

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STROJNÍ
ÚSTAV ŘÍZENÍ A EKONOMIKY PODNIKU



DIPLOMOVÁ PRÁCE

Simulace interní logistiky ve výrobním podniku

Simulation of internal logistics in a manufacturing company

AUTOR: Jan Tomek

STUDIJNÍ PROGRAM: Řízení průmyslových systémů

VEDOUCÍ PRÁCE: Ing. Pavel Scholz

PRAHA 2022

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Tomek** Jméno: **Jan** Osobní číslo: **459648**
Fakulta/ústav: **Fakulta strojní**
Zadávající katedra/ústav: **Ústav řízení a ekonomiky podniku**
Studijní program: **Řízení průmyslových systémů**
Specializace: **Bez specializace**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Simulace interní logistiky ve výrobním podniku

Název diplomové práce anglicky:

Simulation of Internal Logistics in a Manufacturing Company

Pokyny pro vypracování:

1. Úvod – zdůvodnění zadání, cíle a úkoly práce
2. Teoretická část - logistika, simulace a simulační nástroje, tvorba simulačního projektu, popis simulačního nástroje použitého v praktické části
3. Praktická část – charakteristika problému, tvorba koncepčního a simulačního modelu, simulace a analýza současné situace, simulace ověřovaných scénářů a jejich vyhodnocení
4. Závěry a doporučení

Seznam doporučené literatury:

1. 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.
2. KRAJEWSKI, Lee, Naresh MALHOTRA a Larry RITZMAN. Operations Management: Processes and supply chains. 13th. Ohio: Case Western Reserve University, 2021. ISBN 978-1-292-40994-8.
3. MANLIG, František. Využití počítačové simulace výrobních systémů. 1. vyd. Liberec: Technická univerzita v Liberci, 2014. 99 s. ISBN 978-80-7494-162-7.
4. 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.

Jméno a pracoviště vedoucí(ho) diplomové práce:

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

Jméno a pracoviště druhé(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

Prohlášení

Prohlašuji, že jsem tuto práci vypracoval samostatně, a to výhradně s použitím pramenů a literatury, uvedených v seznamu citovaných zdrojů.

V Praze dne:

.....
Podpis

Anotace

Tato diplomová práce se zabývá problematikou interní logistiky. Blíže je přiblížena problematika skladovacích systémů. Práce se dále zabývá simulacemi – jejich historickým vývojem, přednostmi a omezeními, zaváděním v podniku a tvorbou simulačních projektů. Následuje popis důležitých pojmů a funkcí vybraného simulačního softwaru, který je použitý v praktické části. Praktická část diplomové práce je věnována tvorbě simulačního modelu v softwaru FlexSim, kde je simulováno skladování dvou stupňů výroby. Na základě tohoto modelu a provozních dat jsou navrženy indikátory pro zhodnocení současného stavu.

Klíčová slova

Simulace, Modelování, Diskrétní simulace, Interní logistika, Skladování

Annotation

This master thesis deals with the issue of internal logistics. The problem of warehouse systems is presented in more detail. The thesis also deals with simulations - their historical development, advantages and limitations, implementation in the enterprise and the creation of simulation projects. This is followed by a description of important concepts and functions of the selected simulation software used in the practical part. The practical part of the thesis is devoted to the creation of a simulation model in the FlexSim software, where two stages of production logistics are simulated. On the basis of this model and operational data, indicators are proposed to evaluate the current.

Keywords

Simulation, Modeling, Discrete simulation, Internal logistics, Warehousing

Poděkování

Ve své práci bych rád poděkoval svému vedoucímu diplomové práce Ing. Pavlovi Scholzovi za vedení, trpělivost, ochotu a odbornou pomoc s vypracováním této závěrečné práce, Martinovi Krausovi ze společnosti AIMTEC a.s. za ochotu a hodnotné rady.

Seznam zkratek

Obsah

	Logistika.....	13
	1.1 Logistické aktivity	13
	1.2 Logistické systémy.....	15
	1.3 Výrobní logistika.....	15
1	1.3.1 Typy výrobních procesů	16
	1.4 Materiálový tok.....	16
	1.4.1 Sankeyův diagram	17
	1.4.2 Spaghetti diagram	18
	1.4.3 Postupový diagram.....	19
	1.4.4 Value stream mapping	19
	1.5 Role optimalizace	20
	Skladovací systémy.....	21
2	2.1 Pojetí a funkce skladů.....	21
	2.2 Přínosy a náklady skladů	22
	2.3 Popis skladovacího systém.....	24
	2.4 Regálové systémy	24
	2.4.1 Policové regály	25
	2.4.2 Paletové regálové systémy.....	25
	2.4.3 Průjezdové (konzolové) regály	26
	2.4.4 Spádové (gravitační) regály	27
	2.5 Optimalizace ve skladech	28
	2.5.1 KPI.....	28
3	2.5.2 Oblasti optimalizace.....	29
	2.5.3 Využití simulace pro optimalizaci	32
	Simulace podnikových procesů	33
	3.1 Historie simulací	33
	3.2 Simulace a modelování	34
	3.3 Diskrétní simulace (DES).....	36
	3.4 Přednosti a omezení simulací.....	36
	3.4.1 Přednosti	36
	3.4.2 Omezení	38
	3.5 Oblasti využití simulací.....	39
	3.6 Zavádění simulačních nástrojů v podniku	42

	3.7	Tvorba simulačního projektu	44
		Simulační nástroj FlexSim.....	47
	4.1	Přehled uživatelského rozhraní.....	47
	4.2	Základní objekty 3D modelu.....	48
	4.3	Procesní tok.....	49
4	4.3.1	Token.....	50
	4.3.2	Aktivity.....	51
	4.3.3	Sdílené zdroje.....	51
	4.3.4	Konektory.....	52
	4.3.5	Složené bloky.....	52
		Úvod do praktické části.....	53
	5.1	O společnosti.....	53
5	5.2	Charakteristika řešeného problému	54
		Koncepční model a sběr dat.....	54
6		Simulační model – první fáze	56
7	7.1	Počáteční naskladnění.....	58
	7.2	Výroba na vstřikolisech	63
	7.3	Odvoz vozíků od lisů do gravitačních regálů	67
	7.4	Odvoz vozíků z regálu ke kaširovacím lisům	71
8		Simulační model – druhá fáze	72
	8.1	Počáteční naskladnění prázdných vozíků.....	75
	8.1.1	Zpětný proud – levá část	75
	8.1.2	Zpětný proud – pravá část.....	78
	8.2	VZV1: vstřikolisy a gravitační regály	81
	8.2.1	VZV1 – společná část.....	81
9	8.2.2	VZV1 – levá část.....	82
	8.2.3	VZV1 – pravá část	86
10	8.3	VZV2: gravitační regály a kaširovací lisy	90
		Verifikace a validace.....	94
	9.1	Verifikace.....	94
	9.2	Validace	94
		Návrh klíčových indikátorů výkonnosti a jejich analýza.....	94
	10.1.1	Vytíženost gravitačních regálů	95
	10.1.2	Vytíženost zpětných regálů	98

10.1.3	Výstup ze vstřikolisů.....	100
10.1.4	Využití přepravní vozíky	102
	Závěr	103
	Seznam obrázků.....	105
	Seznam grafů.....	107
	Seznam tabulek.....	107
	Seznam použité literatury.....	108

Úvod

Vysoké požadavky na efektivní výrobu a logistiku vytvářejí konstantní tlak na společnosti. Prudký technologický vývoj roste rychleji, než znalosti lidí, a proto je již běžnou praxí využívání různých softwarů pro řízení a plánování výroby a skladů. Z hlediska optimalizačního se stále častěji využívá simulačních modelů. S rostoucí dostupností simulačního softwaru klesají náklady analýz za pomoci simulačních modelů. Všechny tyto kroky pak mohou podnikům poskytovat cennou konkurenční výhodu, jelikož v globálním měřítku je konkurence výrobních podniků obrovská.

Z výše uvedených důvodů je zpracována i tato diplomová práce. Jejím cílem je navrhnout simulační model skladovacího systému včetně klíčových indikátorů výkonnosti pro řízení a vyhodnocování provozu tohoto systému. Vytvořený simulační model pak bude v zadavatelském podniku sloužit jako nový nástroj na podporu řízení v rámci současného řešení.

První kapitola teoretické části diplomové práce se zabývá logistikou, jejím definováním, dělením a také významem. Část kapitoly se blíže věnuje materiálovému toku a různým nástrojům pro jeho analýzu. Závěrem kapitoly je přiblížení zdrojů plýtvání. Druhá kapitola přibližuje problematiku skladovacích systémů, jako je jejich funkce, náklady a přínosy. V této části je také přiblížena optimalizace ve skladech a případné využití simulace pro optimalizaci. Třetí kapitola se věnuje historii a vývoji počítačové simulace a simulačním modelům. Dále se věnuje přednostem a omezením simulací, oblastem využití a také zaváděním simulačních nástrojů v podniku. Kapitulu uzavírá popis tvorby simulačního projektu.

Čtvrtá kapitola slouží jako přechod mezi teoretickou a praktickou částí. Tato kapitola je věnována vybranému simulačnímu softwaru FlexSim, který je v praktické části využit pro vytvoření simulačního modelu. Ve čtvrté kapitole jsou přiblíženy důležité funkcionality

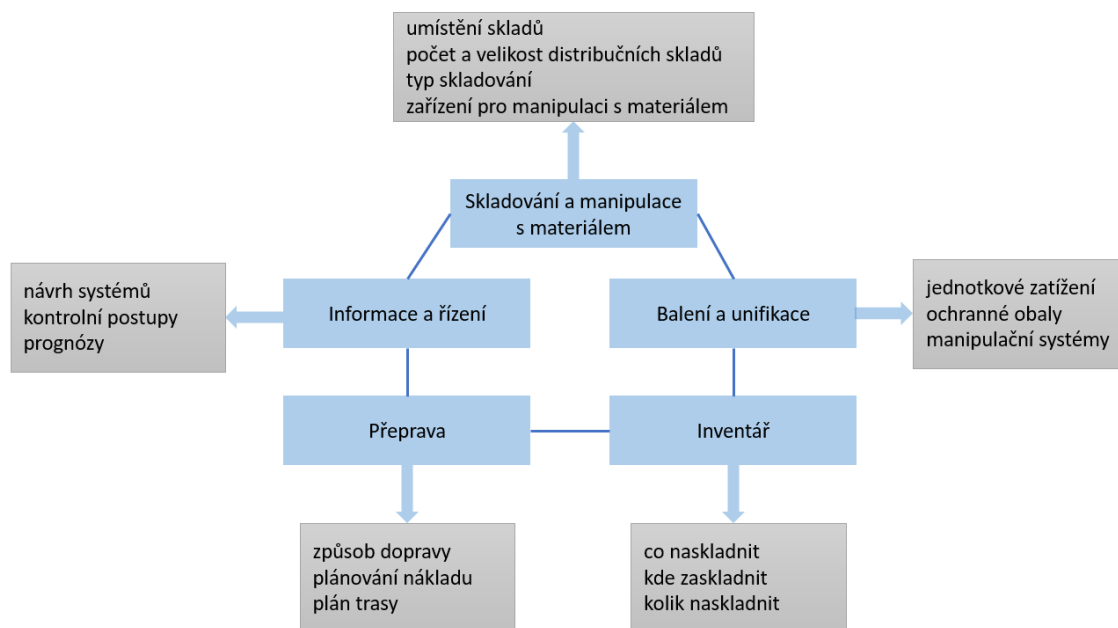
vybraného softwaru, konkrétně dělení 3D modelu a procesního toku. Jsou zde popsány i aktivity procesního toku, na jejichž základě je tvořena logika simulačního modelu.

Praktická část začíná krátkým úvodem do problematiky řešeného problému a představením vybraného podniku, této části je věnována kapitola pět.

V kapitole šest je definován koncepční model, využité softwary a jednotlivé kroky tvorby simulačního modelu. Kapitola sedmá se věnuje tvorbě první fáze simulačního modelu, konkrétně přenesení koncepčního modelu do simulačního softwaru. Je zde popsána kompletní logika pro řízení tohoto modelu. V kapitole osmé je simulační model rozšířen o zpětný proud, jsou zde upraveny některé procesy a aktivity a opět je popsáno veškeré nastavení logiky modelu. V následující kapitole devět je simulační model verifikován a validován. V kapitole deset jsou navrženy klíčové indikátory a jejich analýza na základě simulačního modelu a provozních dat. Součástí analýzy je závěrečné doporučení. Tím je praktická část uzavřena.

