné hodnoty pro definované normální rozdělení. Blíže je toto popsáno v části věnující se experimentům.

■ Fronta požadavků

Požadavky ze Source putují do objektu Fronta (queue, viz. 6.3), ze kterého pak putují na pracoviště překládky (zde je přeložen předmět z AutoStore boxu do objednávkového koše), pokud je pracoviště volné. Při nastavení Fronty byla použita funkce přiřazení výstupu **Round Robin if Available**. To znamená, že fronta bude požadavek (reprezentovaný žlutým válcem) posílat střídavě na jednotlivá pracoviště, pokud to bude možné. Například pokud se ve frontě objeví první požadavek, přejde automaticky na pracoviště č.1, to požadavek zpracuje a znovu se uvolní. Poté přijde do systému další požadavek, který ale přejde na pracoviště č.2, i když je pracoviště č.1 volné, protože se queue snaží jednotlivé výstupy prostrědat. Kdyby ale bylo pracoviště č.2 blokováno, požadavek by opět putoval na č.1. Pokud by tato funkce nastavena nebyla, požadavek by putoval na první pracoviště a až, pokud by toto první pracoviště bylo blokováno, by putoval na další přiřazené pracoviště. Typ obsluhy Round Robin if Available byl zvolen z toho důvodu, že se používá v průmyslové praxi,



Obrázek 9.2: Sekvenční diagram simulačního modelu AutoStore pro vybrané objekty

aby byla pracovní zátěž mezi pracovníky rozdělena rovnoměrně.

■ StaniceX_pozadavky

Obsazení dané stanice v abstraktním smyslu představuje queue nazvaná jako **StaniceX_pozadavky**, kde X je číslo dané odbavovací stanice. Queue má nastaven maximální počet uskladněných objektů na jeden, jelikož daná odbavovací stanice může zpracovávat v jednu chvíli pouze jeden požadavek. Tato queue má přiřazen process flow, který řídí proces vyskladňování jednotlivých boxů ze skladu. Tento process flow je popsán níže.

Po příchodu požadavku na pracovní stanici se v process flow vygeneruje token a roboti začnou vyskladňovat příslušné boxy.

■ Sklad

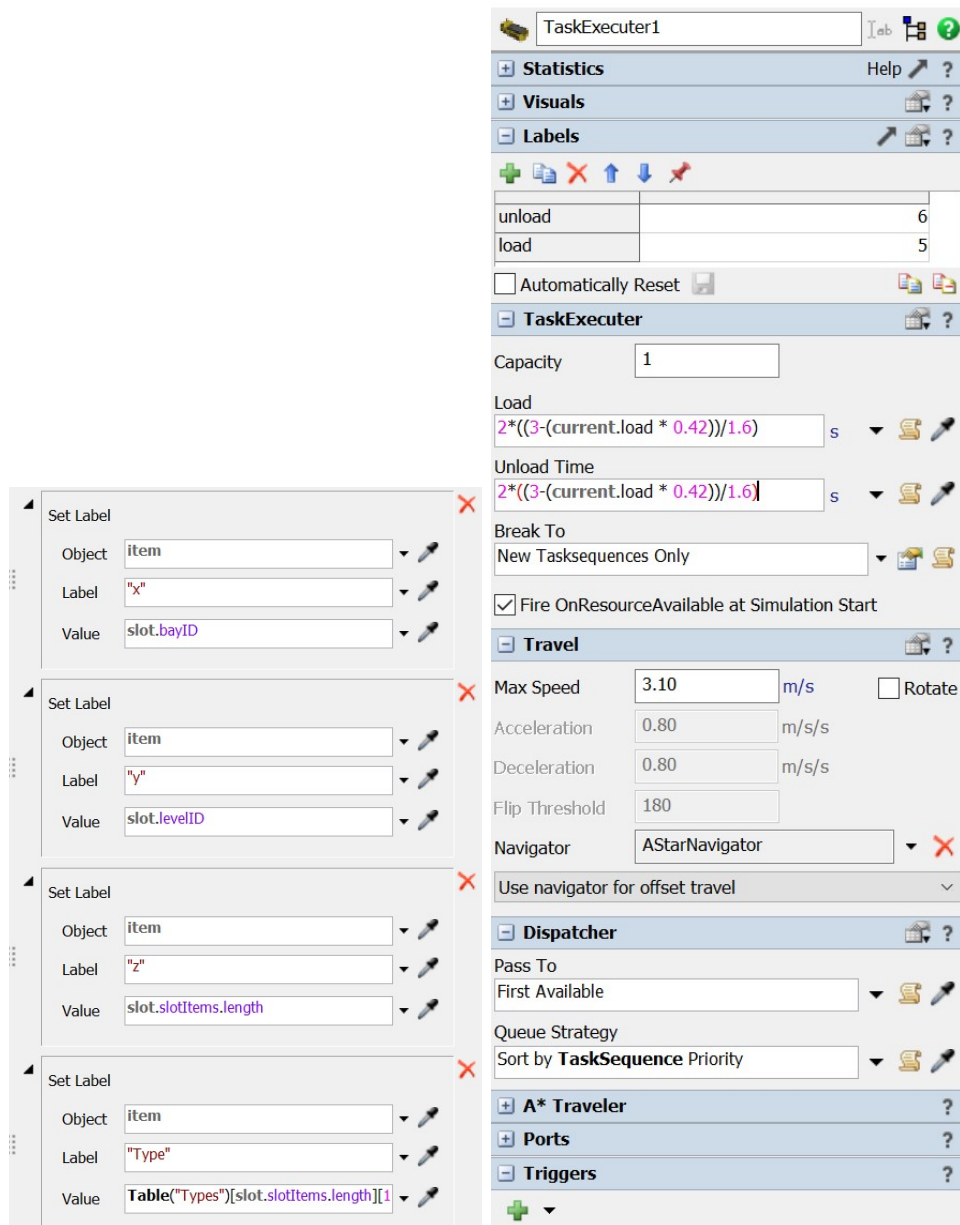
Sklad se skládá z několika objektů. Vlastní skladovací prostor je tvořen Floor Storage. Její velikost je 25×20 skladovacích slotů (jeden slot má obdélníkový tvar o rozměrech $0.48 \text{ m} \times 0.68 \text{ m}$ tak, jak bylo specifikováno v zadání). Pro případ AutoStore skladu o výšce 10-ti boxů pak bylo uvažováno 16×16 skladovacích pozic (slotů). Boxy jsou vytvořeny v Source, který je vazbou spojen s Floor Storage. Generování je nastavené na čas 0, tedy po spuštění simulace dojde k vytvoření daného počtu boxů, které se naskladní do Floor Storage. Jakmile do FloorStorage vstoupí box, spustí se trigger, který boxu přiřadí souřadnice x, y a z. Pomocí těchto souřadnic je pak možné identifikovat box ve skladišti a je tak přesně zdefinována jeho pozice. Přiřazení triggerů je možné vidět na obrázku 9.3.

O vyložení a naložení boxu se starají Task Executors (viz. 6.3). Task Executors jsou v process flow reprezentovány pod objektem Resource. Task Executors mají vytvořeny vlastní design ve stylu vozítka R5. Jejich nastavení je vidět na obrázku 9.3. Labely load a unload, které je možné vidět na obrázku 9.3, určují z-ovou pozici vyzvedávaného boxu. Tato pozice nabývá celočíselné hodnoty a charakterizuje pořadí vyzvedávaného boxu ve sloupci, např. $z=3$ znamená, že box je na třetí pozici, pod ním se tedy nachází ještě další dva boxy. Z-ová pozice vyzvedávaného boxu se pak používá pro výpočet času vyzvednutí či vyložení boxu. Výpočet času je pak možné vidět na obrázku 9.3 v podoknu TaskExecutor.

Sklad je dále tvořen vstupním a výstupním I/O portem (ty reprezentuje objekt queue). Do vstupního queue robot vyloží box a ten pak putuje na pracoviště překládky (vyjmutí předmětu z boxu). Do výstupního portu (queue) naopak putují boxy z pracoviště překládky (ze kterých již byl vyložen požadovaný předmět). Jakmile do výstupního portu přijde odbavený box, v process flow Zaskladnění se vygeneruje token. Tento process flow aktivuje příjezd robota ke výstupnímu portu, naložení odbaveného boxu a jeho zaskladnění zpět do skladu.

■ Stanice_X

Překládku zboží z boxu do objednávkového koše modeluje objekt **Stanice_X** (processor, viz 6.3), kde X reprezentuje pořadové číslo odbavovací stanice. Nastavení času procesu pak odpovídá definovaným časům překládky ze sběru dat. Jelikož odbavovací stanice AutoStore skladu jsou obvykle konstruovány jako revolverové (disponují zásobníkem pro další boxy, které teprve budou zpracovávány), bylo toto promítnuto i v modelu s využitím pomocné queue umístěné před processorem. Ta umožňuje skladovat až dva další boxy. Po zpracování všech boxů v procesoru se požadavek (celá objednávka, která reprezentována žlutým válcem) přesune do queue **Odbavene_pozadavky** (definováno v process flow). StaniceX_pozadavky se tak uvolní a je připravena přijmout další požadavek.



Obrázek 9.3: Vlevo: přiřazení souřadnic boxu po vstupu do FloorStorage. Vpravo: nastavení Task Executors (AutoStore vozítek)

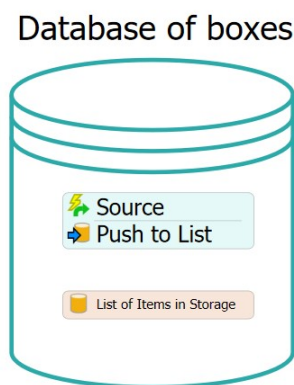
9.2.2 Process flow pro AutoStore

V předchozí části uvedené 3D objekty reprezentují fyzické vlastnosti systému a modelují procesní tok reálných objektů. Aby se ale mohl model chovat tak, jako se chová reálný AutoStore systém, je třeba ještě naprogramovat chování robotů a vyskladňování a naskladňování boxů. Program, který má toto na starost, byl vytvořen za použití Process Flow v softwaru FlexSim. Problém Autostore skladu byl rozdělen do třech subprocesů:

- Generování databáze
- Vyskladnění boxu
- Zaskladnění boxu

■ Generování databáze:

Je spuštěno kdykoliv, kdy box vstoupí do FloorStorage. Schéma proces flow je vidět na obrázku 9.4. Source představuje Event-Trigger zdroj tokenů (při vstupu boxu do Floor Storage se vygeneruje token). Token pak putuje do push-to-list, ten má na starosti "posunout" token do global listu s názvem ItemList (zde je pak možné z tohoto listu tento token v rámci procesu vyskladnění "vytáhnout").



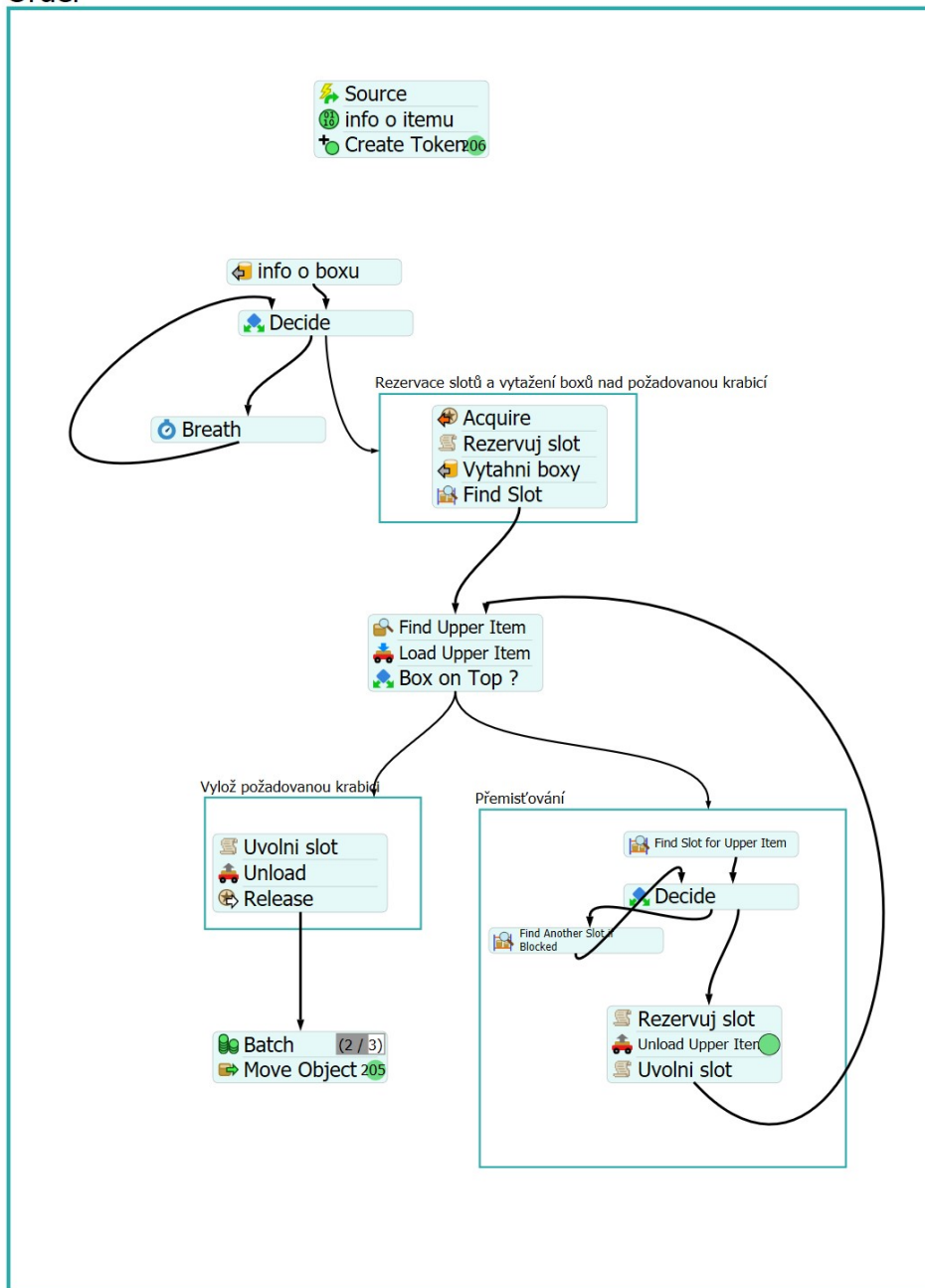
Obrázek 9.4: Process Flow pro generování databáze boxů