Logistika

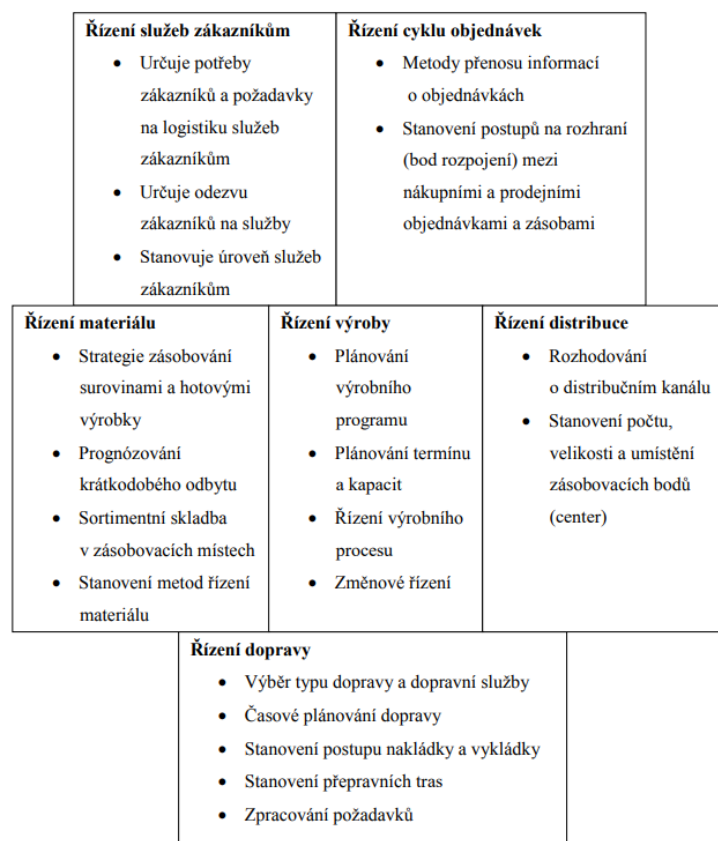
1 Klíčové složky logistiky - doprava, zásoby, skladování – tvoří základní prvky průmyslového a hospodářského života již řadu let, ale teprve v posledních zhruba 20 letech byla logistika uznána jako samostatná a významná oblast. Hlavním důvodem, proč k tomuto uznání došlo teprve relativně nedávno, je povaha logistiky jako takové. Jedná se o oblast složenou z mnoha dílčích funkcí a subsystémů, z nichž je každý stále považován za samostatnou řídicí operaci. Konsenzus v dnešní době je, že je třeba přijmout ucelenější pohled na tyto různé operace, aby bylo možné zohlednit jejich vzájemnou provázanost a interakci. Uvědomění si rozsahu a významu logistiky a dodavatelského řetězce vedlo k tomu, že se k tomuto tématu přistupuje více vědecky. Tento přístup je zaměřen na pojetí logistické funkce jako celku, ale co je důležité, zahrnuje i vzájemné vztahy jednotlivých subsystémů. Velká část tohoto přístupu se zabývá potřebou a způsoby plánování logistiky a dodavatelského řetězce. [1]



Obrázek 1: Klíčové prvky logistiky a distribuce [1]

1.1 Logistické aktivity

Logistické aktivity, které spadají pod logistický systém, můžeme rozdělit na klíčové aktivity a na podpůrné aktivity. Klíčové aktivity se realizují v každém logistickém kanálu, podpůrné aktivity se realizují ve vybraných společnostech dle konkrétních okolností. Tyto klíčové aktivity se soustřeďují do procesů řízení. Tyto procesy jsou na následujícím obrázku. [31]



Obrázek 2: Klíčové aktivity logistiky [31]

Podpůrné aktivity mohou být v některých společnostech stejně důležité jako aktivity klíčové. Jedná se o následující:

- Skladování – rozmístění zásob a návrh míst pro nakládání, konfigurace skladů, umístění skladovacích systémů v dostupném prostoru.
- Manipulace s materiálem – výběr vhodných zařízení, strategie rozhodování o zařízení, postupy výběru objednávek, uskladnění a vyzvedávání zásob, rozmístění zdrojů.
- Nákup – výběr zdroje dodávek, časové rozvržení nákupů, nakupovaná množství.
- Balení – návrh manipulace, návrh skladování, návrh ochrany před ztrátou či poškozením.
- Správa informací – sběr, ukládání a nakládání s informacemi a analýza dat. [31]

1.2 Logistické systémy

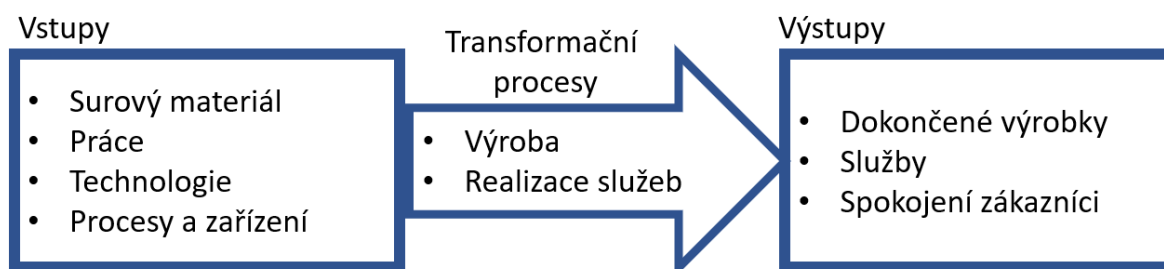
Logistiku můžeme také pojmut jako určitý obor činností, ve kterém je několik vnitřních systémů, které zabezpečují činnost logistických funkcí. Tyto systémy jsou ve své podstatě tři:

- Zásobovací logistický systém, kterým navazuje podnik na trh ve funkci odběratele. Prostřednictvím trhu zajišťuje tento systém hmotný tok surovin, materiálu a subdodávek vybranými dodávkovými cestami od dodavatelů do výrobního procesu
- Výrobní logistický systém, který zabezpečuje pohyb surovin a subdodávek ve výrobním procesu: materiál, polotovary a hotové výrobky jsou manipulovány mezi výrobními operacemi a jsou skladovány ve výrobních skladech.
- Odbytový logistický systém, kterým navazuje podnik cestu zpět na trh, ale tentokrát jako dodavatel. Tento systém zabezpečuje tok hotových výrobků přes odbytové a distribuční sklady a případně přes zprostředkovatele k cíli (k odbytovému trhu). [37]

1.3 Výrobní logistika

Smyslem výrobní logistiky je zajištění dostačujícího a pravidelného přísunu materiálu k zabezpečení výrobního procesu a zajištění plynulého odsunu hotových výrobků do skladu a jejich následná expedice. [37]

Výrobní logistika zahrnuje všechny činnosti, které jsou součástí procesu transformace. Mezi ně může patřit navrhování procesů, organizace lidských zdrojů, plánování a řazení výroby, kontrolu kvality, řízení výkonnosti a realizaci strategie organizace. Základní transformační diagram je na následujícím obrázku. [1]



Obrázek 3: Základní transformační diagram vstupů a výstupů [1]

Úspěch výrobní logistiky je závislý na dvou faktorech:

- Na kvalitě a propracovanosti výrobního plánu.
- Na kvalitě návaznosti logistického systému dodávek materiálu do výroby a odběru výrobků z výrobní sféry. [37]

1.3.1 Typy výrobních procesů

Objem vyráběného produktu a různorodost produktů určuje nejvhodnější typ procesu. Zároveň mají tyto typy ovlivňují logistiku např. využití manipulační prostředky, vytížení skladových systémů apod. Typy procesů pro výrobky lze rozdělit do následujících kategorií takto:

- Projektové - velmi nízký objem (jednorázový) a vysoká rozmanitost. Obvykle velký rozsah, složitost a výrobek je stacionární.
- Zakázkové - malý objem a vysoká rozmanitost, například vytvoření výrobku na míru pro zákazníka.
- Dávkové - střední objem a střední rozmanitost.
- Hromadné - velký objem a malá rozmanitost.
- Kontinuální - velmi vysoký objem a velmi nízká rozmanitost. [1]

1.4 Materiálový tok

Materiálový tok je středem logistických procesů podniku. Jedná se řízený pohyb materiálu, surovin, polotovarů, který umožňuje charakterizovat dynamiku výroby v prostoru a čase. Rozvržení výrobních zařízení a pracovních jednotek ovlivňuje materiálový tok. Prostřednictvím vhodného rozvržení strojů, pracovních úseků, skladů a samotných budov lze dosáhnout značných úspor samotného materiálu, času i finančních prostředků.

Průběh a realizaci materiálové toku ovlivňuje:

- Druh a typ výrobního procesu, objem, sortiment.
- Úroveň technologické složitosti a členitosti výrobních procesů, montáže.
- Počet operací uskutečňovaných v jednotlivých fázích výrobního procesu.
- Tvar, rozvržení a specifika prostoru výrobního procesu.
- Způsob řešení přepravy.

- Umístění podpůrných služeb provozu – údržba, výdej náradí.

Pro analýzu a projektování manipulace a následného toku je nutné rozložení procesu manipulace na menší části:

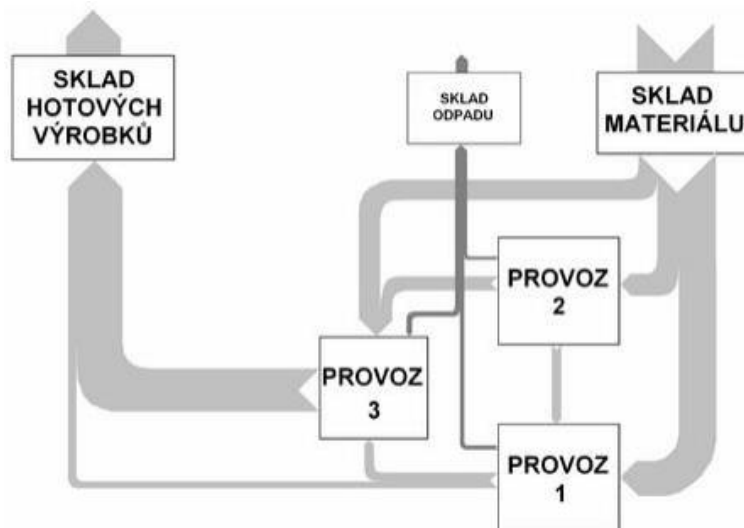
- proces
- subprocess
- operace
- úkon
- pohyb.

Základní jednotkou manipulace je operace, kterou lze definovat jako záměrnou, nepřetržitou změnu polohy předmětu, kterou uskutečňuje manipulační prostředek či pracovník, popř. oba dohromady.

Při analýze materiálového toku je vhodné se soustředit na nejdůležitější přesuny materiálu mezi jednotlivými místy vstupu a výstupu materiálu. Zkoumá se efektivnost pohybu materiálu v rámci jednotlivých částí výrobního procesu, což znamená, že dochází ke znázornění podstatných požadavků výrobních, dopravních, manipulačních a skladovacích procesů a jejich vzájemných vazeb s cílem odhalit úzká místa a určit způsob a rámec možné optimalizace. S analýzou materiálového toku pomáhají různé druhy nástrojů jako Sankeyův diagram, Spaghetti diagram a další, které jsou popsány v následujících podkapitolách. [2]

1.4.1 Sankeyův diagram

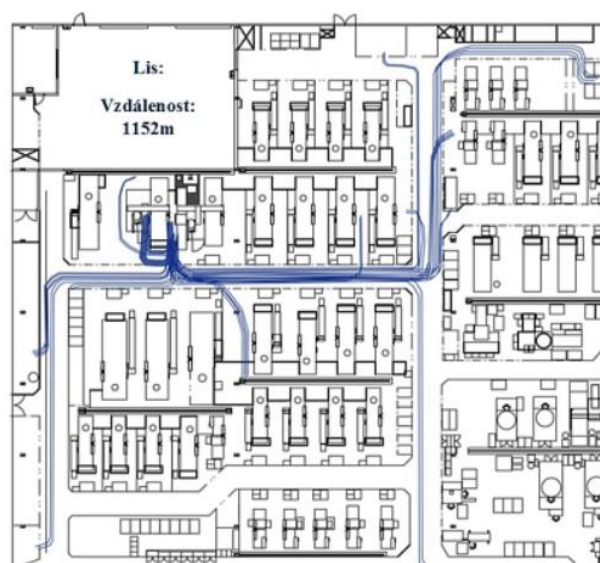
Jeden z nejnáměšších a nejpoužívanějších způsobů vizualizace materiálového toku v podniku. Principem je grafické znázornění struktury a časového průběhu vybrané veličiny v daném systému. Šířka čar (šipek) je přímo úměrná velikosti průtoku. V současné době je použití tohoto diagramu usnadněno použitím softwarových aplikací. Příklad tohoto diagramu je na následujícím obrázku. [2]



Obrázek 4: Sankeyův diagram [33]

1.4.2 Spaghetti diagram

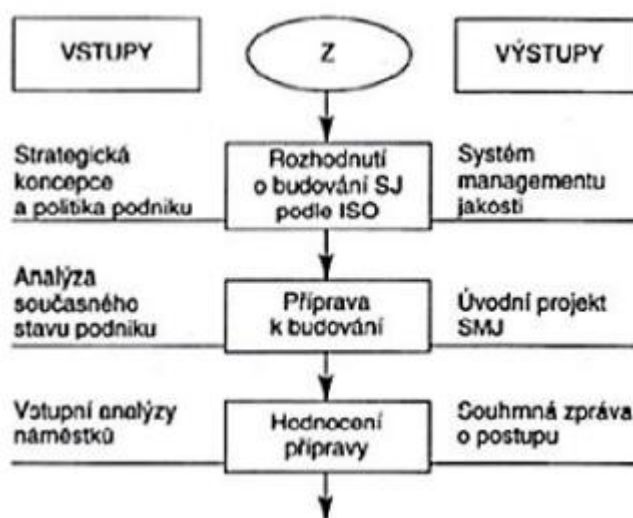
Jedna z nejjednodušších metod analýzy materiálového toku, která se používá při mapování interního toku a hledání nejvhodnější přepravní cesty nebo návrhu rozvržení pracoviště. Tento diagram je založen na principu přesného zakreslení každého pohybu pracovníka na určeném pracovišti a určeném intervalu. Barevně se rozlišuje typ cesty – zbytečná cesta, cesta bez plného vytížení, cesta s plným vytížením. V současné době je možné opět využít podporu a různých elektronických zařízení, které tuto cestu zaznamenávají. Na následujícím obrázku je příklad tohoto diagramu. [2]



Obrázek 5: Spaghetti diagram [32]

1.4.3 Postupový diagram

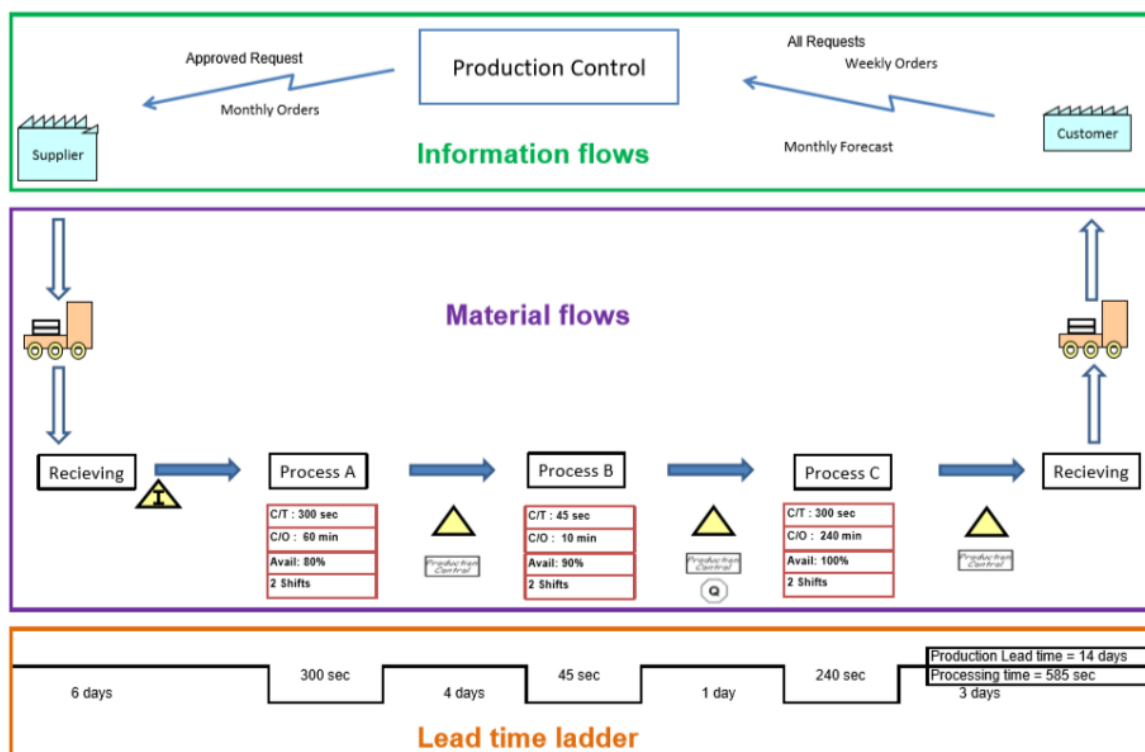
Hlavním cílem tohoto diagramu je znázornění posloupnosti všech manipulačních, technologických a kontrolních operací, které jsou prováděny na určitém výrobku nebo výrobní dávce. Může být zároveň využito i u nevýrobních operací, jako jsou např. objednávky. Tento diagram využívá vlastní sadu symbolů. Výsledkem je kvantifikace, proporcionalita a délka každé operace. Slouží k posouzení vhodnosti kooperace mezi jednotlivými pracovišti, identifikaci činností přinášejících hodnotu a to pomocí mapování a racionalizace procesů. Postupový diagram vstup/výstup je na následujícím obrázku. [2]



Obrázek 6: Vývojový diagram [34]

1.4.4 Value stream mapping

Tuto metodu vyvinula společnost Toyota a v rámci přístupu ke štíhlým procesům je hodnotným nástrojem. Využívá speciálních symbolů, které se mohou dělit do kategorií podle toho, jaký typ toku zobrazují – např. informační nebo materiálový. Popisem procesů, které přidávají, nebo nepřidávají hodnotu je možné identifikovat místa rezerv a plýtvání ve výrobním procesu. Příklad VSM je na následujícím obrázku. [2]



Obrázek 7: Value stream mapping [35]

1.5 Role optimalizace

Úspora nákladů, efektivní výroby a logistické procesy mají vliv na konkurenceschopnost podniků. Je důležité se zaměřovat na zajištění efektivity logistických procesů a v této souvislosti je důležitá štíhlá logistika. Hlavním cílem štíhlé výroby a logistiky je návrh nových procesů a změna těch stávajících tak, aby byla zkrácena celková doba výroby a aby byly odstraněny zdroje plýtvání. To má za výsledek růst produktivity práce a pokles nákladů na výrobu.

Identifikujeme několik zdrojů plýtvání:

1. **Nadvýroba**, která vzniká předčasnou výrobou, dříve, než požadují zákazníci nebo výrobou většího množství, než požadují zákazníci. Za hlavní příčinu můžeme považovat nesrovnalost mezi proměnlivými požadavky zákazníků a snahou o to vyrábět co největší výrobní dávky. To vede k dalšímu zásadnímu zdroji plýtvání a tím jsou zbytečné zásoby.

2. **Zbytečné zásoby** mají za následek zvýšení nákladů spojených s držbou zásob, kde dochází ke ztrátám právě kvůli držbě a také dochází ke ztrátám způsobeným navázáním kapitálu v zásobách. Zbytečné zásoby mají mimo jiné za následek zbytečné manipulační procesy, které plýtvají dostupné zdroje.
3. **Realizace zbytečných činností**, které zahrnují činnosti nad rámec technologických a pracovních instrukcí, činnosti, které nemají žádnou hodnotu pro zákazníka.
4. **Zbytečné prostoje** zahrnují čekání pracovníků a strojů z důvodu nedostatku materiálů a dílů. Důsledkem jsou pak nenahraditelné kapacitní ztráty ze špatného řízení materiálových toků. Dalším příkladem je také čekání hotových výrobků v důsledku dřívějšího ukončení jejich výroby v distribučních skladech.
5. **Zbytečná doprava a manipulace**, která obsahuje zpětnou dopravu výrobků a dílů na výchozí pozice, dlouhé přepravní cesty z důvodu neefektivního layoutu výrobních hal a zboží ve skladech, které vede k růstu nákladů. Roste také pravděpodobnost rizika poškození těchto přepravovaných dílů či zboží.
6. **Opravy**, především neplánované, a odstraňování vad, mají za následek ztrátu času, který je nutný pro realizaci činností, které mají přidanou hodnotu pro zákazníka. [3]

2

Skladovací systémy

Sklady různých provedení a typů jsou i v dnešní době nerozlučnou součástí moderních dodavatelských řetězců i přesto, že znamenají dočasné přerušení toku materiálu a z toho plynoucí nutnost držení zásob. Sklady jsou součástí různých míst celého dodavatelského řetězce – u výrobců, distributorů, v prodejnách a v některých případech i u koncových zákazníků. Toto rozmístění je důležité pro efektivní realizaci logistických činností v nákupu, výrobě, distribuci a oblasti zpětného toku.

2.1 Pojetí a funkce skladů

Zásadní změnou v souladu se zvyšováním pružnosti dodavatelských systémů je změna orientace řízení skladovacích systémů. Historicky sklady zastávaly funkci zásobníku, který absorboval dle plánu vytvářené výrobky, polotovary, suroviny. Z pohledu materiálových toků se jednalo o aplikaci principu tlaku, kdy je sklad místem, kde končí podle plánu tlačným způsobem výrobky vytvářené v předcházejících prvcích dodavatelského systému ve formě zásob. Role skladu tedy byla převážně pasivní.

Dnešní pojetí skladů spočívá v jeho vymezení jako poskytování vyšší úrovně služeb jeho zákazníkům, tedy v tom, že činnosti realizované ve skladovacím systému zvyšují hodnotu pro navazujícího partnera v dodavatelském systému. To znamená, že i sklad obdobně jako výrobce a další prvky dodavatelského systému vychází při realizaci dodávek z požadavků zákazníka, tedy v operativním řízení toků zboží ve skladu je stále více uplatňován princip tahu, zejména při řízení vstupů. Většina skladů však plní obě z uvedených funkcí. Na principu tlaku vytváří distribuční sklad podle odhadu budoucího vývoje pojistnou zásobu, podle principu tahu dle objednávek zákazníku vystavuje objednávky svým dodavatelům, kompletuje a expeduje dodávky.

Zatímco pojetí skladů se s vývojem fungování trhů postupně mění, tak pokud jde o jejich funkce, ty zůstávají v zásadě stále stejné. Především se jedná o funkce vyrovnávacího rázu. Mezi tyto funkce můžeme zařadit následující:

- Geografická – optimální lokalizace skladů v dodavatelském systému umožňuje přiblížení výrobků k centrům spotřeby. Jedná se o využití vhodně lokalizovaných skladů pro zvýšení konkurenceschopnosti v jednotlivých regionech, a to právě díky zkrácení termínu vyřízení objednávek.
- Sezónní – určené k překlenutí časového rozdílu mezi výrobou a spotřebou sezónních výrobků. Jedná se například o paliva či zemědělské produkty.
- Kapacitní – v celém dodavatelském systému, a i dílčích částech jako jsou výrobní, skladovací a přepravní systémy, dochází ke kapacitním rozporům. Může docházet k požadavku na dočasné skladování vznikajícího přebytku.
- Kompletační – tato funkce je dána sortimentním rozporem mezi relativně úzkým sortimentem výrobců a požadavky na obchodní síť na ucelené dodávky skládající se z velkého počtu položek dodávaných různými výrobci.
- Pojistná – v dodavatelském systému je nutné udržovat ve skladech pojistnou zásobu, která umožní reagovat na různé výkyvy. [3]

2.2 Přínosy a náklady skladů

Sklady mohou přinášet výhody dvou typů – zlepšení úrovně služeb a úsporu nákladů. Obě skupiny přínosů jsou blíže rozebrány dále.

Zlepšení úrovně služeb:

- Soustředování zásob do skladů pro kompletaci požadavků dalším článkům distribučního systému.
- Krátkodobé zásoby sezónního zboží umožňují soustředit toto zboží krátce před sezónou přímo do dislokovaných skladů nebo prodejen.
- Sklady udržující zásoby pro drobné podnikatele umožňují rychlé a levnější pořizování materiálových vstupů přímo v oblasti jejich působení. [3]

V rámci analýzy skladových nákladů je důležitá skutečnost, že většina uvedených bodů má povahu převážně fixních nákladů, nezávislých na skladovaném množství i obratu skladu.