■ Vyskladnění boxu:

Je realizováno pomocí process flow přiřazeným objektu StaniceX_pozadavky. Tento proces flow je vidět na obrázku 9.5. Zjednodušený princip modelu AutoStore je možný popsat následujícím algoritmem, podrobný popis je pak popsán v dalších odstavcích.

1. vytvoř token po příchodu požadavku
2. vytvoř tokeny dle pravděpodobnosti velikosti objednávky
3. načti informace o požadovaném boxu
4. rezervuj slot požadovaného boxu
5. načti informace o slotu požadovaného boxu
6. nalož vrchní box
7. Je vrchní box nahoře ?
ANO: přejdi na krok 11
NE: přejdi na krok 8
8. Najdi neobsazený slot
9. Vylož box na neobsazený slot
10. vrať se na krok 6

Order

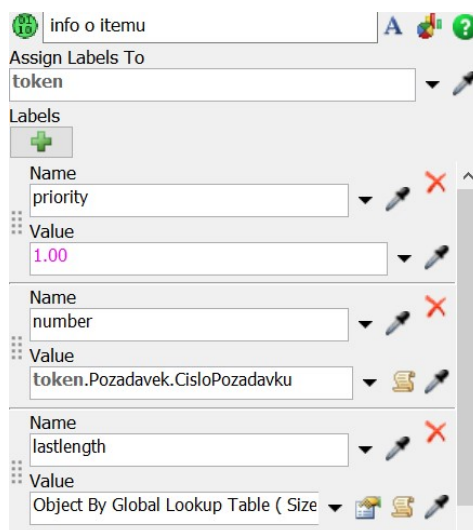


Obrázek 9.5: Process flow vyskladnění požadavku

11. vylož požadovaný box do příslušného output portu

Když požadavek přijde na **StaniceX_pozadavky**, automaticky se v process flow vygeneruje token. Ten putuje do **Assign Labels** nazvaném info o itemu, kde dochází k načtení několika důležitých labelů. Tyto labely je možné vidět na obrázku 9.6. Label priority definuje, které tokeny budou upřednostněny v **Acquire Resource**. Upřednostňovány jsou tokeny, které se

starají o zaskladnění boxů zpět do skladu (tedy tokeny procházející subprocessem Zaskladnění boxu). Token number přebírá z flow itemu reprezentující požadavek (žlutý válec) příslušné číslo požadavku. Tento label je důležitý v objektu Create Tokens pro generování náhodných čísel reprezentujících velikost požadavku. Stejnou funkci má i token lastlength, který se pak uplatňuje v generování příslušného typu zboží v požadavku (pro seříděné varianty skladu).



Obrázek 9.6: Důležité labely přiřazené v objektu info o itemu

Token dále putuje do **Create Tokens**, kdy spuštění této aktivity vygeneruje další tokeny, které reprezentují počet předmětů v objednávce. Tyto nové tokeny se generují v objektu **info o boxu** a budeme je nazývat jako *Child Tokens*. Create Tokens má některé důležité vlastnosti a labely, které souvisí převážně s generování náhodných čísel z rozdělení definujících velikost objednávky, a požadovaný typ zboží dle obrátkovosti (pro seříděné sklady, více viz. experimentální část). Nastavení objektu Create Tokens můžeme vidět na obrázku 9.7. V nastavení této aktivity je nutné vyplnit údaj o počtu tokenů, které se mají vygenerovat (položka Quantity, viz. obr. 9.7). Počet nově vygenerovaných tokenů, reprezentujících velikost požadavku, se vezme z tabulky "*SizeofOrder*" (zobrazena na obrázku 9.7 vpravo) ve sloupci *Velikost*, který definuje velikost požadavku. Řádek, ze kterého se má vzít příslušná velikost požadavku, definuje label "*number*", který charakterizuje pořadí požadavku v čase.

Obdobně je nově vzniklým tokenům přiřazen i label *typ*, který se vezme ze sloupce *Typ* v tabulce na obrázku 9.7. Label *typ* charakterizuje příslušný typ vyskladňovaného předmětu. Příslušný řádek, charakterizující typ předmětu, se vypočte jako:

$$\text{token.children.length} + \text{token.lastlength}$$

kde `token.children.lenght` udává pořadí vygenerovaného "*child tokenu*" (např. v pořadí druhý vygenerovaný "*child token*" bude mít pořadové číslo 2)

	Velikost	Typ	Vel.předchozí
1	3	5	0
2	5	5	3
3	3	2	8
4	3	4	11
5	1	4	14
6	4	5	15
7	3	3	19
8	5	4	22
9	1	4	27
10	3	5	28
11	3	5	31
12	4	4	34

Obrázek 9.7: Vpravo: objekt Create Tokens. Vlevo: tabulka SizeofOrder která slouží pro přiřazení externě vygenerovaných (v MS EXCEL) hodnot z náhodných rozdělení pro velikost požadavku a typ požadovaného boxu.

a token.lastlenght udává počet vygenerovaných child tokenů v celém průběhu simulace. Tato hodnota se vezme z tabulky *SizeOfOrder* (obr. 9.7) ze sloupce *Vel.předchozí*.

Tokeny vygenerované v Child Tokens putují do objektu Pull from List nazvaném **info o boxu**, kde dochází k pomyslnému vytažení požadovaných tokenů reprezentujících příslušné boxy ve skladu AutoStore. Tyto tokeny však zatím zůstávají v **ItemListu**, načítají se pouze informace o jejich souřadnicích (**Assign Labels**).

V **Decide** pak dochází k rozhodnutí, zda je pro robota daný slot volný, tedy zda ve slotu nedochází k vyskladňování jiného boxu. To by mohl být problém a působilo by to zmatek mezi roboty. Pokud je slot obsazený, token se posune do delay **Breath**, kde je nastavena doba čekání 1 sekunda a poté token opět putuje do Decide. Kdyby totiž token z Decide putoval rovnou zpět do Decide, došlo by k neustálému vykonávání dotazu na volný slot a to by mělo za následek výrazné zpomalení či zaseknutí programu.

Pokud je slot volný, v **Acquire** dochází k rezervaci robotického vozítka a následně v **Rezervuj slot** dochází k rezervaci příslušného slotu. Toho je dosaženo pomocí tabulky, která v podstatě reprezentuje půdorys skladu, kde

v daných buňkách (reprezentujících sloty) jsou nuly. V případě obsazení slotu dochází k zapsání jedničky do příslušné buňky (slotu).

V objektu **Vytahni boxy** dochází ke skutečnému vytažení tokenů reprezentujících příslušný box k vytažení a všechny další boxy nacházející se nad ním. Tím je z databáze vytažen celý sloupec a není ho tak možné již vytáhnout, pokud by například přišel paralelně další požadavek na vyskladnění boxu umístěného nad již vyskladňovaným boxem. Token pak dále prochází objektem Find Slot, kde je načtena pozice slotu přiřazená požadovanému boxu.

Dále pak v process flow dojde k aktivaci objektů: **Find Upper Item**, **Load Upper Item** a **Box on Top ?**, které slouží k naložení vrchního boxu a určení, jestli je vrchní box boxem požadovaným nebo ne. Pokud ano, pak tokeny putují do sub-procesu **Vylož požadovaný box**, který slouží k vyložení boxu do příslušného I/O portu.

Pokud vrchní box není požadovaným boxem, poté token putuje do subprocesu přemisťování, kdy dojde k vyložení boxu na pozici vzdálenou maximálně o 2 sloty a s nejnižší hodnotou souřadnice z. Query (SQL dotaz), který má toto v objektu **Find Slot for Upper Item** na starost, vypadá následovně (pro případ AutoStore skladu o výšce 10ti boxů):

```
WHERE slot !=$1.slot
AND slot.bayID < ($1.pulled.Item.x+3)
AND slot.bayID > ($1.pulled.Item.x-3)
AND slot.levelID < ($1.pulled.Item.y+3)
AND slot.levelID > ($1.pulled.Item.y-3)
AND slot.slotItems.length < 12
ORDER BY slot.slotItems.length ASC
```

Následně token putuje opět do objektu Decide a pokud daný slot pro vyložení boxu není obsazený, putuje dále do objektů Rezervuj slot, Unload Upper Item (vyloží box na příslušný slot) a Uvolni slot a poté zpět do **Find Upper Item**.

V případě, že je daný slot obsazený, je použit nový dotaz a model se opět vrací do Decide. Dotaz v objektu **Find Another Slot if Blocked** je zapsán ve tvaru:

```
WHERE slot.bayID !=$1.pulled.Item.x
AND slot.bayID < ($1.pulled.Item.x+6)
AND slot.bayID > ($1.pulled.Item.x-6)
AND slot.levelID < ($1.pulled.Item.y+6)
AND slot.levelID > ($1.pulled.Item.y -6)
AND slot.slotItems.length <12
ORDER BY RAND()
```

Pokud robot naloží požadovaný box (tedy token projde podmínkou definovanou v objektu **Box on Top?**, dojde k uvolnění slotu, vyložení požadovaného předmětu do příslušného output portu a uvolnění task executora. Dále pak

token putuje do **Batch**. Objekt **Batch** sloučí všechny tokeny stejné instance (všechny tokeny reprezentující předměty v daném požadavku). Batch čeká, dokud nedojde k nahromadění všech tokenů dané objednávky (vyskladnění všech boxů s požadovanými předměty) a teprve pak tokeny sloučí do jednoho a pustí dál. Sloučený token pak přejde do objektu **Move Object**, který posune válec (Flow Item) reprezentující požadavek do queue s odbavenými požadavky.