Úspora nákladů:

- Utváří podmínky pro tvorbu hromadných objednávek. To má za následek úsporu přepravních nákladů díky plnému vytížení dopravních prostředků.
- Při využití kombinované dopravy je dosaženo úspory přepravních nákladů.
- Skladování sezónních surovin a výrobků znamená, že výrobce může vyrábět výrobky rovnoměrněji a tím odstranit nepříznivý vliv progresse fixních nákladů na jednotku produkce.
- Konečná úprava výrobků za využití skladů také nese výrazné úspory (např. výstupní kontrola v rámci distribučního skladu). [3]

Mezi hlavní náklady spojené se sklady a jejich provozem patří:

- Odpisy a náklady na údržbu vybavení skladu.
- Náklady na energie – osvětlení, zabezpečení skladovacích podmínek (chlazení, topení, klimatizace), provoz manipulačních prostředků.
- Náklady na obaly a fixační materiály.
- Náklady na manipulační prostředky – palety, kontejnery.
- Náklady na nakupované služby.
- Osobní náklady.
- Administrativní náklady, náklady na provoz informačního systému.

Odpisy a náklady na údržbu a osobní náklady jsou výše zmíněné fixní náklady. Náklady na energie a administrativní náklady potom obsahují výrazný podíl fixní složky.

K nákladům je nutné přičíst i ztráty způsobené manipulací ve skladu, nedodržení skladovacích podmínek, překročení povolené záruky apod. [3]

2.3 Popis skladovacího systému

Pro popis skladovacího systému se využívá jeho rozdělení na čtyři části:

1. Statická – pod statickou část spadají volné či zastřešené skladovací plochy, nádrže, soustavy, jednopodlažní i vícepodlažní budovy s různými typy regálových soustav.
2. Dynamická – sem spadají prvky, které v systému zabezpečují manipulační operace jako je příjem a uložení zboží, vyskladnění, balení a kompletace. Jedná se tedy o dopravníky, zakladače, výtahy.
3. Informační subsystém – v jednoduchém provedení slouží k evidenci skladových položek, jejich pohybu a související administrativě. V moderním provedení se jedná o systémy pro řízení skladů (Warehouse Management Systems), které jsou schopné řídit provoz skladu a podporovat rozhodovací procesy.
4. Pracovníci – členové managementu, vedoucí útvarů, operátoři, skladníci, ...

Rozložení, struktura, vybavení a použité skladovací technologie závisí především na tom, co budeme skladovat a v jaké podobě. Nejčastější skladovanou položkou je kusové zboží. Dalšími typy jsou pevné látky a ty jsou nejčastěji volně ložené, v kontejnerech, krabicích. Plynné látky, které jsou ukládány v podzemních zásobnících, stlačené v tlakových lahvích nebo samostatných plynojemech. V poslední řadě se jedná o kapalné látky, které jsou ukládány v nádržích a kontejnerech.

2.4 Regálové systémy

Převažujícím typem komerčních skladů jsou sklady umístěné uvnitř budov se zavedenými regálovými systémy. Níže jsou popsány nejčastější typy regálových systémů, mezi které patří policové regály, paletové regálové systémy, průjezdové (konzolové) regály a spádové (gravitační) regály.

2.4.1 Policové regály

Tento typ regálu je využíván zejména pro skladování kusového zboží o menších rozměrech a hmotnosti, skladování menších dílů v obalech, krabicích apod. Výhodou tohoto řešení je snadné přizpůsobení rozsáhlému sortimentu. Z hlediska manipulace není využívána drahá technika, jedná se především o systém s ruční obsluhou. Celkové využití skladovací plochy je nízké. V případě využití vysokých polic do 4,5 m sice o něco roste využití skladovací plochy, ale je již nutné pořídit pojízdný manipulační prostředek.



Obrázek 8: Policový regál [4]

2.4.2 Paletové regálové systémy

Jak již název napovídá, manipulační jednotkou je v tomto případě paleta. Jedná se o nejrozšířenější skupinu regálů umístěných v budovách. Tyto systémy jsou stavěny ve výškách od 7 do 45 m a šířka se odvíjí od použitého manipulačního prostředku (vysokozdvíhací vozík, regálový zakladač). Hloubka se pak odvíjí podle rozměru palet. V tomto regálovém systému je možné skladovat jakékoliv výrobky umístěné na paletách v různých obalech – od krabic až po sudy vážící několik set kilogramů. Systém je flexibilní,

lze snadno přestavět úložné úrovně. Oproti policovým regálům má další výhodu a tou je možnost nasazení automatizačních prostředků.



Obrázek 9: Paletový regál [4]

2.4.3 Průjezdové (konzolové) regály

Vysokého využití skladovacího prostoru lze dosáhnout výstavbou vjezdových nebo průjezdových regálů. Manipulační prostředky zajíždějí přímo do regálových uliček a ukládají palety na postranní lišty. Pro vjezdové regály platí, že lze manipulaci provádět pouze z jedné strany, u průjezdových jsou palety přístupné z obou stran regálového systému. Tento typ regálového systému dosahuje vysokého využití prostor. Nevýhodou tohoto řešení je nepřístupnost k jednotlivým paletám, navíc u vjezdového systému lze využít pouze systém LIFO (last in, first out).



Obrázek 10: Průjezdový regál [4]

2.4.4 Spádové (gravitační) regály

Spádové regály tvoří další možnost, jak zvýšit využití skladového prostoru. Zboží může být skladováno na paletách, ale v i jiných manipulačních obalech. Nakloněné regály jsou tvořeny válečkovými tratěmi o různé konstrukci a pohyb zboží zabezpečují gravitační síly. Zadní strana je vyhrazena pro vstup do regálů, po zasunutí se paleta (nebo jiná manipulační jednotka) postupně posouvá k čelní straně, odkud je vyskladňována. Mezi nevýhody patří riziko poruchy válečkové trati a také nutnost zabezpečení stálé rychlosti pojezdu v případě velké hmotnosti manipulační jednotky. [3]



Obrázek 11: Spádový regál [4]

2.5 Optimalizace ve skladech

Cílem optimalizace skladů je zvýšení výkonnosti podniku, pokud jde o výrobu řízenou poptávkou (tažný systém), zajištění její plynulosti a produktivity (Just-in-time přístup), minimalizování zásoby v dodavatelském řetězci, poskytování dalších služeb a snížení nákladů na dopravu.

2.5.1 KPI

V souvislosti s celosvětovým tlakem na efektivitu všech podnikových činností je třeba se zaměřovat i na optimalizaci ve skladech. V této souvislosti jsou klíčové ukazatele výkonnosti (KPI), bez jejichž znalosti by bylo nereálné optimalizaci provádět. V rámci skladu můžeme KPI rozdělit do těchto 4 oblastí:

1. spolehlivost
2. flexibilita
3. náklady
4. využití majetku

Spolehlivost zahrnuje včasné dodání, míru plnění objednávek a přesnost. Nejlepším měřítkem flexibility je pravděpodobně doba cyklu objednávky, protože zahrnuje všechny aspekty procesu objednávky zákazníka: způsob, jakým objednávku původně vyřizujeme, zda jsou k dispozici skladové zásoby, jak rychle můžeme objednávku zpracovat ve skladu a nakonec, jak rychle můžeme objednávku doručit zákazníkovi. V rámci měření nákladů můžeme zahrnout náklady jako procento z tržeb a produktivitu v poměru k pracovní době. Využití majetku pak bude zahrnovat efektivní využití skladových prostor, manipulačních zařízení, zaměstnanců a skladovacích zařízení. Využití skladu se obvykle měří množstvím využitých podlahových ploch. Je však reálnější měřit kubické využití budovy. Dalším ukazatelem může být počet využitých paletových míst oproti celkovému počtu míst, která jsou k dispozici. [5]

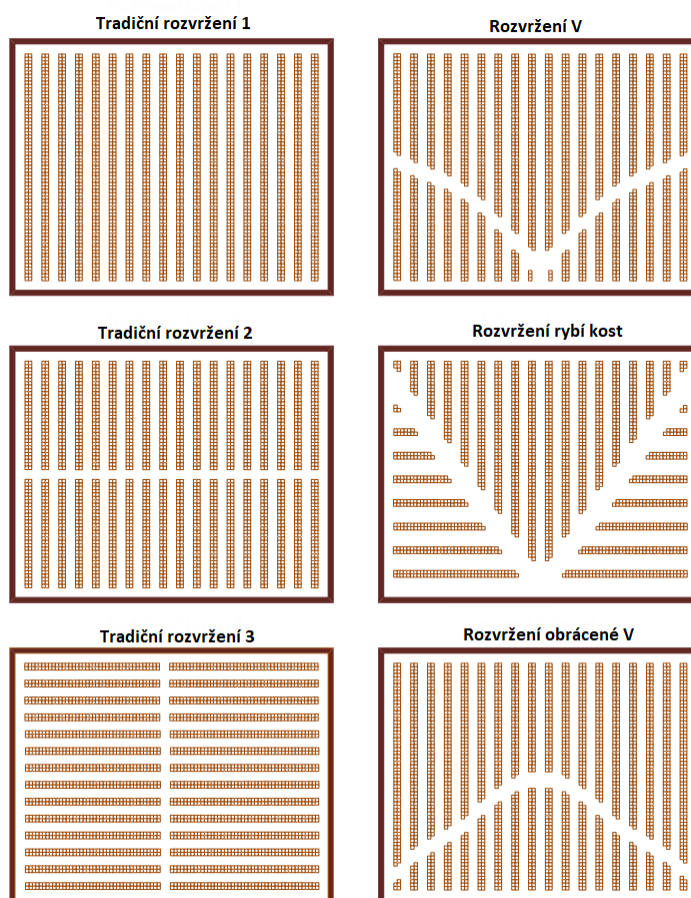
2.5.2 Oblasti optimalizace

Problematika optimalizace skladů se obecně zaměřuje zejména na 3 kategorie, kterými jsou optimalizace technické struktury, optimalizace provozní struktury a optimalizace řízení skladu. Jednotlivé kategorie jsou blíže popsány dále.

- Optimalizace technické struktury

Základní technická struktura zahrnuje např. návrh rozvržení logistického skladu nebo celého distribučního centra, výběr a dimenzování dopravníků a skladových zařízení, návrh fyzických rozhraní se sousedními systémy a další atributy související s technickou strukturou. Faktory, které je třeba vzít v úvahu při návrhu rozvržení, jsou následující:

počet bloků, délka, šířka a počet vychystávacích uliček, počet a tvar příčných uliček, pokud jsou ve skladu přítomny, počet úrovní regálů a umístění vstupních a výstupních bran ve skladu. Nejčastější typy rozvržení skladu v rámci jeho technického řešení je pak možné vidět na následujícím obrázku. [6]



Obrázek 12: Typy skladových rozvržení [6]

- Optimalizace provozní struktury

Provozní a organizační rámec kombinuje různé aspekty z mnoha oblastí, např. řízení podniku, řízení zásob, řízení organizace, řízení dopravy a mnoha dalších oblastí řízení.

V rámci provozní struktury se optimalizace zaměřuje např. na politiku přidělování skladů, které existují 2 a to náhodná a vyhrazená. Zatímco náhodná strategie umožňuje uložit paletu na libovolné prázdné místo se stejnou pravděpodobností nebo na nejbližší prázdné místo, vyhrazená strategie umožňuje uložit paletu pouze na určená místa. U vyhrazené strategie jsou pak skladovací místa často nějakým způsobem organizována. Organizace může být např. skladování podle tříd, kde je zboží seskupeno podle četnosti objednávek. Tato politika přiřazuje nejčastěji poptávané zboží na nejlepší (nejbližší) místa od vstupních/výstupních bran. Další možností je použití rodinného seskupení, kdy je zboží seskupeno podle vztahů nebo podobností mezi výrobky nebo objednávkami.

Další důležitou oblastí v rámci optimalizace provozní struktury je problematika vychystávání. Jednou z nejpoužívanějších metod vychystávání jsou jednotlivé objednávky. S tím je spojena skladová jednotka, která představuje nejmenší fyzickou jednotku výrobku, se kterou společnost manipuluje, např. krabice, druh bedny nebo kartonu skládající se z vnitřních obalů a krabic nebo jednotlivých kusů výrobku, ale ve velkých distribučních centrech to může to být pouze paleta. V souvislosti s vychystáváním je důležité se zaměřit na zásady směrování (routing), které by měly zajistit optimální cestu skladem pro vychystávání objednávek. Na obrázku č. 13 můžeme vidět nejběžnější metody routingu.

Pokud je zakázka malá a zdaleka nedosahuje kapacity vychystávání, je možné během jedné vychystávací jízdy vychystat více zakázek naráz. Tento postup je v znám jako dávkování objednávek nebo jednoduše dávkování (batching). Proces vychystávání zakázek je také možné rozdělit na zóny. Zboží patřící do stejné skupiny výrobků je uloženo ve stejné zóně. Ve srovnání s dávkováním nemá zónování významný vliv na výkonnost systému vychystávání. Výhoda zónování spočívá ve snížení přeplněnosti uliček, a když je zboží opravdu v jedné malé oblasti, sníží se i cestování. Hlavní nevýhodou je konsolidace objednávek, když ji kompletuje více vychystávačů z různých zón. [6]

Optimalizace se tedy používá z následujících důvodů, které však vychází z požadavků zákazníků. [6]

2.5.3 Využití simulace pro optimalizaci

Výrobci čelí stále rostoucí poptávce po výrobcích na míru a zároveň po výrobcích šetrných k životnímu prostředí. Tato situace má vliv na životní cykly výrobků i procesů s nimi spojenými. Pro co nejlepší rozhodování v oblasti těchto výrobních cyklů se sleduje výkonnost výrobních systémů pomocí sběru a následné analýzy velkého množství dat, především z úrovně výrobních hal. Relativně nový obor, Data Mining, pomáhá odhalit poznatky a vzorce skryté v těchto datech. Samotné poznatky jako takové však nemusí vždy vést k doporučením, která by bylo možné reálně využít. S rostoucím počtem možných provozních podmínek navíc roste i složitost a obtížnost pochopení a vyhodnocení těchto dopadů.

K testování a vyhodnocování dopadů různých rozhodnutí na výkonnost v různých provozních podmínkách se tedy čím dál častěji používají simulační modely. Uživatelé mohou se simulačním modelem provádět experimenty v rámci definovaných situací, aby pak lépe porozuměli chování daného systému pro vybraný soubor podmínek a scénářů. Vzhledem k dynamické povaze výrobních operací je většina simulačních modelů stochastických. Možnosti, respektive účely využití simulace v různých fázích životního cyklu výrobku a jeho výrobních procesů je možné vidět v tabulce č. 1.

V případě potřeby lze simulační modely aktualizovat pomocí dat: pro možné předpovědi dopadu nových zakázek, poruch zařízení a změn v provozu – např. změny rozvržení ve výrobní hale, směn a počtu zařízení či personálu. Simulace může být také použita pro generování a vyplnění mezer pro analýzu jinými metodami. [7]

Tabulka 1: Simulace v různých fázích životního cyklu výrobního a logistického systému; [7]

Fáze	Cíl	Typ modelu
Plánování	Objem výroby, požadavky podniku	Systémová dynamika, teorie řízení
Základní návrh	možnosti uspořádání layoutu, analýza propustnosti, souhrnná analýza	Systémová dynamika, diskrétní simulace, multiagentní modelování
Detailní návrh	Layout, specifikace vybavení, možnosti řízení výroby	Diskrétní simulace, agent-based modely, multiagentní modelování
Nastavení a optimalizace	Ověření výroby	Diskrétní simulace

Výhod modelování a simulace výrobních a logistických systémů je několik.

- Pomáhají identifikovat a kvantifikovat výrobní zařízení a personál.
- Mohou předpovídat rozdělení pravděpodobnosti spojené s výkonem
- Lze je využít k vyhodnocení provozních procedur
- Mohou zohlednit stochastické chování systému

3

Simulace podnikových procesů

Počítačová simulace se už přes 50 let široce využívá jako nástroj pro návrhy a analýzu podnikových systémů a procesů. Dá se aplikovat v oborech jako je např. výroba, obchod, logistika, stavebnictví. Počítačovou simulaci si tedy nyní blíže představíme a zaměříme se přitom na její historii, charakteristiku, přednosti a omezení, oblasti aplikace, způsoby zavádění a postup řešení simulační studie.

3.1 Historie simulací

Až do poloviny 80. let se dá vývoj počítačového simulačního softwaru popsat jako vývoj programovacích jazyků pro simulace – jedná se o první tři generace simulačních programů. Od té doby došlo k velice výrazné proměně. Změnila se složitost návrhů, vývoj a provedení simulace. Tyto změny vyplívají převážně z technologického pokroku, kdy dochází k radikálnímu zlepšování vlastností hardwaru i softwaru. Zároveň však rostly i požadavky koncových uživatelů. V dnešní době se využívají čím dál více SW s 3D animacemi,

množstvím různých knihoven a modulů pro optimalizace napříč obory. V následující tabulce je vývoj simulačního softwaru od první generace až po současnost. [8]

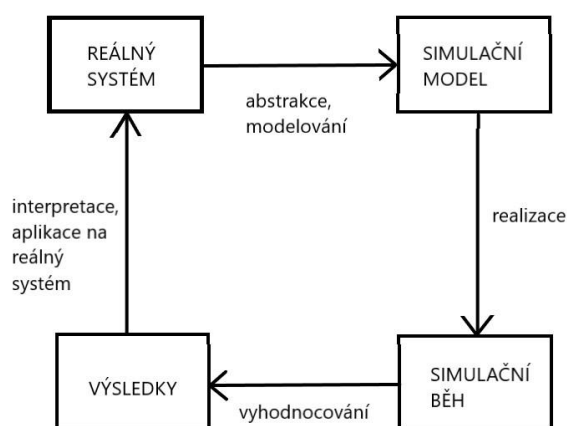
Tabulka 2: Historie simulačních programů [9]; vlastní zpracování

Gen.	Časové období	Charakteristika	Příklad SW
1.	50. léta	Obecné programovací jazyky	Fortran, ALGOL
2.	60. léta	Simulační jazyky	GPSS, SIMULA
3.	70. léta	Grafické zpracování výsledků, diskrétní, spojitá a kombinovaná simulace	GASP IV, ACSL
4.	1. polovina 80. let	Problémově orientované simulátory	SIMAN/CINEMA, SIMFACTORY
5.	2. polovina 80. let	Využití AI a expertních systémů	SIMKIT
6.	90. léta	Objektově orientované simulační systémy	ARENA, WITNESS
7.	Přelom tisíciletí	Využití 3D animace a virtuální reality	WITNESS, SIMIO

3.2 Simulace a modelování

Simulace je vyjádření nějakého systému nebo procesu za pomoci počítačového modelu. Tento model napodobuje vlastnosti a reakce skutečného systému na události a podněty odehrávající se v průběhu času. [10]

Simulace je tedy napodobení komplexního systému naším modelem a zkoumání jeho chování. Jednoduchý průběh simulace můžeme vidět na následujícím obrázku.



Obrázek 14: Princip počítačové simulace [9]; vlastní zpracování

Modelování znamená navrhování zjednodušeného obrazu komplexního systému, který má za cíl předpovězení (vykreslení) chodu onoho složitějšího systému se zaměřením na důležité

sledované prvky a faktory. Právě toto zjednodušené zobrazení se nazývá model. Je navrhnut tak, aby zachytil určité, pro nás důležité, aspekty chování našeho řešeného systému.

Tyto modely mohou mít několik podob, např.:

- a) Fyzický model – zjednodušený, popř. zmenšený model nějakého objektu (např. model dopravního prostředku)
- b) Analytický nebo matematický model – je soubor vztahů nebo rovnic mezi matematickými proměnnými
- c) Počítačový model – vyjádření systému pomocí programu

[11]

Pro naše účely a z důvodu přehlednosti výsledků se zabýváme modelem počítačovým.

Cílem je, aby námi vybrané sledované faktory byly zachyceny jako funkce času. Z tohoto úhlu pohledu rozlišujeme tři hlavní typy simulačních modelů – diskrétní, spojitý a kombinovaný.

U diskrétní simulace (Discrete event simulation) je systém vymodelován jako série událostí v časovém období. Události jsou vybrané body v čase, ve kterých dojde ke změně v našem systému. Nepředpokládá se, že mezi dvěma po sobě jdoucími událostmi, dojde ke změně. Diskrétní simulace běží také o dost rychleji proti ostatním typům, protože nemusí zpracovávat každý časový interval.

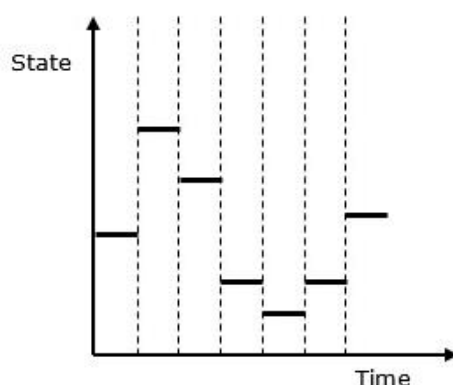
Naproti tomu u spojitých modelů (continuous), jak již název napovídá, pozorujeme systém nepřetržitě v každém časovém bodě. Systém je tedy popsán pro všechny body v čase.

Často bývají při tvorbě simulačního modelu relevantní oba úhly pohledu, tento způsob se nazývá kombinované spojitě diskrétní modelování. [12]

3.3 Diskrétní simulace (DES)

Modelování na bázi událostí je základní metoda prezentování informací a našich znalostí o diskretním systému, kde je dynamika tohoto systému zobrazena v grafu událostí. Tento graf je síťovým modelem, který reprezentuje logické a časové vztahy mezi událostmi. Je formálním modelem pro provedení simulace, který lze jednoduše implementovat pomocí metodiky příští události. [13]

Na následujícím obrázku vidíme graf změn stavu systému v průběhu času. Každá z těchto změn je nazývána událostí.



Obrázek 15: Chování diskretních simulací [14]

3.4 Přednosti a omezení simulací

S vývojem simulačních softwarů, jejich rozšířeností a rostoucí výkoností výpočetní techniky nezbývá moc důvodů, proč počítačovou simulaci nevyužít. Jedná se o podpůrný nástroj v oblasti projektového managementu a investování do nových výrobních systémů nebo zefektivnění těch stávajících. Několik argumentů pro využití simulace je shrnuto v následujících bodech. Samozřejmě i u simulací existují určitá omezení – viz kap. 3.4.2.

3.4.1 Přednosti

Mezi hlavní přednosti patří:

1. Simulací lze řešit jednoduché, ale i velmi složité systémy, které nelze řešit klasickými analytickými metodami, nebo naopak by ostatní metody daný problém příliš zjednodušily. Je to také jeden ze způsobů ověření výsledků jiných metod.

2. Možnost pozorování vývoje systému v reálném, zrychleném nebo zpomaleném čase. Nastavení parametru času je velkou výhodou v tom, že v řádu minut dostaneme průběh např. výroby několika dnů, týdnů či měsíců. Díky tomu získáme rychle představu o eventuálním chování daného systému. Získáme tím výstupy pro různé varianty řešení i krizové situace, ze kterých pak můžeme vycházet při výběru vhodné alternativy.
3. Lze získat přehled zkoumaných procesů. Pomocí grafických prvků, animací a zvýraznění lze sledovat jednotlivé elementy a zaměřit se na ty, které nás zajímají. Lze pozorovat aktuální stav i průběh simulace v rámci delšího období (viz bod 2).
4. Komplexní pohled na danou problematiku a její vícekritériální analýza. Na modelu lze současně sledovat jak jednotlivé parametry, tak i zároveň samostatné subsystémy a jejich propojení. Hledá se právě optimální řešení celého systému (globální optimum). Při způsobu hledání lokální optimalizace se totiž zvyšuje riziko při následujícím propojení a jednotlivé subsystémy spolu budou velice těžko spolupracovat.
5. Díky simulaci lze přesně prověřit návrhy řešení. Jde vlastně o rychlé ověření navrhovaných variant – různé možnosti uspořádání výroby, počty pracovníků na stanovištích, strategie. Zároveň si můžeme připravit alternativy pro neočekávané situace – porucha stroje, nepřítomnost pracovníka.
6. Vytvoření daného simulačního modelu je poměrně složité, je nutná důkladná znalost systému reálného. Již při přípravě modelu (vstupní analýza, sběr dat) můžeme přijít na nedostatky a rezervy.
7. Pozorování modelu má za následek hlubší porozumění systému reálnému. Můžeme např. změnou jednoho či více parametrů pozorovat vliv na celkový systém.
8. Týmová práce. Složitost situace vyžaduje spolupráci odborníků z mnoha oblastí. Právě kvůli oné komplexnosti by jeden pracovník na řešení problému nestačil. Je

proto vhodné vytvořit tým lidí z více oblastí, často i na první pohled problematice vzdálené. Tento tým samozřejmě nemusí být stálý.