■ Zaskladnění boxu

Pro zaskladnění boxu byl vytvořen process flow na obrázku 9.8

Jakmile se box přesune na příslušný output port, dojde k vytvoření tokenu v **Source**. Následně jsou tokenu přiřazeny labely nutné pro uložení slotu ve **Find Slot** a také label **priorita**. Hodnota labelu prioritita je v případě subprocessu zaskladnění nastavena na 2. Objekt resource, který přiřazuje task executory pro jednotlivé tokeny, má nastavený dotaz na obrázku 9.9, který upřednostňuje tokeny s prioritou 2:

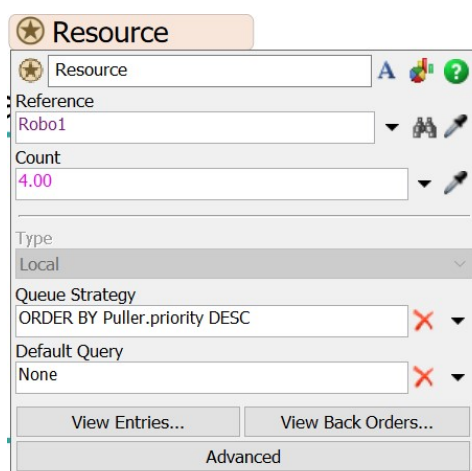


Obrázek 9.8: Process flow zaskladnění boxu zpět do skladu

Po přiřazení labelu je tokenu přiřazen robot, dochází k naložení boxu. Ve Find Slot je nalezen volný slot pomocí následující query:

```
ORDER BY slot.slotItems.length ASC
```

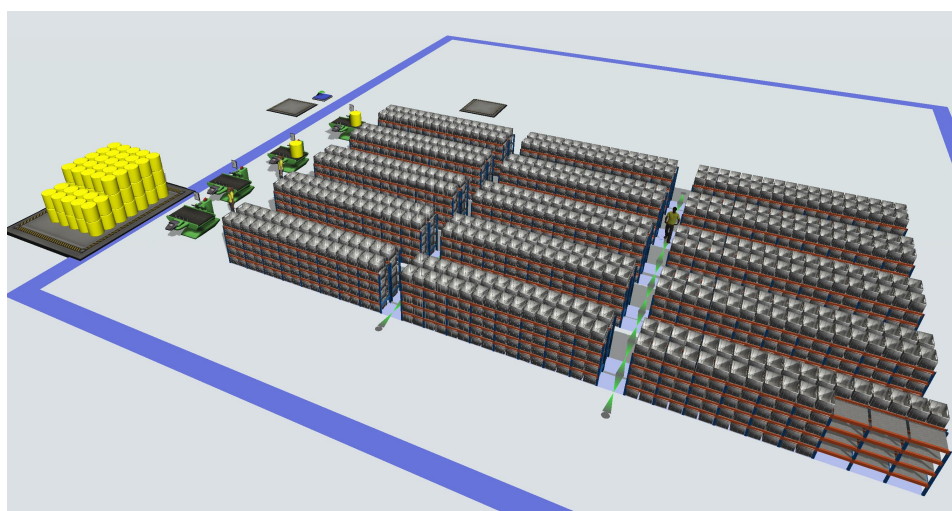
Po nalezení slotu je box vyložen pomocí objektu Unload na pozici volného slotu a následně je uvolněn přiřazený robot. Tím celý proces zaskladnění končí.



Obrázek 9.9: Nastavení objektu resource a prioritizování tokenů z subprocessu zaskladnění

9.3 Regálový sklad

Řešení regálového skladu je podobné jako u skladu AutoStore. Model se skládá ze 3D objektů a ProcessFlow. Grafická reprezentace vytvořeného regálového skladu je vidět na obrázku 9.10



Obrázek 9.10: Vytvořený model regálového skladu

9.3.1 3D objekty pro regálový sklad

Nastavení některých 3D objektů je stejné jako v případě modelu AutoStore. Mezi shodné objekty patří:

- Generování požadavků (source)

- Fronta požadavků (queue)
- Odbavovací stanice (processor)
- Generování boxů (queue) - zde je rozdíl, boxy jsou generovány v rámci process flow do queue, ze které pak jdou do skladu
- Odbavené požadavky (queue)
- A*navigator

Objekty, které přibily navíc oproti modelu AutoStore jsou:

- Pracovníci (task executor-operator)
- Regálový sklad (rack)

Tyto nové 3D objekty jsou pak podrobně vysvětleny v textu níže. Interkace mezi všemi 3D objekty modelu regálového skladu je možno vidět na grafu 9.11.

■ Pracovníci

Jsou reprezentováni objektem operator. Jedná se o objekt typu Task Executor. Každá pracovní stanice má přiděleného jednoho pracovníka, který po příchodu požadavku na pracovní stanici odejde do skladu a postupně za sebou z jednotlivých boxů bere požadované předměty. Tyto operace jsou naprogramovány pomocí process flow.

■ Regálový sklad

Regálový sklad je tvořen 10 regály. Na začátku simulace se v rámci process flow vygeneruje požadovaný počet 2470 boxů, které jsou přes pomocnou frontu naskladněny do regálového skladu. Každému boxu je při vygenerování přiřazeno ID číslo (v rámci labelu daného boxu), které jednoznačně identifikuje daný box.

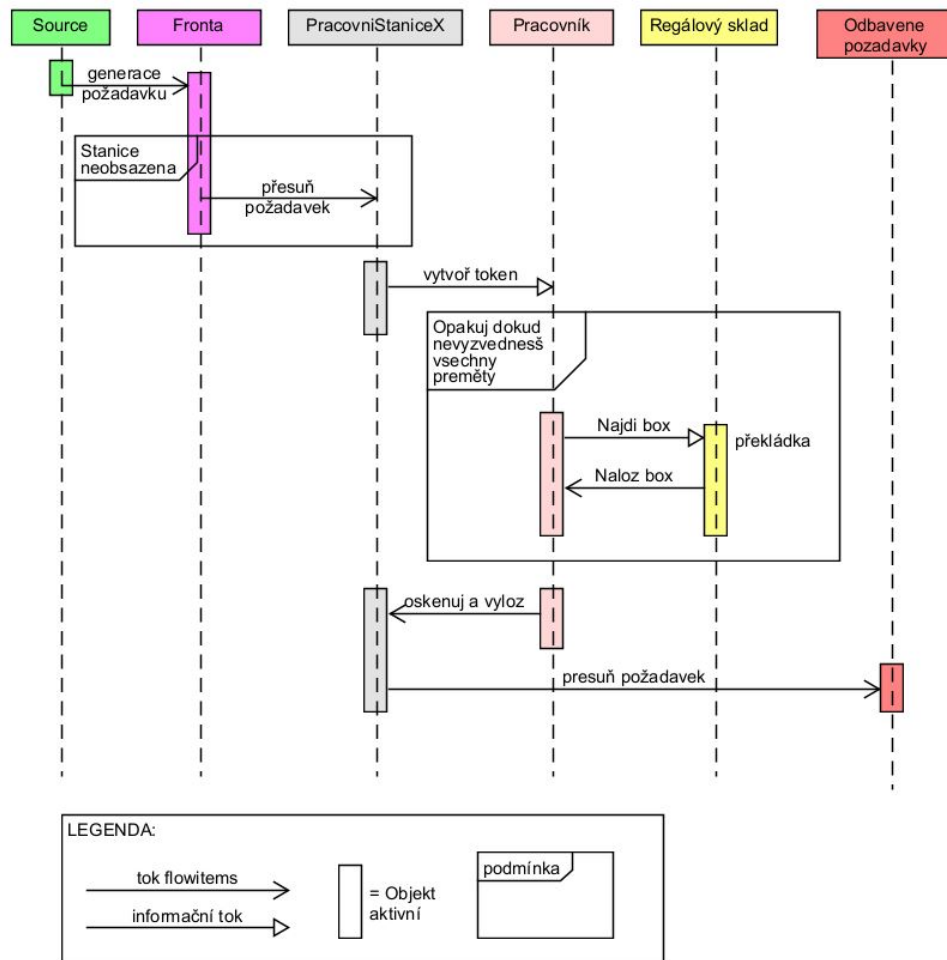
■ 9.3.2 Process flow

Process flow je v modelu regálového skladu rozdělen na:

- Generování boxů a databáze
- Odbavení požadavku

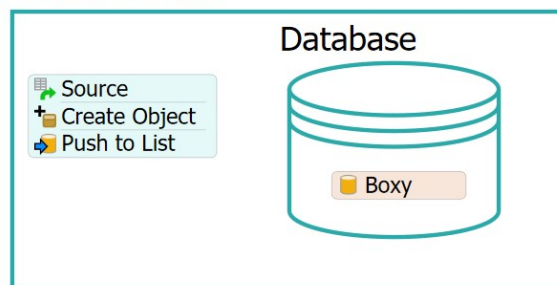
■ Generování boxů a databáze:

Generování databáze se děje pomocí process flow zobrazeného na obrázku 9.12. Source je zdroj tokenů dle definované tabulky. V tomto případě generujeme stejný počet tokenů jako boxů v regálovém skladu (2470 boxů). Create Object pak při průchodu tokenu vytvoří v queue generování boxů fyzické objekty boxů v modelu. Push to List pak má na starosti vytvořit databázi všech boxů ve skladu.



Obrázek 9.11: Sekvenční diagram pro objekty regálového skladu

Generace boxů



Obrázek 9.12: generování boxů a vytvoření databáze (ItemListu)

■ Odbavení požadavků:

Je process flow přiřazený objektu pracovní stanice. Struktura je vidět na obrázku 9.13. Tento process flow začíná ve chvíli, kdy na danou stanici dorazí požadavek a **prichod požadavku** vygeneruje token. Následně se v objektu

Create tokens vytvoří dceřinné tokeny (Child Tokens). Tyto tokeny se vytvoří v objektu **Pull from List**.

V **Assign Labels** je pak přiřazen label batch, který přiřadí přes mateřský token všem dceřinným tokenům informaci o počtu vytvořených dceřinných tokenů (child tokenů). To je důležité pro objekt **Batch**.

V objektu **Pull from list** dochází k vytáhnutí tokenů reprezentujících boxy dle náhodného pořadí (funkce ORDER BY RAND()) a dále jsou pak načteny informace o pozici tohoto předmětu. Do procesu **Vezmi předmět z boxu** může vstoupit pouze jeden token (viz. objekty **Enter Zone** a **Zone**), který představuje předmět k vyzvednutí.

Objekt **Travel** má přiřazeného operátora pro příslušnou pracovní stanici, které je proces flow přiřazen. Objekt (**Travel**) pak způsobuje pohyb pracovníka na místo kde se nachází požadovaný box (vytažený z **Pull from list**). Naložení požadovaného předmětu je abstraktně vyjádřeno zpožděním (**Delay**), tedy dobou, kdy pracovník nakládá požadovaný předmět. Token pak opouští zónu a umožňuje příchod dalšího tokenu. Pracovník se tedy může vydat naložit další předmět z boxu.

Objekt **Batch** pak na závěr bloku aktivit nazvaných Vezmi předmět z boxu (modrý obdélník na obrázku 9.13) zkompletuje všechny child tokeny charakterizující počet zboží v objednávce do jednoho tokenu a následně ho pošle do bloku aktivit **přeložení do koše**. Tento blok zahrnuje pohyb pracovníka zpátky na pracovní stanici (**Travel**) a dobu, která charakterizuje čas překládky předmětů do objednávkového koše (**Delay**). Na konci tohoto procesu je požadavek obslužen a přesune se do fronty odbavených požadavků pomocí **Move Object**.

9.4 Verifikace a validace

9.4.1 Verifikace

Verifikace modelu probíhala převážně pomocí vizuální kontroly výsledného modelu při kontrolní simulaci. K debugování programu byla použit compiler console a system console (na obrázku 9.14), které zobrazují chyby související s chodem programu při simulaci nebo chyby vzniklé při kompilaci kódu.

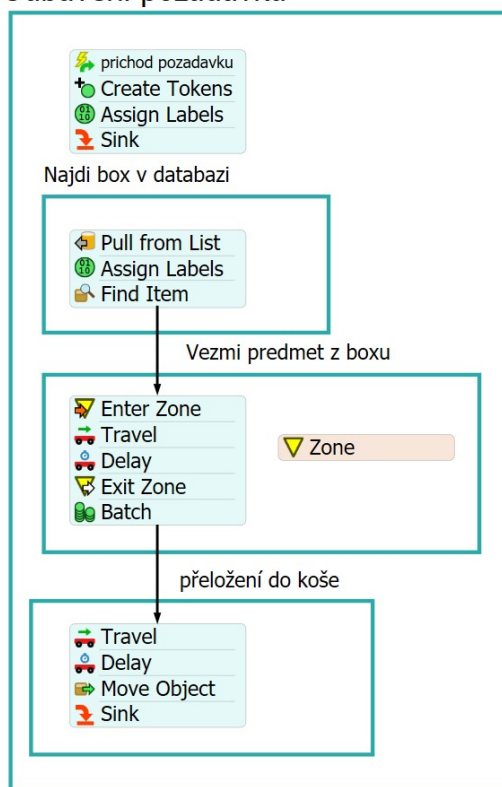
9.4.2 Validace

Model pro sklad AutoStore obsahuje omezení oproti reálnému systému, což je dáno zejména limitovaným počtem objektů process flow.

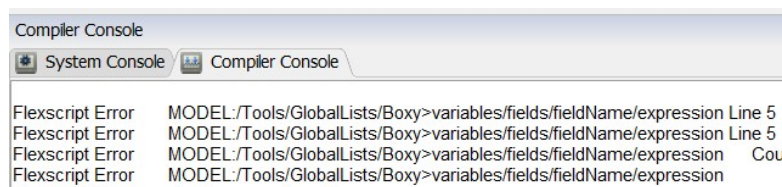
V reálném systému jsou ostatní boxy, které byly po naložení požadovaného boxu vlivem přemístování uloženy okolo, naskladněny zpět do příslušného slotu. Tato činnost by však v rámci process flow potřebovala vlastní subprocess, který byl s ohledem na nedostatek objektů v dané bezplatné verzi zanedbán.