9. Možnost použití vytvořeného modelu i v jiných činnostech podniku. Již vytvořený simulační model můžeme využít např. k zavádění a zlepšování výroby, popisu nových strategií a změn, zaškolování nových zaměstnanců.

10. A další.

[9]

3.4.2 Omezení

Mezi hlavní omezení patří:

1. Kvalita simulačního modelu je závislá na vstupních datech. Bez správných dat nelze vytvořit efektivní simulaci.

2. Simulace není optimalizací. Simulace je pouze nástrojem, který nám s optimalizací pomáhá.

3. K vytvoření simulačních modelů je nutné speciální školení. Kvůli nejednoznačným a špatně postaveným modelům často dojde k nepovedenému pokusu o získání nějakých poznatků ze simulace.

4. Vzhledem k tomu, že je se simulací spojeno velké množství náhodných jevů (vstupů), může být těžké rozlišit, zda je pozorování výsledkem vzájemných vztahů v systému a jeho fungování, nebo je výsledkem náhodnosti.

5. Vymodelování a analýza simulace může být velice obtížná časově i finančně. [15]

6. DES poskytuje pouze odhad výsledků modelu – stejný problém jako u modelů matematických a tradiční analýzy. [16]

7. DES lze efektivně použít pouze tehdy, dokáže-li simulační model v dostatečné úrovni napodobit reálnou situaci. [17]

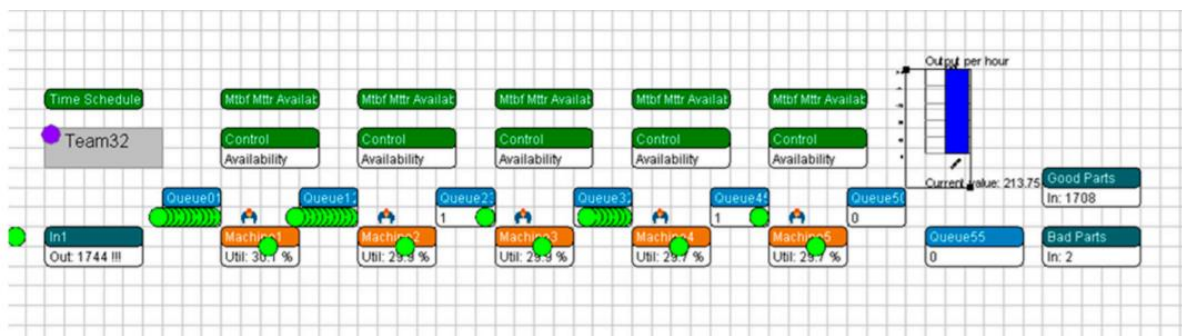
8. A další.

3.5 Oblasti využití simulací

Simulace se v dnešní době využívá v mnoha oblastech a odvětvích průmyslu, od výroby až po management.

Výroba

Studování faktorů, které mají vliv na proces výroby, např. nedostatek pracovní síly, poruchy strojů, délka výroby, rozložení strojů a tok materiálu [15]. Dalším příkladem je využití simulace jako základ ABC (activity-based costing), které se pak použije pro optimalizaci výrobních procesů a redesignu ve výrobě [19]. Uplatnění také najde při optimalizaci výrobních linek. Lze otestovat změnu logiky řízení výroby, přidání paralelních strojů a vyladit jednotlivé části výrobního systému při sériové a velkosériové výrobě tak, aby nedocházelo k nežádoucím frontám mezi stroji či montážemi. Model výrobní linky je na následujícím obrázku. [20]



Obrázek 16: Model výrobní linky [18]

Logistika

Simulačních modelů pro optimalizaci vnitropodnikových logistických systémů se využívá za účelem zvýšení produkce, snížení prostojů, snížení počtu pracovníků a celkově hledání nejlepšího možného nastavení výrobních a procesních systémů.

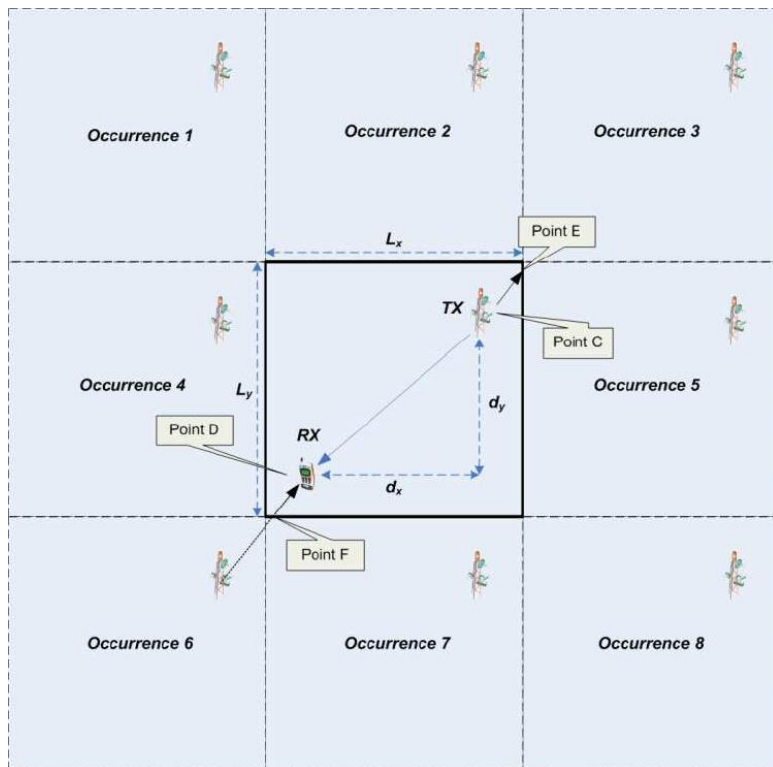
Vytvářejí se proto simulační modely systémů přesunu zásob ze skladu, analýzy toku materiálu, analýzy vnitropodnikového dopravního systému, řízení zásob podniku, analýzy manipulačních jednotek v podniku. Řešení logistiky skladu pomocí SW FlexSim je na následujícím obrázku. [21]



Obrázek 17: Řešení logistiky skladu [22]

Komunikační technologie

Simulace počítačových sítí, kde se před zavedením simulují nové protokoly pro různé scénáře zatížení sítě. Využívá se pro modely call center [15]. Dalším využitím je simulace komunikačních sítí (WiFi, Bluetooth, GSM, UMTS), kde se zkoumá pokrytí plochy danou sítí nebo vysílání signálu antén, mobilní funkce, mobilita a přepojování mezi sousedícími anténami. Příklad simulace mobilní sítě je na následujícím obrázku. [23]

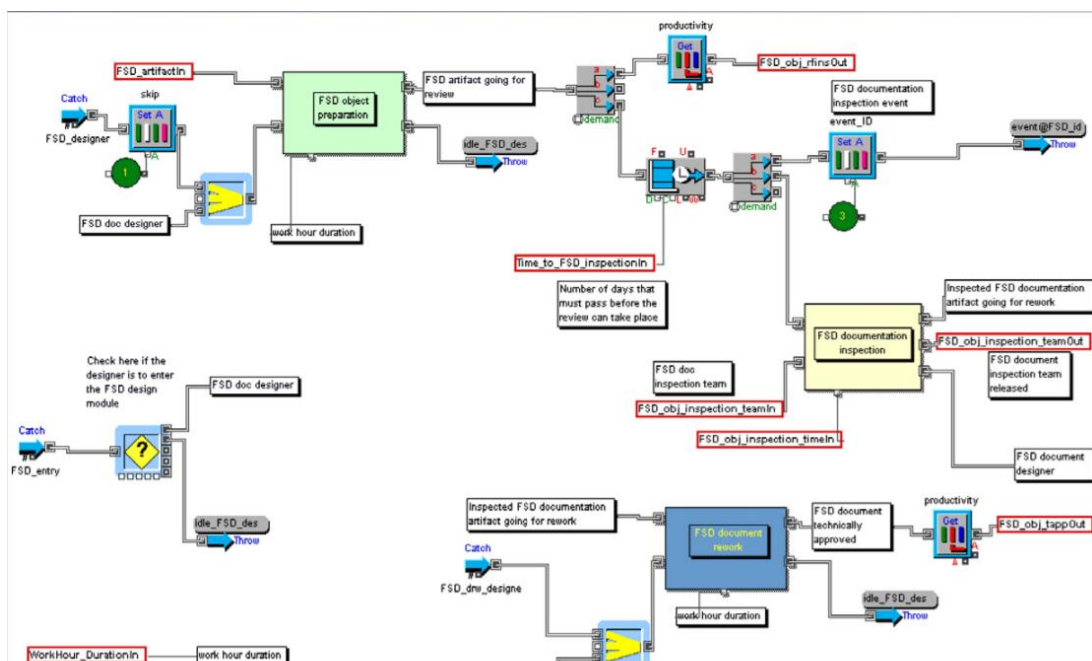


Obrázek 18: Pokrytí oblasti mobilní sítě [23]

Management

Simulace obchodních procesů (agent-based modeling), simulace výkonů obchodu a cen, simulování rizik únavy v nepřetržitých provozech. Využívá se taky při zavádění návrhů, kde pomáhá překonat problémy s implementací – chyby kódování, problémy s opotřebením.

Příklad řízení SW projektu je na následujícím obrázku. [15]



Obrázek 19: Řízení software projektu [25]

Zdravotnictví

Modelování postupu pacientů, čekáren, kapacity pohotovosti, délky pobytu pacientů.[15] Pro výběr vhodné léčby se v některých případech může využít modelování postupu nemoci jako dalšího zdroje informací. Mnoha využití najde simulace při řízení provozu nemocničního prostředí, krom výše zmíněných to jsou např. plánování rozvrhu zaměstnanců, přidělování zdrojů, vyhodnocování efektu provozních změn. Simulace v modulu FlexSim: Healthcare je na následujícím obrázku. [26]



Obrázek 20: Simulace pacientů na pohotovosti [23]

3.6 Zavádění simulačních nástrojů v podniku

Simulaci lze v podnicích zavádět několika způsoby. V základu se jedná o zavádění ve vlastním podání, zakoupení simulačního projektu nebo kombinaci těchto dvou způsobů. Jednotlivé způsoby jsou popsány dále.

a) Ve vlastním podání

Simulační projekt si zpracuje podnik sám. Zakoupí simulační software, navrhne a zpracuje model vlastními zaměstnanci. Tento způsob se využívá v podnicích, kde se simulací řeší více projektů po celý rok. Z tohoto důvodu se vyplatí spíše pro větší firmy, kde se vlastní simulací ušetří značné náklady (např. časté nakupování již hotových projektů). Nevýhodou jsou vysoké náklady spojené s vlastní simulací – nákup softwaru, školení pracovníků, licence a vyčlenění minimálně jednoho pracovníka přímo na simulace. Na druhou stranu však ve firmě zůstane know-how, takže se mohou modely opětovně využít a upravit pro jiné vnitropodnikové činnosti.