Další omezením je skutečnost, že v subprocessu zaskladnění není respektováno, jestli je daný slot zabraný či nikoliv. Tento problém má velmi jednoduché

Odbavení požadavku



Obrázek 9.13: Process flow pro odbavení požadavku



Obrázek 9.14: Compiler console a system console a ukázka chyb vzniklých při kompilaci kódu.

řešení (např. stejné jako bylo použito v subprocessu přemísťování boxů), ovšem opět z nedostatku flow itemů nebylo možné řešení realizovat.

Autor se však domnívá, že tyto omezení mají jen malý vliv na výsledný chod modelu. Při modelování byly z hlediska úspory použitých objektů obecně zanedbány aspekty, které dle autora mají minimální vliv na vlastnosti modelu a naopak detailněji vymodelovány ty části procesu, který mají velký dopad na model.

Kapitola 10

Experimentální část

V experimentální části byly provedeny simulační běhy pro sklad typu AutoStore i pro regálový sklad. Pro každý model bylo vytvořeno několik různých variant. Pro tyto varianty pak ještě probíhaly zvlášť simulační běhy dle konfigurace varianty z hlediska počtu pracovních stanic, operátorů nebo robotů. V každém simulačním běhu byly sledovány výsledky pro jednotlivé metriky. Celkově bylo realizováno 20 simulačních běhu pro každou variantu. Z těchto metrik pak byly vyhodnoceny průměry, směrodatné odchylky a maximální a minimální hodnoty. Výsledky pro každou konfiguraci jsou pak uvedeny v tabulkách.

10.1 Příprava dat pro simulaci

Jelikož bezplatná licence FlexSim Express neumožňuje vytvářet nové posloupnosti náhodných čísel (tzv. random streams) a používá stále jednu a tu samou posloupnost vygenerovanou při prvním vytvoření modelu (pozn. simulace produkuje stále stejné výsledky), bylo nutné před samotnou experimentální částí připravit data náhodných veličin pro simulaci. Tato příprava dat spočívala v generování náhodných čísel v sešitu MS Excel, transformaci těchto čísel do hodnot příslušných rozdělení a následném export do programu FlexSim.

Náhodná čísla v MS Excel byla vytvořena pomocí příkazu

`=NÁH.ČÍSLO()`

který vygeneruje pseudonáhodné číslo v intervalu $(0, 1)$. Náhodné číslo je pak transformováno do hodnot rozdělení definujících:

1. velikost požadavku (počet vyskladňovaných boxů v požadavku)
2. typ vyskladňovaného boxu v požadavku
3. příchod požadavku v čase (normální rozdělení), viz. kapitola 8.1

Pro každé rozdělení jsou generovány samostatné soubory náhodných čísel. První a druhé rozdělení jsou diskrétní rozdělení. Hodnota pro tyto rozdělení se dostane přiřazením hodnoty z daného intervalu pro vygenerované náhodné číslo. V Excelu je toto realizováno pomocí příkazu:

=S.VYHLEDAT(hledat;tabulka;sloupec;1)

Tabulka s intervaly pro rozdělení definující velikost objednávky je zobrazena na obrázku 10.1. Tabulky s intervaly definujícími typ vyskládňovaných boxů jsou uvedeny v popisu varianty 2 a varianty 4.

Pro určení hodnot normálního rozdělení byla použita transformace náhodné veličiny daná rovnicemi 10.1 a 10.2

$$y = \sum_{i=1}^{12} y_i \quad (10.1)$$

$$x = \mu + (y - 6) \cdot \sigma \quad (10.2)$$

kde y_i je pseudonáhodné číslo vygenerované v MS Excel, μ je střední hodnota příslušného normálního rozdělení, σ je rozptyl příslušného normálního rozdělení a x je hodnota generovaná normálním rozdělením. Hodnoty pro zvolené rozdělení jsou $\mu = 45s$ a $\sigma = 15s$.

Pravděpodobnost velikosti objednávky		
<i>P jevu</i>	<i>interval</i>	<i>hodnota</i>
0,15	0	1
0,25	0,15	2
0,3	0,4	3
0,2	0,7	4
0,1	0,9	5

Obrázek 10.1: Pravděpodobnostní rozdělení pro velikost objednávky a hodnoty intervalů pro generování velikosti objednávky z pseudonáhodných čísel

Příslušné hodnoty náhodných veličin pak byly do softwaru FlexSim importovány pomocí nástroje Excel Interface, který například umožňuje importovat tabulku při každém restartu simulace (bylo využíváno pro simulační běhy).

10.2 AutoStore var.1 - výška 5 boxů, náhodné rozmístění

V této variantě proběhli simulace pro AutoStore sklad s náhodným rozmístěním boxů o počáteční skladovací výšce 5-ti boxů nad sebou. Parametry tohoto modelu jsou:

- 2470 boxů
- 25x20 skladovacích slotů
- rozměry skladu (délka × šířka × výška) 12 × 13.6 × 3 metry
- počet odbavovacích stanic: 1

- počáteční skladová výška: 5 boxů
- maximální počet krabic umístěných nad sebou: 7

Pro tuto variantu byly uvažovány 2 konfigurace (dle počtu robotů). Pro každou konfiguraci bylo provedeno 20 simulačních běhů. Konfigurace jsou:

1. konfigurace - 2 roboti AutoStore (výsledky v tabulce 10.1)
2. konfigurace - 3 roboti AutoStore (výsledky v tabulce 10.2)

Parametr	Průměr	Sm.Odch	MAX	MIN
Vytíženost robotů [%]	69,2	3,4	76,3	62,0
Vytíženost operátora [%]	63,16	3,1	69,8	56,5
Čas pro odbavení 200 požadavků [s]	9311	269	9810	8731
Poč. požadavků za 15 min	20,0	0,6	21,4	18,9
Najezdřená vzdálenost [m]	11936	437	12760	14
Průměrná doba odbavení požadavku [s]	44	2,5	48	11
Minimální doba pro odbavení požadavku [s]	16	1,2	19	2
Maximální doba pro odbavení požadavku [s]	94	31,2	128	2
Průměrná doba požadavku ve frontě [s]	5	2,3	12	1
Maximální doba požadavku ve frontě [s]	58	13,5	79	1
Maximální velikost fronty	2	0,6	3	1

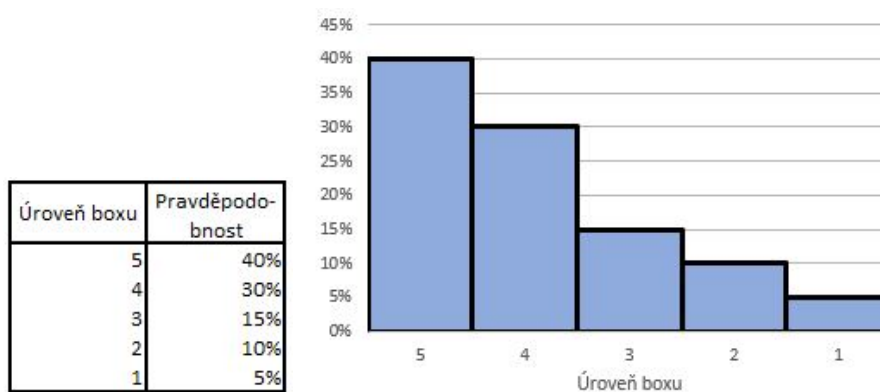
Tabulka 10.1: AutoStore - varianta č.1, 2 roboti

Parametr	Průměr	Sm.Odch	MAX	MIN
Vytíženost robotů [%]	44,8	2,8	65,9	40,5
Vytíženost operátora [%]	62,4	4,0	66,2	46,9
Čas pro odbavení 200 požadavků [s]	9272	215	9706	8919
Poč. požadavků za 15 min	20,1	0,5	20,9	19,2
Najezdřená vzdálenost [m]	7573	325	8109	6831
Průměrná doba odbavení požadavku [s]	32	1,2	35	31
Minimální doba pro odbavení požadavku [s]	16	1,3	18	13
Maximální doba pro odbavení požadavku [s]	70	11,5	99	55
Průměrná doba požadavku ve frontě [s]	1	0,6	3	0
Maximální doba požadavku ve frontě [s]	30	11,8	61	15
Maximální velikost fronty	1	0,5	3	1

Tabulka 10.2: AutoStore - varianta č.1, 3 roboti

10.3 AutoStore var. 2 - výška 5 boxů, rozmístění dle obrátkovosti

V této variantě proběhly simulace pro AutoStore sklad s rozmístěním boxů dle obrátkovosti o počáteční skladovací výšce 5-ti boxů nad sebou. Rozdělení dle obrátkovosti je možné vidět na obrázku 10.2, kdy 70% objednávek je na zboží umístěné v horních dvou boxech, které tvoří 30% všech naskladněných boxů. Parametry modelu pro tuto variantu jsou:



Obrázek 10.2: Pravděpodobností rozdělení obrátkovosti zboží dle úrovně boxu

- 2470 boxů
- 25x20 skladovacích slotů
- rozměry skladu (délka × šířka × výška) 12 × 13.6 × 3 metry
- počet odbavovacích stanic: 1
- počáteční výška: 5 boxů
- maximální počet krabic umístěných nad sebou: 7
- rozmístění boxů dle obrátkovosti (viz obr. 10.2)

Pro tuto variantu byly uvažovány 2 konfigurace (dle počtu robotů). Pro každou konfiguraci bylo provedeno 20 simulačních běhů.

1. konfigurace - 2 roboti (tabulka 10.3)
2. konfigurace - 3 roboti (tabulka 10.4)

Parametr	Průměr	Sm.Odch	MAX	MIN
Vytíženost robotů [%]	60,8	2,4	64,9	56,2
Vytíženost operátora [%]	63,2	2,3	66,7	58,9
Čas pro odbavení 200 požadavků [s]	9299	189	9686	8967
Poč. požadavků za 15 min	20,0	0,4	20,8	19,2
Naježděná vzdálenost [m]	11158	414	11694	10291
Průměrná doba odbavení požadavku [s]	36	2,5	43	32
Minimální doba pro odbavení požadavku [s]	16	2,3	18	6
Maximální doba pro odbavení požadavku [s]	87	19,6	137	65
Průměrná doba požadavku ve frontě [s]	2	1,5	8	1
Maximální doba požadavku ve frontě [s]	39	20,5	97	16
Maximální velikost fronty	2	0,7	4	1

Tabulka 10.3: AutoStore - varianta č.2, 2 roboti

Parametr	Průměr	Sm.Odch	MAX	MIN
Vytíženost robotů [%]	39,8	1,5	43,8	36,9
Vytíženost operátora [%]	63,1	1,7	69,1	61,0
Čas pro odbavení 200 požadavků [s]	9284	233	9655	8752
Poč. požadavků za 15 min	20,0	0,5	21,3	19,3
Naježděná vzdálenost [m]	7288	241	7899	6825
Průměrná doba odbavení požadavku [s]	28	0,8	30	27
Minimální doba pro odbavení požadavku [s]	16	1,1	18	14
Maximální doba pro odbavení požadavku [s]	59	13,3	89	44
Průměrná doba požadavku ve frontě [s]	1	0,6	3	0
Maximální doba požadavku ve frontě [s]	23	12,5	57	13
Maximální velikost fronty	1	0,0	1	1

Tabulka 10.4: AutoStore - varianta č.2, 3 roboti

10.4 AutoStore var.3 - výška 10 boxů, náhodné rozmístění

V této variantě proběhli simulace pro AutoStore sklad s náhodným rozmístěním boxů o počáteční skladovací výšce 10-ti boxů nad sebou. Parametry tohoto modelu jsou:

- 2470 boxů
- 25x20 skladovacích slotů
- rozměry skladu (délka × šířka × výška) 12 × 13.6 × 3 metry
- počet odbavovacích stanic: 2
- počáteční skladová výška: 10 boxů
- maximální počet krabic umístěných nad sebou: 12

Pro tuto variantu byly uvažovány 3 konfigurace (dle počtu robotů). Pro každou konfiguraci bylo provedeno 20 simulačních běhů.

1. konfigurace - 3 roboti (tabulka 10.5)
2. konfigurace - 4 roboti (tabulka 10.6)
3. konfigurace - 5 robotů (tabulka 10.7)

Parametr	Průměr	Sm.Odch	MAX	MIN
Vytíženost robotů [%]	94,6	2,6	98,7	86,5
Vytíženost operátora [%]	30,0	1,0	32,4	27,7
Čas pro odbavení 200 požadavků [s]	9385	215	10381	9417
Poč. požadavků za 15 min	18,7	0,4	19,7	17,9
Naježděná vzdálenost [m]	7718	269	8092	7206
Průměrná doba odbavení požadavku [s]	211	76	369	116
Minimální doba pro odbavení požadavku [s]	27	12,0	72	18
Maximální doba pro odbavení požadavku [s]	460	157,1	782	261
Průměrná doba požadavku ve frontě [s]	121	86,6	335	27
Maximální doba požadavku ve frontě [s]	348	155,7	666	145
Maximální velikost fronty	7	3,9	15	1

Tabulka 10.5: AutoStore - varianta č.3, 3 roboti

Parametr	Průměr	Sm.Odch	MAX	MIN
Vytíženost robotů [%]	76,5	3,9	82,4	65,8
Vytíženost operátora [%]	30,4	1,4	33,6	27,8
Čas pro odbavení 200 požadavků [s]	9402	178	9972	9233
Poč. požadavků za 15 min	19,2	0,4	20,2	18,6
Naježděná vzdálenost [m]	5752	206	6298	5499
Průměrná doba odbavení požadavku [s]	81	4,4	97	69
Minimální doba pro odbavení požadavku [s]	19	2,0	25	16
Maximální doba pro odbavení požadavku [s]	173	21,9	212	138
Průměrná doba požadavku ve frontě [s]	5	1,8	8	2
Maximální doba požadavku ve frontě [s]	72	25,3	146	42
Maximální velikost fronty	2	0,5	3	1

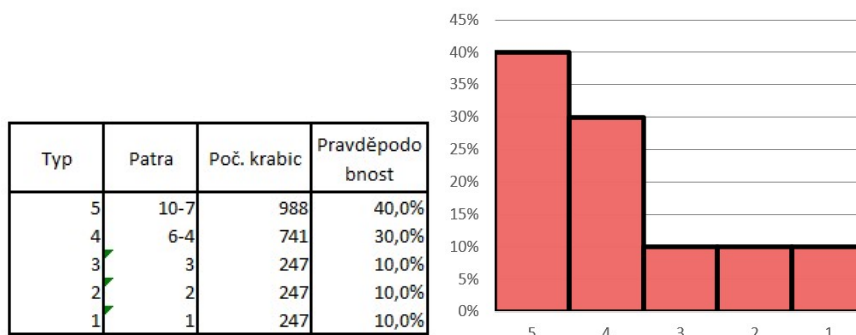
Tabulka 10.6: AutoStore - varianta č.3, 4 roboti

Parametr	Průměr	Sm.Odch	MAX	MIN
Vytíženost robotů [%]	59,0	2,5	64,7	52,7
Vytíženost operátora [%]	30,9	1,2	33,3	28,7
Čas pro odbavení 200 požadavků [s]	9282	166	9737	8805
Poč. požadavků za 15 min	19,7	0,4	20,6	19,1
Naježděná vzdálenost [m]	4623	141	5011	4415
Průměrná doba odbavení požadavku [s]	69	1,9	72	65
Minimální doba pro odbavení požadavku [s]	18	1,5	23	16
Maximální doba pro odbavení požadavku [s]	142	16,8	180	117
Průměrná doba požadavku ve frontě [s]	2	1,0	4	0
Maximální doba požadavku ve frontě [s]	51	19,0	101	26
Maximální velikost fronty	2	0,7	3	1

Tabulka 10.7: AutoStore - varianta č.3, 5 robotů

10.5 AutoStore var.4 - výška 10 boxů, rozmístění dle obrátkovosti

V této variantě proběhly simulace pro AutoStore sklad s rozmístěním boxů dle obrátkovosti o počáteční skladovací výšce 10-ti boxů nad sebou. Rozdělení dle obrátkovosti je možné vidět na obrázku 10.3, kdy 70% objednávek je na zboží umístěné v horních dvou boxech, které tvoří 30% všech naskladněných boxů.



Obrázek 10.3: Pravděpodobností rozdělení obrátkovosti zboží do 5 typů a odpovídající úrovně skladu

Parametry modely jsou:

- 2470 boxů
- 25x20 skladovacích slotů
- rozměry skladu (délka × šířka × výška) 12 × 13.6 × 3 metry
- počet odbavovacích stanic: 2
- počáteční skladová výška: 10 boxů

- maximální počet krabic umístěných nad sebou: 12
- rozmístění boxů dle obrátkovosti

Pro tuto variantu byly uvažovány 3 konfigurace (dle počtu robotů). Pro každou konfiguraci bylo provedeno 20 simulačních běhů.

1. konfigurace - 3 roboti (tabulka 10.8)
2. konfigurace - 4 roboti (tabulka 10.9)
3. konfigurace - 5 robotů (tabulka 10.10)

Parametr	Průměr	Sm.Odch	MAX	MIN
Vytíženost robotů [%]	82,3	11,2	98,8	55,9
Vytíženost operátora [%]	30,3	1,5	32,1	26,8
Čas pro odbavení 200 požadavků [s]	9399	305	9885	9005
Poč. požadavků za 15 min	19,9	0,7	20,0	17,2
Najezdřená vzdálenost [m]	6971	623	8280	5623
Průměrná doba odbavení požadavku [s]	109	58,5	295	42
Minimální doba pro odbavení požadavku [s]	23	6,6	43	17
Maximální doba pro odbavení požadavku [s]	292	189,0	895	108
Průměrná doba požadavku ve frontě [s]	33	28,6	102	0
Maximální doba požadavku ve frontě [s]	190	141,7	702	40
Maximální velikost fronty	5	4,4	20	1

Tabulka 10.8: AutoStore - varianta č.4, 3 roboti

Parametr	Průměr	Sm.Odch	MAX	MIN
Vytíženost robotů [%]	82,3	11,2	98,8	55,9
Vytíženost operátora [%]	30,3	1,5	32,1	26,8
Čas pro odbavení 200 požadavků [s]	9399	305	9885	9005
Poč. požadavků za 15 min	19,9	0,7	20,0	17,2
Najezdřená vzdálenost [m]	6971	623	8280	5623
Průměrná doba odbavení požadavku [s]	109	58,5	295	42
Minimální doba pro odbavení požadavku [s]	23	6,6	43	17
Maximální doba pro odbavení požadavku [s]	292	189,0	895	108
Průměrná doba požadavku ve frontě [s]	33	28,6	102	0
Maximální doba požadavku ve frontě [s]	190	141,7	702	40
Maximální velikost fronty	5	4,4	20	1

Tabulka 10.9: AutoStore - varianta č.4, 4 roboti

Parametr	Průměr	Sm.Odch	MAX	MIN
Vytíženost robotů [%]	82,3	11,2	98,8	55,9
Vytíženost operátora [%]	30,3	1,5	32,1	26,8
Čas pro odbavení 200 požadavků [s]	9399	305	9885	9005
Poč. požadavků za 15 min	19,9	0,7	20,0	17,2
Naježděná vzdálenost [m]	6971	623	8280	5623
Průměrná doba odbavení požadavku [s]	109	58,5	295	42
Minimální doba pro odbavení požadavku [s]	23	6,6	43	17
Maximální doba pro odbavení požadavku [s]	292	189,0	895	108
Průměrná doba požadavku ve frontě [s]	33	28,6	102	0
Maximální doba požadavku ve frontě [s]	190	141,7	702	40
Maximální velikost fronty	5	4,4	20	1

Tabulka 10.10: AutoStore - varianta č.4, 5 robotů

10.6 Regálový sklad, var. 5 - náhodné rozmístění

V této variantě proběhly simulace pro regálový sklad s náhodným rozmístěním boxů. Parametry modelu pro tuto variantu jsou uvedeny v následujících bodech.

- 2470 boxů
- 10 regálů o délce 27 metrů
- rozměr skladu: 33x17 metrů

Pro regálový sklad byly vytvořeny následující konfigurace dle počtu operátorů obsluhující sklad.

1. konfigurace - 3 operátoři (tabulka 10.11)
2. konfigurace - 4 operátoři (tabulka 10.12)

Parametr	Průměr	Sm.Odch	MAX	MIN
Vytíženost operátora [%]	67,2	3,3	74,0	60,8
Čas pro odbavení 200 požadavků [s]	9346	252	10008	8975
Poč. požadavků za 15 min	19,9	0,5	20,8	18,5
Naježděná vzdálenost [m]	4323	140	4672	4053
Průměrná doba odbavení požadavku [s]	92	6,0	99	69
Minimální doba pro odbavení požadavku [s]	20	2,7	27	17
Maximální doba pro odbavení požadavku [s]	186	11,7	214	169
Průměrná doba požadavku ve frontě [s]	2	1,0	5	1
Maximální doba požadavku ve frontě [s]	57	15,1	93	34
Maximální velikost fronty	2	0,5	3	1

Tabulka 10.11: Regálový sklad se 3-mi operátory

Parametr	Průměr	Sm.Odch	MAX	MIN
Vytíženost operátora [%]	51,5	2,6	59,3	45,4
Čas pro odbavení 200 požadavků [s]	9240	146	9489	8912
Poč. požadavků za 15 min	20,1	0,4	20,9	19,4
Naježděná vzdálenost [m]	3356	151	3636	3067
Průměrná doba odbavení požadavku [s]	93	3,2	99	88
Minimální doba pro odbavení požadavku [s]	20	2,7	28	16
Maximální doba pro odbavení požadavku [s]	189	16,6	232	173
Průměrná doba požadavku ve frontě [s]	0	0,1	0	0
Maximální doba požadavku ve frontě [s]	21	12,1	42	3
Maximální velikost fronty	1	0,3	2	1

Tabulka 10.12: Regálový sklad se 4-mi operátory

10.7 Regálový sklad, var. 6 - rozmístění dle obrátkovosti

V této variantě proběhly simulace pro regálový sklad s rozmístěním boxů dle obrátkovostních tříd (typů). Celkem bylo zvoleno pět obrátkovostních tříd, kdy 70% vyskladňovaných boxů tvoří 30% všech naskladněných boxů v regálovém skladu. Zavedená obrátkovost je identická s obrátkovostí použitou ve variantě 2 a je zobrazena na obrázku 10.2. Parametry modelu pro setříděný regálový sklad jsou uvedeny v následujících bodech:

- 2470 boxů
- 10 regálů o délce 27 metrů
- rozměr skladu: 33x17 metrů
- řazení boxů dle obrátkovosti viz. obr. 10.2

Pro regálový sklad byly vytvořeny následující konfigurace dle počtu operátorů obsluhující sklad.

1. konfigurace - 3 operátoři (tabulka 10.13)
2. konfigurace - 4 operátoři (tabulka 10.14)

Parametr	Průměr	Sm.Odch	MAX	MIN
Vytíženost operátora [%]	61,5	2,9	68,9	52,9
Čas pro odbavení 200 požadavků [s]	9222	192	9677	8833
Poč. požadavků za 15 min	20,1	0,5	21,1	18,9
Naježděná vzdálenost [m]	3420	164	3827	3208
Průměrná doba odbavení požadavku [s]	84	2,8	90	79
Minimální doba pro odbavení požadavku [s]	20	2,5	23	15
Maximální doba pro odbavení požadavku [s]	186	51,8	402	152
Průměrná doba požadavku ve frontě [s]	1	0,5	2	0
Maximální doba požadavku ve frontě [s]	38	14,3	71	16
Maximální velikost fronty	2	0,5	2	1

Tabulka 10.13: Regálový sklad řazený dle obrátkovosti se 3-mi operátory

Parametr	Průměr	Sm.Odch	MAX	MIN
Vytíženost operátora [%]	45,9	2,7	52,2	39,5
Čas pro odbavení 200 požadavků [s]	9277	180	9637	8830
Poč. požadavků za 15 min	20,0	0,4	21,1	19,3
Naježděná vzdálenost [m]	2495	287	2781	1430
Průměrná doba odbavení požadavku [s]	83	1,8	86	79
Minimální doba pro odbavení požadavku [s]	21	14,0	82	15
Maximální doba pro odbavení požadavku [s]	161	33,7	188	19
Průměrná doba požadavku ve frontě [s]	0	0,1	0	0
Maximální doba požadavku ve frontě [s]	7	8,6	27	0
Maximální velikost fronty	1	0,0	1	1

Tabulka 10.14: Regálový sklad řazený dle obrátkovosti se 4-mi operátory

10.8 AutoStore sklad, var. 7 - odbavení více požadavků najednou

V této variantě proběhli simulace pro AutoStore sklad s rozmístěním boxů dle obrátkovosti o počáteční skladovací výšce 10-ti boxů nad sebou. Rozdělení dle obrátkovosti je možné vidět na obrázku 10.3. V této variantě také pracoviště odbavení může **současně odbavovat až dva požadavky najednou**. Parametry modelu pro tuto variantu jsou:

- 2470 boxů
- 25x20 skladovacích slotů
- rozměry skladu (délka × šířka × výška) 12 × 13.6 × 3 metry
- počet odbavovacích stanic: 1
- počáteční skladová výška: 10 boxů
- maximální počet krabic umístěných nad sebou: 12
- možnost odbavovat dva požadavky současně

Pro tuto variantu byla uvažována následující konfigurace

1. konfigurace - 4 roboti (tabulka 10.15)

Parametr	Průměr	Sm.Odch	MAX	MIN
Vytiženost robotů [%]	68,8	9,9	80,2	5,5
Vytiženost operátora [%]	62,7	1,4	64,4	59,0
Čas pro odbavení 200 požadavků [s]	9378	194	9698	8943
Poč. požadavků za 15 min	19,8	0,4	20,7	19,2
Najezdřená vzdálenost [m]	6010	408	6804	5487
Průměrná doba odbavení požadavku [s]	74	24,8	132	52
Minimální doba pro odbavení požadavku [s]	23	6,2	39	17
Maximální doba pro odbavení požadavku [s]	193	84,6	340	89
Průměrná doba požadavku ve frontě [s]	21	21,1	65	0
Maximální doba požadavku ve frontě [s]	96	70,6	227	17
Maximální velikost fronty	4	2,7	10	1

Tabulka 10.15: Setříděný AutoStore sklad o výšce 10-ti boxů a možnosti odbavovat až dva požadavky najednou, konfigurace se 4-mi operátory.