b) Zakoupením hotového simulačního projektu

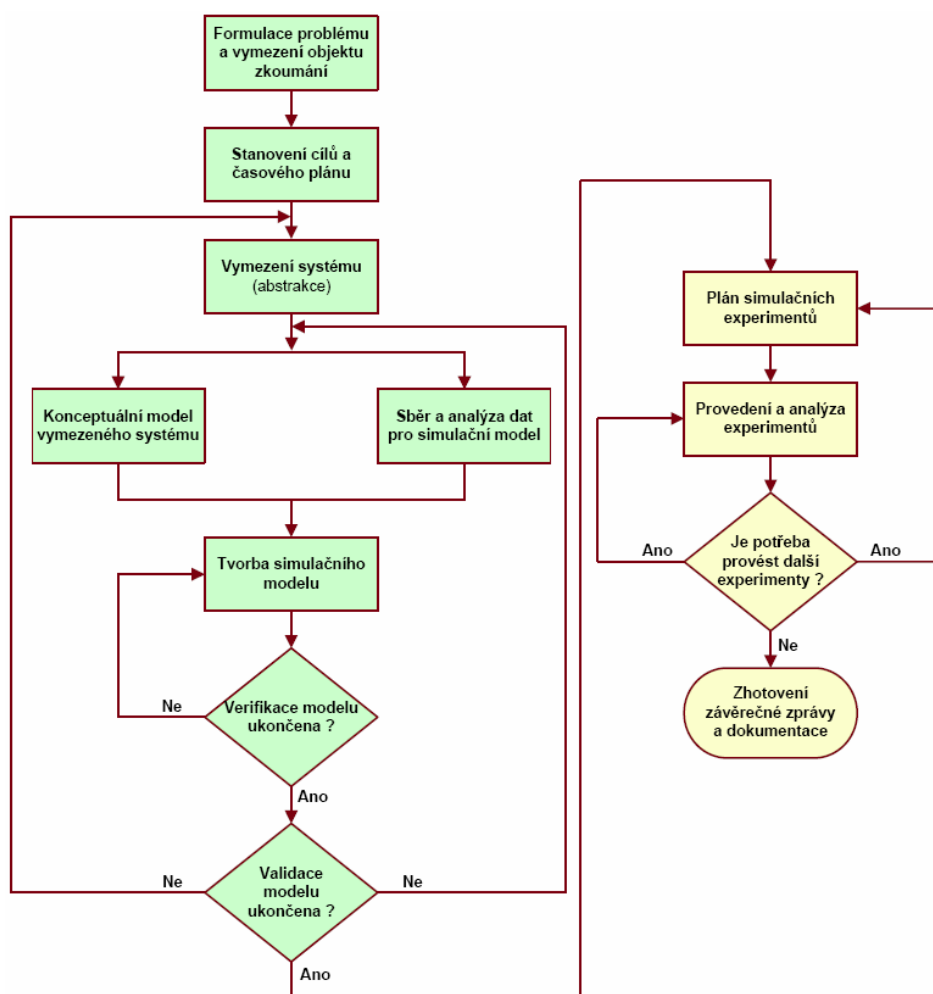
Tento způsob se vyplácí spíše tam, kde podnik simulací řeší malé množství projektů (cca 1-2 za rok), vlastní simulace by byla příliš časově náročná, nebo by ji mohla ohrozit nesprávná formulace požadovaného modelu. Externí podnik může přinést levnější řešení (neplatí pro vlastní simulaci při nižších investicích) a nákup celého projektu naráz, avšak logika řešeného problému zůstává u cizí firmy.

c) Kombinací výše zmíněných

V podstatě se jedná o řešení simulačního projektu podniku v úzké spolupráci s externí konzultační firmou, která se zabývá simulacemi. Na první pohled se tato varianta zdá jako nevýhodná, protože může být řešení ve výsledku dražší než u zakoupení již hotového projektu, ale má i několik výhod – externí simulační know-how, možnost souběžné práce a tím zkrácení času projektu, logika řešení zůstane u zadavatele, využití vlastních pracovníků na projektu. [9]

3.7 Tvorba simulačního projektu

Tvorba simulace a modelu je složitá činnost, která sleduje několik zásadních kroků. Schéma tvorby simulačního projektu je znázorněno na obrázku č. 21. Blíže jsou popsány ty nejdůležitější kroky.



Obrázek 21: Životní cyklus simulačního projektu [27]

Analýza problému a sběr informací

Při tvorbě simulačního modelu musíme nejdříve analyzovat danou problematiku, kterou má simulace řešit. Model systému se nevytváří bezdůvodně, ale většinou za účelem

vyřešení nějakého systémově orientovaného problému. Tedy prvním krokem při tvorbě modelu je pozorování skutečného systému a interakcí mezi jeho jednotlivými částmi a shromažďování údajů o jejich chování. Samotné pozorování však málokdy vede k dostatečnému pochopení chování systému. Lidé obeznámení se systémem nebo jakýmkoli podsystémem, by měli být zapojeni, aby se využilo jejich znalostí. Obsluha, technici, opraváři a údržbáři, inženýři, vedoucí pracovníci a manažeři rozumí určitým aspektům fungování systému nejlépe. V průběhu vývoje modelu mohou vyvstat nové otázky. [36]

Sběr dat

Sbírání relevantních dat je nutné pro určení vstupních parametrů modelu. Analytici mohou formulovat předpokládané rozdělení náhodných proměnných. Při nedostatku vhodných dat je možné určit přinejmenším rozsahy parametrů modelu a provést simulaci pro všechny, nebo některé z těchto parametrů. Sběr informací je také důležitý pro ověření modelu – srovnávají se výsledky s předpověďmi. [28]

Tvorba koncepčního modelu

Sestavení koncepčního modelu je soubor předpokladů o složkách a struktuře systému a hypotéz o hodnotách vstupních parametrů modelu. Následuje koncepční validace, což je porovnání skutečného systému s koncepčním modelem. I koncepční model bývá verifikován a validován. [36]

Tvorba modelu

Po zmapování a nastudování problematiky a získání potřebných dat následuje tvorba modelu. Analytik, nebo jiná osoba pověřená tvorbou modelu, vytváří model na základě získaných dat a implementuje jej jako počítačový program. K tomu se využívá buď programovacích jazyků, nebo v dnešní době častěji simulačních programů. [28]

Ověření modelu (verifikace a validace)

Krok ověření má za cíl zkontrolovat, zda byl model správně zkonstruován – zda odpovídá daným specifikacím a dělá to, co má. Dále se ověřuje vhodnost modelu – přirovnává se k empirickým datům (pokud existují). Jakékoliv větší nesrovnalosti znamenají nevhodnost

navrhovaného modelu pro náš projekt a je nutné provést úpravy. V praxi je běžné několikrát opakovat kroky tvorby, ověření a modifikace modelu. [28]

K verifikaci a validaci lze přistoupit několika způsoby. Tyto způsoby jsou následující:

- Nechat provozní model zkontrolovat někým jiným než jeho tvůrcem, nejlépe odborníkem na použitý simulační software.
- Vytvořit vývojový diagram, který zahrnuje každou logicky možnou akci, kterou může systém provést.
- Pečlivě prozkoumat výstup modelu z hlediska jeho logiky při různých nastaveních vstupních parametrů. Zobrazit z implementovaného modelu výstupních statistik a ty pečlivě prozkoumat.
- Nechat operační model na konci simulace vypsat vstupní parametry, abychom se ujistili, že hodnoty těchto parametrů nebyly neúmyslně změněny.
- Pokud je operační model animovaný, ověřit, zda to, co je vidět v animaci, napodobuje model skutečného systému. Například pohyb AGV či VZV. [36]

Návrh a provedení simulačních experimentů

Po ověření modelu následuje návrh souboru experimentů. Experimenty se provádí za účelem odhadu chování modelu a podpory námi řešeného problému. Analytik zvolí několik různých situací a simulaci spustí. To nám pomůže nahlédnout do toho, jak námi navrhnutý systém funguje. Simulace se spouští vícekrát, abychom získali dostatečné množství informací pro statistickou spolehlivost. [28]

Analýza výsledků

Výsledky experimentů jsou podrobeny logické a statistické analýze podle námi zvoleného způsobu hodnocení chování modelu (z kroku 1). Jde o zvolení nejvhodnějšího návrhu pro naši problematiku. [28]

Závěrečná doporučení

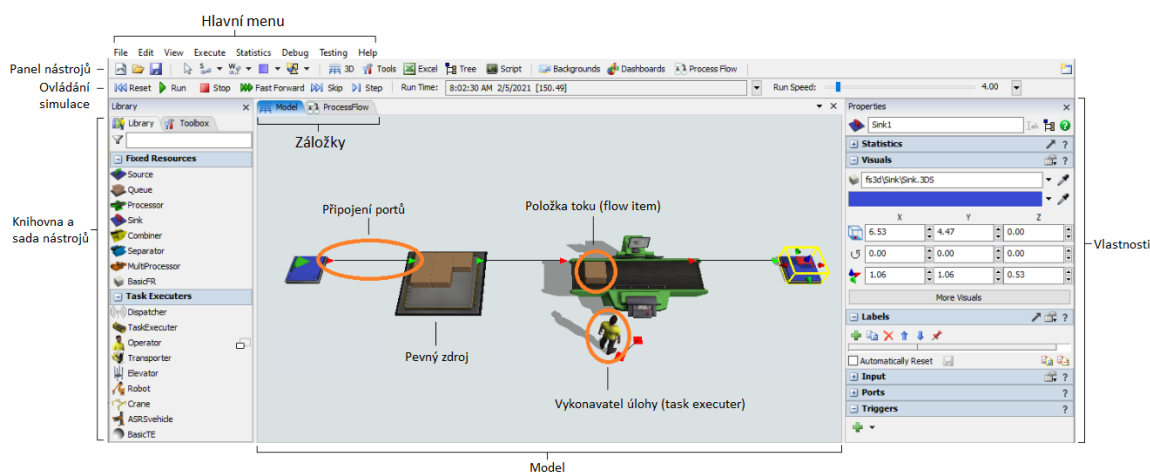
Na závěr analytik provede (na základě analyzovaných výsledků) závěrečné doporučení, či řešení naší problematiky. [28]

Simulační nástroj FlexSim

4 FlexSim je 3D simulační software, který umožňuje modelovat, simulovat, předpovídat a vizualizovat mnohé podnikové systémy v různých odvětvích: výroba, manipulace s materiálem, zdravotnictví, skladování, těžba, logistika a další. Tato kapitola slouží k vysvětlení použitých nástrojů, objektů a tohoto softwaru pro potřeby praktické části.

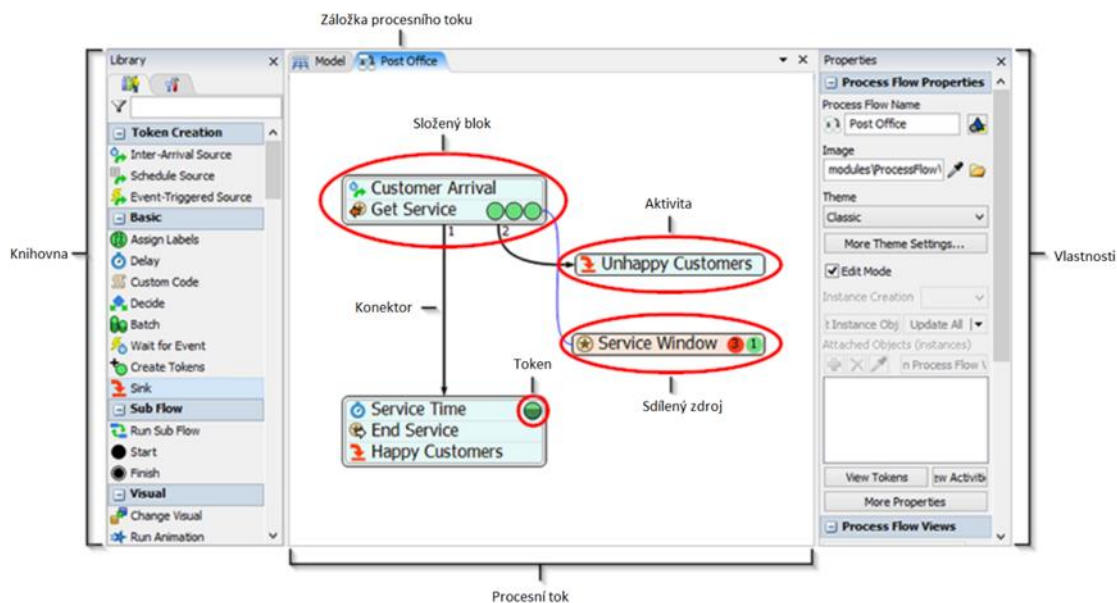
4.1 Přehled uživatelského rozhraní

Simulační model vytváříme především pomocí dvou rozhraní: 3D modelu a nástroje Process Flow (dále procesní tok). 3D model je místo, kde probíhá vizualizace vlastního podnikového systému pomocí 3D grafiky. Procesní tok je naopak místem, kde tvoříme logiku, která ovládá 3D model. Nejdůležitější prvky rozhraní 3D modelu jsou označeny na následujícím obrázku.



Obrázek 22: Uživatelské rozhraní 3D modelu [29]

Rozhraní procesního toku se velmi podobá rozhraní 3D modelu. Nejdůležitější prvky nástroje procesního toku jsou označeny na následujícím obrázku.