Kapitola 11

Vyhodnocení výsledků simulace

V této části simulace jsou vyhodnoceny výsledky z experimentální části. Pro lepší přehlednost je tato část rozdělena na podkapitoly dle výsledků zkoumaných rozdílů a parametrů.

11.1 Porovnání setříděného a nesetříděného AutoStore skladu

V této podkapitole jsou diskutovány výsledky mezi setříděným a nesetříděným skladem AutoStore (varianty 1, 2, 3, a 4). Parametry pro porovnání byly zvoleny na základě procentuální změny mezi výsledky průměrných hodnot získaných pro setříděný a nesetříděný sklad, viz 11.1. Procentuální změna byla vypočtena jako procentuální změna setříděného skladu vůči nesetříděnému (nesetříděný = 100%).

Počáteční výška: 5 boxů			
<i>Počet robotů: 3</i>			
Parametr	Pr. hodnota pro nesetříděný	Pr. hodnota pro setříděný	% změna
Vytíženost robotů [%]	44,8	39,8	-12,7%
Průměrná doba odbavení požadavku [s]	32	28	-14,1%
Maximální doba pro odbavení požadavku [s]	70	59	-18,7%
Maximální doba požadavku ve frontě [s]	30	23	-30,7%
<i>Počet robotů: 2</i>			
Vytíženost robotů [%]	69,2	60,8	-13,8%
Průměrná doba odbavení požadavku [s]	44	36	-22,1%
Maximální doba pro odbavení požadavku [s]	94	87	-8,0%
Maximální doba požadavku ve frontě [s]	58	39	-48,9%

Tabulka 11.1: Významně změněné hodnoty mezi setříděným a nesetříděným skladem pro sklad o počáteční výšce 5 boxů

Z tabulky 11.1 je patrné, že pro obě konfigurace nastala významná změna zejména ve vytíženosti robotů (pokles o 12,7% na 39,8% a o 13,8% na 60,8%), maximální době pro odbavení požadavku (pokles o 18,7% na 59s a o 8%

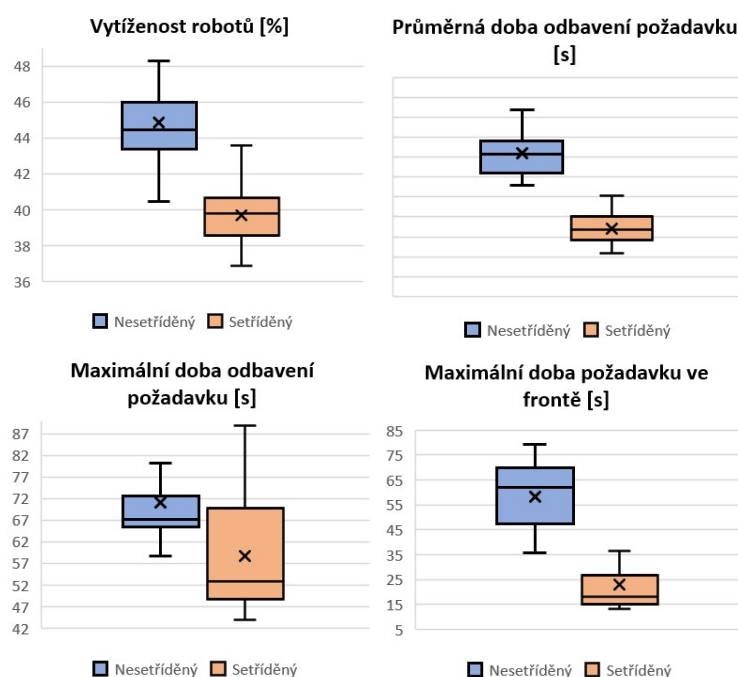
na 87s) maximální době, kterou požadavek stráví ve frontě (pokles o 30,7% na 30s a 48,9% na 39s) a v průměrné době odbavení požadavku (pokles o 14,1% na 28s a o 22,1% na 36s). Pro variantu se dvěma roboty došlo dále k výraznému snížení průměrné doby požadavku ve frontě v případě setříděného skladu. Obdobné výsledky dosahuje i porovnání mezi sklady o výšce 10-ti boxů, jak je možné vidět v tabulce 11.2. Grafické porovnání rozdílů mezi

Počáteční výška: 10 boxů			
Počet robotů: 3			
Parametr	Pr. hodnota pro nesetříděný	Pr. hodnota pro setříděný	% změna
Vytíženost robotů [%]	94,6	82,3	-13,0%
Průměrná doba odbavení požadavku [s]	211	109	-48,4%
Minimální doba pro odbavení požadavku [s]	27	23	-14,6%
Maximální doba pro odbavení požadavku [s]	460	292	-36,5%
Průměrná doba požadavku ve frontě [s]	121	33	-72,6%
Maximální doba požadavku ve frontě [s]	348	190	-45,3%
Maximální velikost fronty	7	5	-26,8%
Počet robotů: 4			
Vytíženost robotů [%]	76,5	67,0	-12,4%
Průměrná doba odbavení požadavku [s]	81	68	-16,0%
Maximální doba pro odbavení požadavku [s]	173	151	-12,7%
Maximální doba požadavku ve frontě [s]	72	59	-18,1%
Počet robotů: 5			
Vytíženost robotů [%]	59,0	50,9	-13,7%
Průměrná doba odbavení požadavku [s]	69	56	-18,9%
Maximální doba pro odbavení požadavku [s]	142	117	-17,6%
Maximální doba požadavku ve frontě [s]	51	32	-37,4%

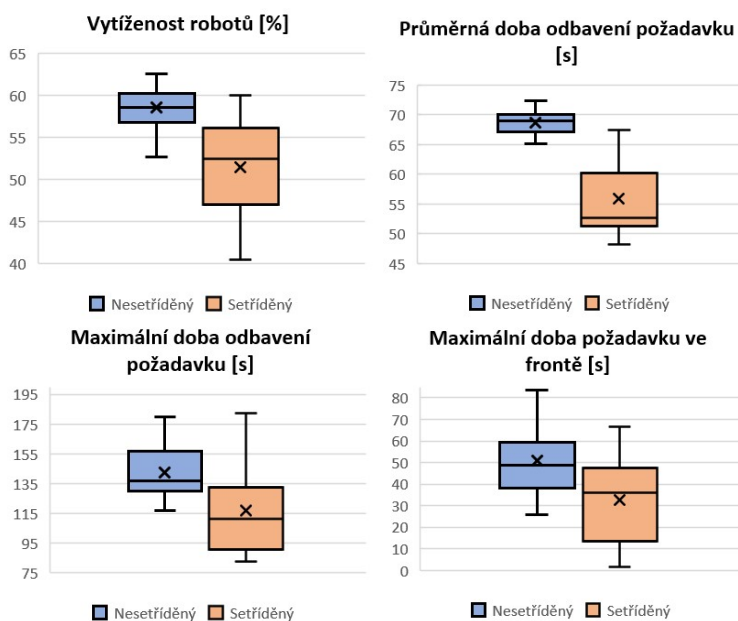
Tabulka 11.2: Významně změněné hodnoty mezi setříděným a nesetříděným skladem pro sklad o počáteční výšce 10 boxů

setříděnými a nesetříděnými sklady pro změny významných parametrů jsou pro skladové výšky 5-ti a 10-ti boxů stejných konfiguracích (varianta 1 vs. varianta 2 a varianta 3 vs. varianta 4) uvedeny na krabicových grafech na obrázcích 11.1 a 11.2.

11.1. Porovnání setříděného a nesetříděného AutoStore skladu



Obrázek 11.1: Krabicový graf pro setříděný a nesetříděný sklad o výšce 5-ti boxů a konfiguraci třech robotů.



Obrázek 11.2: Krabicový graf pro setříděný a nesetříděný sklad o výšce 10-ti boxů a konfiguraci pěti robotů.

Z tabulek a grafů je patrné, že u všech konfigurací dochází k výrazným změnám hlavně ve:

1. vytížení robotů (v průměru o 13,1%)

2. průměrné době odbavení požadavků (v průměru o 23,9%)
3. maximální době pro odbavení požadavků (v průměru o 14,3%)
4. maximální době požadavku ve frontě (v průměru o 36,1%)

Zajímavé však je, že pro konfiguraci s nižším počtem robotů u varianty skladu s počáteční výškou 10ti boxů dochází k výrazné změně i u dalších parametrů:

1. průměrné době požadavku ve frontě
2. minimální době pro odbavení požadavku
3. maximální velikosti fronty

Autor se domnívá, že změny těchto parametrů pro konfiguraci s nižším počtem robotů jsou způsobeny tím, že tato konfigurace není dobře optimalizována pro definované rozdělení příchodu požadavků (konfigurace dosahuje např. velkých časů, které požadavky stráví ve frontě nebo velkých hodnot pro maximální velikost fronty). To dokládá také to, že změny společných parametrů jsou vyšší u této konfigurace oproti ostatním. Například procentuální změna průměrné doby odbavení požadavků je pro konfiguraci s nejnižším počtem robotů -35,3%, zatímco pro ostatní konfigurace pouze -16,3%.

Z výsledku získaných v této podkapitole vyplývá, že řazení dle obrátkovosti výrazně zvyšuje výkonost AutoStore skladu. Pokud sklad vykazuje prvky větší poptávky po určitých typech zboží, bylo by vhodné, aby podnik určil typy zboží s vysokou obrátkovostí a ty pak v případě AutoStore skladu řadil na vrchní pozice.

11.2 Porovnání AutoStore skladu z hlediska výšky boxů

V této podkapitole jsou diskutovány výsledky a rozdíly mezi AutoStore sklady o výšce 5-ti a 10-ti boxů (varianty 1, 2, 3, a 4). Výsledky ze simulací jsou uvedeny v tabulkách 11.3 a 11.4. Tyto tabulky porovnávají vybrané konfigurace výše zmíněných variant. V tabulkách 11.3 jsou porovnány vybrané konfigurace pro nesetříděný sklad (náhodně rozmístěné boxy) a v tabulce 11.4 jsou porovnány vybrané konfigurace pro sklad setříděný (dle obrátkovostních kategorií). Procentuální změna je definována jako procentuální změna skladu vyšší výšky oproti skladu s nižší výškou (100% = parametr skladu s nižší výškou).

Při zvýšení výšky boxů na 10 boxů umístěných nad sebou již AutoStore sklad nestíhal odbavovat požadavky pouze s použitím jedné odbavovací stanice. Bylo tedy nutné pro tuto výšku použít tyto stanice dvě. V obou variantách (setříděný a nesetříděný sklad) došlo jak výraznému nárůstu průměrné doby odbavení požadavku, tak i průměrné maximální doby odbavení požadavku. Průměrná doba odbavení vzrostla v obou případech téměř dvojnásobně, např. v tabulce 11.3 (nesetříděný AutoStore sklad) změna z 32s na 69s a v tabulce

11.4 (setříděný AutoStore sklad) změna z 28s na 56s. To je spojeno zejména s navýšením množství boxů, které je nutné vyskladnit, než se robot dostane k požadovanému boxu. Maximální doba pak vzrostla pro konfigurace uvedené v tabulce 11.3 z 70s na 142s a pro konfigurace uvedené v tabulce 11.4 z 59s na 117s.

Vyznaným negativním dopadem při zvýšení počáteční výšky na 10 boxů je také snížení vytíženosti operátora, kdy operátor není efektivně vytěžován (téměř 70% času stráví čekáním na příchod boxu). To je dáno zvýšením počtu odbavovacích stanic.

PRŮMĚRNÉ HODNOTY PARAMETRŮ SKLADŮ O ROZDÍLNÝCH VÝŠKÁCH			
Výška skladu	5	10	
Parametr	Nesetříděný: 3 roboti	Nesetříděný: 5 robotů, 2 stanice	% změna
Vytíženost robotů [%]	44,8	59,0	32%
Vytíženost operátora [%]	62,4	30,9	-50%
Čas pro odbavení 200 požadavků [s]	9272	9282	0%
Naježděná vzdálenost [m]	7573	4623	-39%
Průměrná doba odbavení požadavku [s]	32	69	115%
Maximální doba pro odbavení požadavku [s]	70	142	102%
Maximální doba požadavku ve frontě [s]	30	51	70%

Tabulka 11.3: Významně odlišné parametry pro nesetříděné sklady o různých výškách

PRŮMĚRNÉ HODNOTY PARAMETRŮ SKLADŮ O ROZDÍLNÝCH VÝŠKÁCH			
Výška skladu	5	10	
Parametr	Setříděný: 3 roboti	Setříděný: 4 roboti, 2 stanice	% změna
Vytíženost robotů [%]	39,8	50,9	28%
Vytíženost operátora [%]	63,1	30,9	-51%
Čas pro odbavení 200 požadavků [s]	9284	9358	1%
Naježděná vzdálenost [m]	7288	4293	-41%
Průměrná doba odbavení požadavku [s]	28	56	100%
Maximální doba pro odbavení požadavku [s]	59	117	98%
Maximální doba požadavku ve frontě [s]	23	32	39%

Tabulka 11.4: Významně odlišné parametry pro setříděné sklady o různých výškách

Pro lepší využití operátorů v případě vysokých AutoStore skladů by bylo vhodné zvážit použití dané odbavovací stanice pro odbavování více požadavků najednou. Pro zjištění vlastností skladu s možností odbavovat více požadavků najednou byla vytvořena varianta 7. V té byla realizována simulace pro setříděný AutoStore sklad o výšce 10-ti boxů, s jednou odbavovací stanicí a možností odbavovat až dva požadavky najednou. Tato konfigurace byla srovnána s příslušnou konfigurací z varianty 4. Výsledky jsou uvedeny v tabulce 11.6, kde jsou světle červenou barvu vyznačeny významné změny parametrů.

Z výsledků je patrné, že ve skladu s možností odbavovat více požadavků najednou může dojít k přetížení pracoviště odbavení. To se ve statistice promítlo v poměrně velké směrodatné odchylce ve vytížení operátora vůči skladu z varianty 4, kdy směrodatná odchylka pro sklad s možností odbavení více požadavků najednou je 10,2% (varianta 7) a pro běžný AutoStore sklad se dvěma pracovišti (varianta 4) pouze 0,9%. Zablokování pracoviště má také negativní vliv na čas potřebný k odbavení jednoho požadavku. To dokazují zejména změny maximální hodnoty jak pro průměrnou dobu odbavení požadavku, tak pro maximální, které je možné vidět v tabulce 11.6. S tím souvisí i výrazný nárůst času stráveného ve frontě (např. průměrný čas ve frontě pro variantu 4 jsou 4s a pro variantu 7 to je 21s). Výhoda tohoto řešení je ovšem v lepším využití pracovníka oproti skladu z varianty 4.

Významným faktorem, který je pro jednotlivé varianty třeba zmínit, je výrazné snížení skladovací plochy v případě AutoStore skladu o výšce 10-ti boxů (zmenšení skladovací plochy o 49%, jak můžeme vidět v tabulce 11.5).

	AutoStore 10	AutoStore 5
Plocha [m ²]	83,6	163,2

Tabulka 11.5: Porovnání plochy skladu pro Autostore o výšce 5-ti a 10-ti boxů a regálového skladu

Maximální počet najednou odbavovaných požadavků: počet robotů	2				1				%změna průměru vůči skladu se 2-mi pracovišti
	Průměr	Sm. Odch.	Max	Min	Průměr	Sm. Odch.	Max	Min	
Vytíženost robotů [%]	68,8	9,9	80,2	57,5	67,0	6,2	75,8	60,2	3%
Vytíženost operátora [%]	72,5	10,2	89,5	59,0	31,4	0,9	32,7	29,8	131%
Čas pro odbavení 200 požadavků [s]	9378	194	9801	8943	9358	194	9815	###	0%
Poč. požadavků za 15 min	19,7	0,4	20,7	18,8	20,0	0,4	20,8	19,0	-2%
Naježděná vzdálenost [m]	6010	408	6804	5487	5840	362	6610	###	3%
Průměrná doba odbavení požadavku [s]	74	24,8	132	52	68	7,5	73	59	9%
Minimální doba pro odbavení požadavku [s]	23	6,2	39	17	20	2,3	25	18	15%
Maximální doba pro odbavení požadavku [s]	193	84,6	340	89	151	22,2	190	106	28%
Průměrná doba požadavku ve frontě [s]	21	21,1	65	0	4	1,3	5	0	425%
Maximální doba požadavku ve frontě [s]	96	70,6	227	17	59	22,0	95	33	63%
Maximální velikost fronty	4	2,7	10	1	2	0,7	4	1	50%

Tabulka 11.6: Porovnání setříděného skladu s možností odbavovat více požadavků najednou a skladu z varianty 4.

Z výsledku této podkapitoly vyplývá, že zvýšení výšky AutoStore skladu výrazně snižuje nároky na skladovací prostory. Nevýhodou však je zhoršení skladových parametrů, zejména průměrných a maximálních dob odbavení požadavků a průměrných a maximálních dob, které požadavky stráví ve frontě. Dochází také k snížení vytíženosti a efektivity pracovníků (z 72% na 31%). Tento negativní dopad se dá kompenzovat řešením, při kterém dané pracoviště dokáže odbavovat více požadavků najednou.

11.3 Porovnání AutoStore skladu a běžně používaného regálového skladu

V této podkapitole jsou diskutovány výsledky získané z porovnání simulací pro regálový sklad a sklad AutoStore. Porovnání bylo provedeno pro varianty AutoStore skladu (varianty 1, 2, 3 a 4) a regálového skladu nesetříděného (varianta 5) a setříděného (varianta 6). Srovnání s regálovými sklady probíhalo zvláště pro AutoStore sklad o výšce 5-ti boxů a pro sklad o výšce 10-ti boxů, jak je možné vidět v tabulkách 11.7 a 11.8. Uvažované konfigurace zvolené k porovnání lze považovat za optimální z hlediska schopnosti odbavovat požadavky bez velkých front.

Varianty 1 a 2 (výška skladu 5 boxů) v porovnání s regálovým skladem dosahují výrazně lepší průměrné a maximální doby pro odbavení požadavků, kdy dochází ke snížení až o 100% oproti běžným regálovým skladům. Např. setříděný AutoStore sklad s jedním pracovníkem a třemi roboty odbaví požadavek průměrně za 28s, zatímco setříděný regálový sklad se 4-mi pracovníky za 83s. Grafické porovnání průměrných dob je pro jednotlivé varianty a konfigurace je zachyceno na krabicovém grafu na obrázku 11.7. Na druhou stranu je třeba zmínit, že regálové sklady dosáhly o něco lepšího výsledku v porovnání průměrné a maximální doby požadavku ve frontě (např. setříděný regálový sklad o počtu 4 operátorů má maximální dobu požadavku ve frontě 7s, zatímco setříděný AutoStore sklad 23s). Toto je ovšem způsobeno tím, že běžné regálové sklady musely pro dosažení stejných časů pro odbavení 200 požadavků použít více operátorů (3 a 4, zatímco AutoStore sklad využíval pouze jednoho operátora). Lze se domnívat, že v případě zvýšení počtu odbavovacích stanic (operátorů) v případě AutoStore skladu, by došlo ke snížení tohoto rozdílu.

	Regálový sklad				Autostore			
	Nesetříděný		Setříděný		Nesetříděný		Setříděný	
Počet operátorů	3	4	3	4	1	1	1	1
Počet robotů	-	-	-	-	2	3	2	3
Parametr	Průměr		Průměr		Průměr		Průměr	
Vytiženost operátora [%]	67,2	51,5	61,5	45,9	63,2	62,4	63,2	63,1
Čas pro odbavení 200 požadavků [s]	9346	9240	9222	9277	9311	9272	9299	9284
Poč. požadavků za 15 min	19,9	20,1	20,1	20	20,0	20,1	20,0	20,0
Najezděná vzdálenost [m]	4323	3356	3420	2495	11936	7573	11158	7288
Průměrná doba odbavení požadavku [s]	92	93	84	83	44	32	36	28
Minimální doba pro odbavení požadavku [s]	20	20	20	21	16	16	16	16
Maximální doba pro odbavení požadavku [s]	186	189	186	161	94	70	87	59
Průměrná doba požadavku ve frontě [s]	2	0	1	0	5	1	2	1
Maximální doba požadavku ve frontě [s]	57	21	38	7	58	30	39	23
Maximální velikost fronty	2	1	2	1	2	1	2	1

Tabulka 11.7: Porovnání průměrné doby odbavení požadavku AutoStore skladu o výšce 5-ti boxů s variantami regálových skladů.

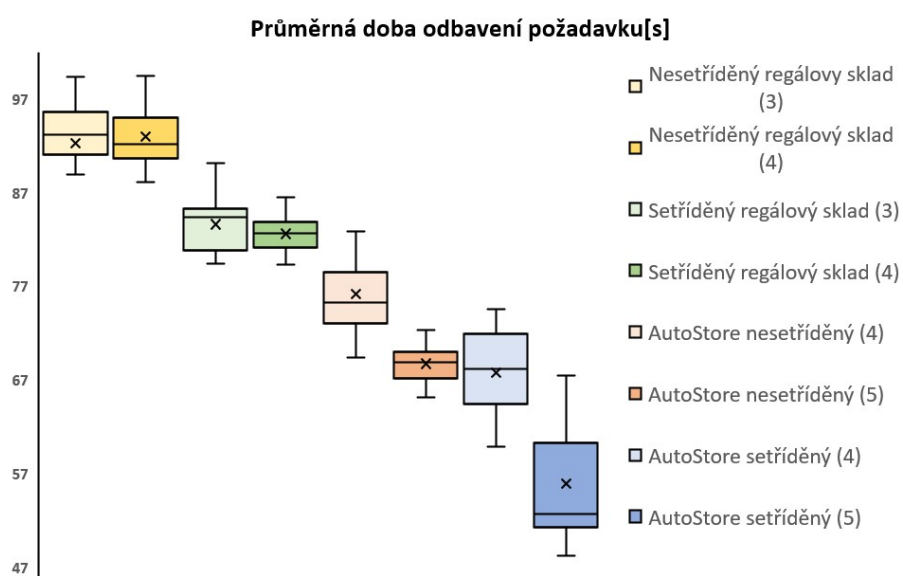
Při navýšení skladové výšky na 10 boxů došlo ke snížení rozdílů mezi AutoStore skladem a klasickým regálovým skladem, zejména co se týče času pro odbavení požadavků. Např. pro konfiguraci AutoStore skladu s nejlepšími výsledky (5 robotů) je průměrná doba pro odbavení požadavku pro setříděný AutoStore sklad 56s a pro nesetříděný 69s, zatímco pro regálový sklad je tato doba pro setříděný sklad 83s a pro nesetříděný 93s. Porovnání průměrných dob pro odbavení jednoho požadavku pro AutoStore sklad o výšce 10-ti boxů a regálového skladu je pro příslušné varianty graficky znázorněno na obrázku 11.3.

Je ovšem třeba zmínit, že AutoStore sklad o výšce 10-ti boxů však stále potřebuje k odbavení daného počtu požadavků menšího počtu operátorů než v případě klasického regálového skladu. Velkou výhodou je také menší fyzická náročnost práce operátorů.

	Regálový sklad				Autostore			
	Nesetříděný		Setříděný		Nesetříděný		Setříděný	
Počet operátorů	3	4	3	4	1	1	1	1
Počet robotů	-	-	-	-	3	5	4	5
Parametr	Průměr		Průměr		Průměr		Průměr	
Vytíženost operátora [%]	67,2	51,5	61,5	45,9	30,4	30,9	31,4	30,9
Čas pro odbavení 200 požadavků [s]	9346	9240	9222	9277	9402	9466	9358	9299
Poč. požadavků za 15 min	19,9	20,1	20,1	20	19,2	19,7	20,0	20
Najezdná vzdálenost [m]	4323	3356	3420	2495	5752	4623	5840	4293
Průměrná doba odbavení požadavku [s]	92	93	84	83	81	69	68	56
Minimální doba pro odbavení požadavku [s]	20	20	20	21	19	18	20	19
Maximální doba pro odbavení požadavku [s]	186	189	186	161	173	142	151	117
Průměrná doba požadavku ve frontě [s]	2	0	1	0	5	2	4	1
Maximální doba požadavku ve frontě [s]	57	21	38	7	72	51	59	32
Maximální velikost fronty	2	1	2	1	2	2	2	1

Tabulka 11.8: Porovnání variant: AutoStore skladu o výšce 10-ti boxů s variantami regálových skladů.

Z výsledků porovnání lze konstatovat, že AutoStore sklad dosahuje stejného výkonu skladu (z hlediska času pro odbavení daného počtu požadavků) jako regálový sklad při nižším množství operátorů. Systém je zároveň rychlejší co se týče času potřebného na vyskladnění požadavku, a to i v případě srovnání neseřazeného AutoStore skladu se seřazeným regálovým skladem. Menší nároky na počet operátorů přináší snížení fixních skladových nákladů a zároveň může napomoci řešení problému nedostatku zaměstnanců v oblasti skladování. Je však také třeba zmínit, že v případě investice do tohoto typu skladu je třeba správně optimalizovat jak počet odbavovacích stanic a robotů, tak také správnou výšku tohoto skladu. Jak je možné vidět z výsledků uvedených v této a předešlé podkapitole, právě výška skladovacího systému má velmi významný vliv na výkonost skladu typu AutoStore (RCSR).



Obrázek 11.3: Porovnání průměrné doby odbavení požadavku AutoStore skladu o výšce 10-ti boxů s variantami regálových skladů.

Závěr

Tato práce se zabývá simulační studií systémů skladování, konkrétně se zaměřuje na sklad AutoStore. V teoretické části byly v souvislosti s praktickou částí podrobně popsány základy průmyslových simulací a jejich problematika včetně popisu vybraných softwarových nástrojů.

AutoStore sklad je moderní robotizovaný sklad typu RCSR (Robot Based Compact Storage and Retrieval System). Tento sklad byl v praktické části v rámci simulační studie analyzován a dále srovnán s běžně používaným typem regálového skladu. V rámci studie byly vytvořeny simulační modely v softwaru FlexSim Express (pozn. bezplatná zkušební licence omezená počtem objektů) pro oba typy skladů. Postup tvorby simulačních modelů je popsán v kapitole 9. Zde jsou také diskutovány možné omezení oproti reálnému AutoStore skladu, které vznikly zejména omezeným počtem modelových objektů ve zkušební licenci.

Pro modelování náhodných jevů v simulaci bylo využito tabulkového procesoru MS Excel, kde byly vytvořeny tabulky pro generování pseudonáhodných čísel, ze kterých se poté generovaly příslušné hodnoty statistických rozdělení. Tyto hodnoty pak byly ve formě tabulky importovány do simulačního modelu, kdy pro každý simulační běh došlo ke generování nových náhodných čísel.

V experimentální části simulační studie bylo simulováno několik variant scénářů skladování. Výsledky těchto scénářů pak byly diskutovány v podkapitole 11.

Z výsledku vyplývá, že sklad AutoStore přináší oproti manuálně odbavovaným regálovým skladům stejné velikosti úsporu zejména v počtu operátorů (např. AutoStore sklad zvládal odbavovat požadavky pouze s jedním operátorem, zatímco regálový sklad potřeboval operátory minimálně 3). Tím dochází ke snížení fixních skladových nákladů. V současné době, kdy je na pracovním trhu nedostatek zaměstnanců v oblasti skladování, AutoStore sklad může napomoci podnikům tento nedostatek kompenzovat. AutoStore sklad také dosahoval významně lepších výsledků v parametrech souvisejících s časem nutným pro odbavení jednoho požadavku. AutoStore sklad o počáteční výšce 5-ti boxů obecně zvládal odbavovat požadavky okolo 40s, zatímco regálovému skladu to trvalo přibližně 87s. Rozdíl v těchto parametrech oproti regálovému skladu však výrazně klesá se skladovou výškou AutoStore skladu. To bylo v simulační studii demonstrováno na případě skladu o výšce 5-ti a 10-ti boxů

nad sebou, kdy AutoStore sklad o výšce 10ti boxů odbavoval požadavky okolo 70s. Pro případ AutoStore skladu o výšce 10-ti boxů muselo dojít také k navýšení počtu operátorů o jednoho, aby nedocházelo ke kumulaci požadavků ve frontě. Tím byla výrazně snížena vytíženost operátora (z 60% na 30%). Navýšení počtu operátorů a jejich poměrně malé vytížení je možné kompenzovat variantou, kde jedno pracoviště odbavení bude odbavovat více požadavků najednou. Na druhou stranu u této varianty v některých simulačních bězích docházelo k přetížení pracoviště, což se projevilo zejména na zvýšení průměrné maximální doby odbavení jednoho požadavku (z 151s na 193s) a průměrné doby, kterou požadavek strávil ve frontě (ze 4s na 21s) pro konfiguraci se 4-mi roboty.

Provozní parametry AutoStore skladů lze zlepšit umístováním boxů s častěji poptávaným zbožím na horní pozice. Výsledky pro takto setříděný a nesetříděný AutoStore sklad jsou diskutovány v podkapitole 11.1. Pro setříděné sklady (vzhledem k nesetříděným) došlo pro jednotlivé varianty a konfigurace k poklesu hlavně ve vytíženosti robotů (v průměru o 13,1%), průměrné době pro odbavení požadavku (v průměru o 23,9%), maximální době odbavení požadavku (v průměru o 14,3 %) a maximální době požadavku ve frontě (v průměru o 36,1%).

Velkou výhodou AutoStore skladu je také úspora skladovacího prostoru. Například AutoStore sklad o počáteční výšce 10-ti boxů zabírá plochu 83,6 m^2 , AutoStore sklad o počáteční výšce 5-ti boxů zabírá plochu 163,2 m^2 , zatímco regálový sklad o stejném počtu skladovacích boxů (2470) zabírá plochu 561 m^2 .

Z výsledků experimentů lze konstatovat, že sklady typu AutoStore jsou pro podniky velmi atraktivním řešením zejména díky nižším osobním nákladům a úspoře skladovací plochy. Pro investici do skladu typu AutoStore je však nutné brát zřetel zejména na velikost skladu a jeho zvolenou výšku. Pro větší výšky skladů nemusí dojít k výraznému zlepšení parametrů (zejména průměrné doby pro odbavení požadavku). Lze se domnívat, že čím větší bude poměr plochy skladu ku jeho výšce, tím lepšího výsledku bude AutoStore sklad dosahovat, protože jezdové rychlosti robotů jsou větší než průměrné rychlosti chůze operátorů. Pro větší výšky skladu by měl také podnik zvážit možnost realizovat odbavovací systém, který operátorovi umožňuje odbavovat více požadavků najednou.

Z výsledku simulační studie plyne, že sklad AutoStore má svá určitá omezení, a proto autor doporučuje jeho použití za předpokladu důkladné analýzy podmínek, za jakých bude tento sklad používán.



Literatura

- [1] BANKS, Jerry. Handbook of Simulation - Principles, Methodology, Advances, Applications, and Practice. John Wiley & Sons, 1998 [cit. 2022-05-01]. ISBN 978-0-471-13403-9.
- [2] MANLIG, František. Využití počítačové simulace výrobních systémů. Liberec: Technická univerzita v Liberci, 2014 [cit. 2022-05-01]. ISBN 978-80-7494-162-7.
- [3] NANCE, Richard a Charles OVERSTREET. History of computer simulation software: An initial perspective. Winter Simulation Conference (WSC). 2017 [cit. 2022-05-01], 243-261. Dostupné z: doi:10.1109/WSC.2017.8247792
- [4] TORCHER, K. D. a D. G. OWEN. The automatic programming of simulations. Journal of Simulation. Taylor & Francis, 2008 [cit. 2022-05-01], 2(3), 143-152. Dostupné z: doi:10.1057/jos.2008.13
- [5] LAW, Averill a David KELTON. Simulation modeling and analysis. 3rd. New York: McGraw-Hill, 2007 [cit. 2022-05-01]. ISBN 978-00-7110-051-9.
- [6] DLOUHÝ, Martin. Simulace podnikových procesů. 2., upr. vyd. Brno: Computer Press, 2011 [cit. 2022-05-01]. ISBN 978-80-251-3449-8.
- [7] ROSSETTI, Manuel D. Simulation modeling and Arena. Second edition. Hoboken, New Jersey, 2016 [cit. 2022-05-01]. ISBN 978-111-8607-916.
- [8] GALGONEK, Jakub a kol. Generátory pseudonáhodných čísel. Sborník příspěvků, fyzikální týdny na FJFI ČVUT [online]. 2001 [cit. 2022-05-01], 1-2. Dostupné z: <https://tydenvedy.fjfi.cvut.cz/2001/web/sbornik>
- [9] ULRYCH, Zdeněk a Pavel RAŠKA. Modelování a simulace a DP [DVD-ROM]. Plzeň: Smartmotion, 2012 [cit. 2022-05-01]. ISBN 978-80-87539-15-6.
- [10] NEPOR, Vladimír. Rychlý průvodce simulacemi a simulačním softwarem. VRAPPS.com [online]. 2021 [cit. 2022-05-01]. Dostupné z: <https://www.vrapps.cz/blog/simulace-a-simulacni-software>

- [11] Game4automation. *Game4automation.com* [online]. in2Sight, c2019 [cit. 2022-06-01]. Dostupné z: <https://game4automation.com/en/virtual-commissioning-with-unity>
- [12] Arena Simulation Software. *Rockwellautomation.com* [online]. c2022 [cit. 2022-06-01]. Dostupné z: <https://www.rockwellautomation.com>
- [13] Plant Simulation & Throughput Optimization. *www.plm.automation.siemens.com* [online]. Siemens, c2022 [cit. 2022-06-01]. Dostupné z: <https://www.plm.automation.siemens.com/global/en/products/manufacturing-planning/plant-simulation-throughput-optimization.html>
- [14] Visual Components: 3D manufacturing simulation software. *Visual-components.com* [online]. c2022 [cit. 2022-07-12]. Dostupné z: <https://www.visualcomponents.com/>
- [15] 3D Simulation Modeling and Analysis Software. *FlexSim.com* [online]. Orem (Utah): FLEXSIM SOFTWARE PRODUCTS, c2022 [cit. 2022-06-01]. Dostupné z: <https://www.flexsim.com>
- [16] SCHOLZ, Pavel. Počítačová simulace ve výrobě - nepublikované přednášky. Praha, 2021 [cit. 2022-06-01].
- [17] HOUŠKA, Milan. Simulační modely I. V Praze: Česká zemědělská univerzita, Provozně ekonomická fakulta, 2005 [cit. 2022-06-01]. ISBN 978-80-213-1334-7.
- [18] KOCHENDERFER, Mykel J. a Tim A. WHEELER. Algorithms for optimization. Cambridge, Massachusetts: The MIT Press, 2019 [cit. 2022-06-01]. ISBN 978-0262039420.
- [19] SIXTA, Josef a Václav MAČÁT. Logistika: teorie a praxe. Brno: CP Books, 2005 [cit. 2022-06-01]. Business books (CP Books). ISBN 80-251-0573-3.
- [20] LAMBERT, Douglas M. a Lisa M. ELLRAM. Logistika: příkladové studie, řízení zásob, přeprava a skladování, balení zboží. Praha: Computer Press, 2000 [cit. 2022-06-01]. Business books (Computer Press). ISBN 80-722-6221-1.
- [21] HÁJEK, Dominik. Skladování v Podniku pro výrobu vajec v Kosičkách s.r.o. Pardubice, 2021 [cit. 2022-06-01]. Diplomová práce. Univerzita Pardubice.
- [22] STAZZONE, Shelly. 6 Types of Warehouse Storage Systems. In: *Camcode.com* [online]. Claveland: CAMCODE, 2022 [cit. 2022-06-07]. Dostupné z: <https://www.camcode.com/blog/types-of-warehouse-storage-systems/>

- [23] Česká logistika: Sklad [online]. *Ceskalogistika.cz*, c2022 [cit. 2022-06-07]. Dostupné z: <https://www.ceskalogistika.cz/sklad/>
- [24] AZADEH, Kaveh, M. B. M. DE KOSTER a Debjit ROY. Robotized Warehouse Systems: Developments and Research Opportunities. SSRN Electronic Journal [online]. [cit. 2022-06-07]. ISSN 1556-5068. Dostupné z: [doi:10.2139/ssrn.2977779](https://doi.org/10.2139/ssrn.2977779)



Příloha A

Obsah CD

1. diplomovaprace.zip
 - soubory diplomové práce v \LaTeX
 - soubor diplomové práce v pdf
2. modely.zip
 - soubory simulačních modelů vytvořené v softwaru FlexSim Express 2021.