Obrázek 23: Uživatelské rozhraní procesního toku [29]

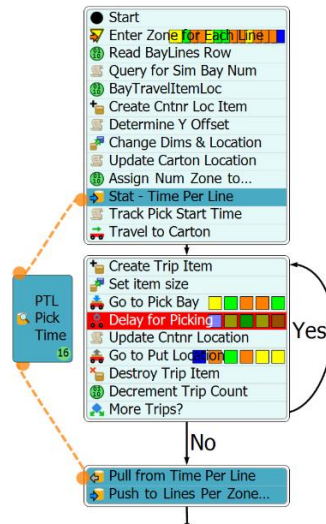
4.2 Základní objekty 3D modelu

Objekty jsou základními stavebními kameny 3D modelu. Různé typy objektů mají v rámci simulačního modelu různé účely a funkce. Mezi nejčastěji používané objekty patří (viz obrázek č. 22):

- Položky toku (flow items) - objekty, které se pohybují (neboli "proudí") simulačním modelem, obvykle z jedné stanice (obvykle pevného zdroje) do jiné navazující stanice. Tyto položky mohou představovat výrobky, zákazníky, papíry, díly nebo jakékoli jiné položky, které se pohybují do různých stanic v podnikovém systému.
- Pevné zdroje (fixed resources) - objekty, které zůstávají ve 3D modelu nehybné. Každý pevný prostředek plní určitou funkci. Zdroj například vytváří položky toku v určitých intervalech a zavádí je do modelu. Fronta uchovává položky toku, dokud nejsou potřeba dále v toku. Sink odebírá položky toku z modelu atd.
- Vykonavatelé úloh (task executers) - objekty, které se mohou pohybovat v 3D modelu a provádět úlohy, jako je přeprava a manipulace položek toku, obsluha strojů atd. Nejběžnějším typem vykonavatele úloh je operátor, který může v simulačním modelu představovat zaměstnance. [29]

4.3 Procesní tok

Procesní tok je složen z jednotlivých aktivit a jeho hlavním účelem je řídit logiku simulačního modelu. Ve FlexSimu můžeme pracovat se třemi typy procesního toku. Příklad procesního toku pro řešení logiky skladu můžeme vidět na následujícím obrázku.



Obrázek 24: Logika průběhu procesů skladu [22]

- Obecný – tento typ procesního toku je ideální pro vytváření logiky procesních toků, která bude muset být globálně přístupná mnoha objektům v simulačním modelu. Obecný procesní tok je vhodným místem pro umístění jakýchkoli atributů (štítků), proměnných nebo logiky, kterou musí používat více objektů v modelu.
- Objektový – slouží pro vytvoření logiky pro vlastní objekty pevných zdrojů nebo pokud je vyžadováno, aby měl objekt další funkce nad rámec standardní logiky. Při definování tohoto toku objektu definujeme logiku z pohledu samotného objektu. Tento způsob je obecně synonymem objektově orientovaného programování a také se shoduje s konceptem agent-based modelování.
- Dílčí tok (sub-flow) – dílčí tok je samostatný procesní tok, který začne běžet, když je spuštěn jinou činností nebo událostí v jiném procesním toku. Dílčí toky jsou části samostatné logiky, které se spustí, když dostanou podnět určitými událostmi v simulačním modelu nebo obecném toku procesu.

Dále jsou představeny klíčové prvky prostředí procesního toku, což je token, aktivita, sdílený zdroj, konektor a složený blok. [29]

4.3.1 Token

Nejzákladnější složkou procesního toku je token. Tokeny se v toku procesu zobrazí pouze tehdy, když je spuštěn simulační model. Token je na následujícím obrázku.



Obrázek 25: Token [29]

Tokeny jsou objekty, které během simulačního běhu procházejí aktivitami v procesním toku. Fungují velmi podobně jako položky toku ve standardním 3D modelu. Stejně jako položky toku se tokeny pohybují z jedné aktivity do druhé, podobně jako se položka toku pohybuje od zdroje k frontě, od procesoru atd. Na rozdíl od položek toku však tokeny nemusí nutně představovat fyzický objekt, který se pohybuje systémem. Tokeny mohou být abstraktnější - to znamená, že mohou představovat cokoli, co chceme, aby představovaly. Mohou například představovat objednávku zákazníka, která je zadávána, hovor do call centra, logické seskupení palet, které jsou nakonec naloženy na nákladní automobil za účelem doručení atd. Tokeny budou často logicky propojeny s fyzickými objekty ve standardním systému (a s položkami toku ve standardním 3D modelu), ale nemusí tomu tak být. Je v podstatě na nás, abychom definovali povahu vazeb mezi tokeny a fyzickými prvky v simulačním modelu.

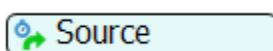
Na nejzákladnější úrovni je token pouze svazek dat, který se pohybuje v procesním toku. Každý token může nést následující základní informace:

- ID - když je token vytvořen pomocí jedné z aktivit vytváření tokenů (jedné ze zdrojových aktivit), je mu automaticky přiřazeno jedinečné identifikační číslo, které lze použít jako referenci. Číslo ID je jedinečné mezi všemi objekty a instancemi procesního toku v rámci celého modelu.
- Název - pojmenování tokenů nám pomáhá lépe identifikovat jejich účel, funkci nebo umístění v toku procesu. Tokeny nemusí mít jméno, ale lze jim ho přidělit, když jsou vytvořeny některou ze zdrojových aktivit toku procesu. Název lze případně změnit i v pozdější fázi procesního toku pomocí aktivity Vlastní kód (Custom code) s výběrovým seznamem Setname nebo příkazem setname().

- Štítky (label) - štítky jsou nezbytné pro vytvoření komplexního a dynamického procesního toku. Štítky uchovávají vlastní informace o tokenu, které lze použít k ovlivnění toho, co se s daným tokenem stane při jeho postupu procesním tokem. Různé aktivity mohou přiřazovat údaje ke štítkům tokenu při jeho pohybu aktivitami ve vývojovém diagramu. [29]

4.3.2 Aktivity

Aktivity jsou logické operace nebo kroky v procesním toku, jsou základními stavebními kameny každého procesního toku. Příklad aktivity je na následujícím obrázku.



Obrázek 26: Aktivita zdroj [29]

Aktivity se přetahují z knihovny do procesního toku a propojují se pomocí konektorů. Jakmile token vstoupí do nějaké aktivity, provede logiku spojenou s daným typem této aktivity. To může zahrnovat přiřazení štítků, přesun položek toku, zpoždění tokenu atd.

V některých aktivitách, jako je například přiřazení štítků, token zahájí a ukončí aktivitu, aniž by v simulačních hodinách uběhl čas. Během běhu modelu nebude token v dané aktivitě viditelný. Pomocí tlačítka krok na ovládacím panelu simulace však můžete vidět, jak se token pohybuje v každé aktivitě po jednom kroku. V jiných aktivitách, jako je získání zdroje, může být token odložen a čekat v dané aktivitě, dokud nebudou splněna určitá kritéria.

4.3.3 Sdílené zdroje

Sdílené aktivum je omezený zdroj, který mohou tokeny nárokovat nebo uvolnit v určitých bodech toku procesu. Přestože jsou velmi podobné aktivitám, existuje několik drobných rozdílů. Ve výchozím nastavení mají jinou barvu než ostatní aktivity. Příklad sdílených zdrojů je na následujícím obrázku.



Obrázek 27: Sdílené zdroje [29]

Sdílené zdroje mohou na tokeny uvalit omezení tím, že token čeká, pokud požadovaný zdroj není k dispozici. Reálným příkladem může být určitý nástroj, který je sdílen třemi různými

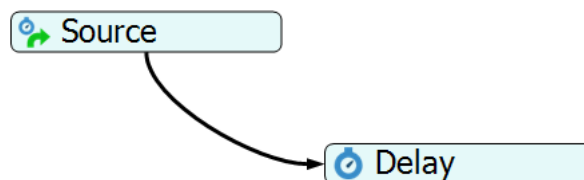
pracovními stanicemi. Pokud jedna stanice potřebuje nástroj, zatímco je již nárokován jinou stanicí, musí tato stanice čekat, dokud nebude nástroj k dispozici. Stejně tak, pokud token potřebuje sdílený zdroj, aby mohl přejít k další aktivitě, bude čekat u své aktuální aktivity, dokud nebude tento zdroj k dispozici.

Existují tři typy sdílených zdrojů:

- Seznam - reprezentuje seznam tokenů, položek toku, vykonavatelů úloh, sekvencí úloh, čísel, řetězců atd. Procesní toky mohou používat seznam, který je lokální pro samotný procesní tok nebo může být vázán na globální seznam v simulačním modelu.
- Zdroj - představuje omezenou zásobu nějakého zdroje, který lze získat a uvolnit. Lze jej použít k simulaci zásoby zboží, služeb, času, materiálu, zaměstnanců atd.
- Zóna - může shromažďovat statistické informace, které nejsou k dispozici pro standardní aktivitu. Může také omezit přístup k části toku procesu na základě určitých statistik nebo jiných kritérií.

4.3.4 Konektory

K propojování jednotlivých aktivit se používají konektory. Konektor je spojení mezi dvěma aktivitami. Příklad je na následujícím obrázku.

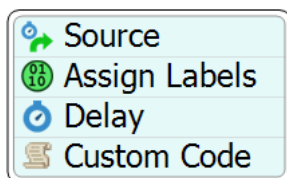


Obrázek 28: Konektor [29]

Během simulačního běhu budou tokeny používat konektory pro přechod z jedné aktivity do další navazující. Některé aktivity umožňují více než jeden odchozí konektor.

4.3.5 Složené bloky

Složený blok nebo jen samotný blok slouží pro popis sady aktivit, které byly spojeny do jednoho pohyblivého bloku. Příklad složeného bloku je na následujícím obrázku.



Obrázek 29: Složený blok [29]

Složené bloky umožňují snadno propojit sadu aktivit do jediné sekvence kroků, takže je lze snadno přesouvat a upravovat, jako by šlo o jeden blok. Když jsou aktivity v jednom bloku, jsou automaticky propojeny, jako by každá aktivita byla spojena konektory.

Když token vstoupí do tohoto bloku, začne vždy u nejvyšší aktivity a postupně se propracuje až k té nejnižší. Jakákoliv spojení do bloku budou připojena k té nejvrchnější bez ohledu na to, kam spojení v naskládaném bloku směřuje. Jakákoliv odchozí spojení z bloku budou připojena od nejspodnější aktivity. [29]

Úvod do praktické části

5

Praktická část se zabývá reálným problémem výrobního podniku. Postup je dle kroků simulační studie v kapitole 3.7. Na začátku je krátce představena společnost, následuje charakteristika řešeného problému, dále koncepční model a následuje tvorba simulačního modelu. Tvorba samotného simulačního modelu je rozdělena na 2 fáze. V první fázi se přenáší koncepční model do simulačního softwaru. Je zde využito pouze několik řádků dat a objektů, jelikož cílem bylo testovat logiku procesního toku. V druhé fázi je tento simulační model rozšířen o zpětný proud, kterým putují prázdné palety zpět do prvního stupně výroby. Zároveň je využito kompletního souboru dat pro 24 hodin provozu (3 směny). V poslední části je tento simulační běh vyhodnocen. Na základě výstupních dat je zformulováno doporučení.

5.1 O společnosti

V praktické části je řešen reálný případ s ostrými daty, proto si společnost nepřeje, aby bylo uvedeno její jméno. Z toho důvodu bude popsána anonymně a čísla budou zaokrouhlena nebo uvedena pouze řádově.

Jedná se o mezinárodní společnost, která je součástí dodavatelského řetězce známých automobilových značek. Česká pobočka vyrábí a dodává komponenty a panely pro osobní automobily. V této pobočce je zaměstnáno přes 2 000 zaměstnanců a plocha přesahuje

30 000 m². Ročně je zde vyrobeno několik set tisíc komponentů. Tržby se se pohybují do 100 mil. Kč, a vzhledem k typu průmyslu a velikosti společnosti aktiva přesahují částku 3 mld. Kč.

5.2 Charakteristika řešeného problému

Společnost úspěšně funguje řadu let, ovšem stává se, že ve skladu dojde k přetížení kapacity regálů nebo chybí prázdné přepravní vozíky na vyrobené díly. Konkrétně se jedná o následující problém:

Praktická část spočívá v návrhu simulačního modelu na přiřazení přepravních vozíků (transport trolley) do skluzových regálů pro 2 stupně výroby. Pozice ve skladových regálech pro přepravní vozíky jsou rozděleny do tzv. zón, které se mohou překrývat. Aktuální nastavení alokace do zón je fixní. V této souvislosti je nutné vytvořit simulační model, který na základě vstupních výrobních dávek v čase vyhodnotí a ukáže, které zóny budou přetížené.

V popisu modelu se často vyskytují pojmy jako přepravní vozík a paleta. Tyto pojmy jsou pro praktickou část zaměnitelné. Paleta je zvolena jako objekt 3D modelu, který představuje přepravní vozík.

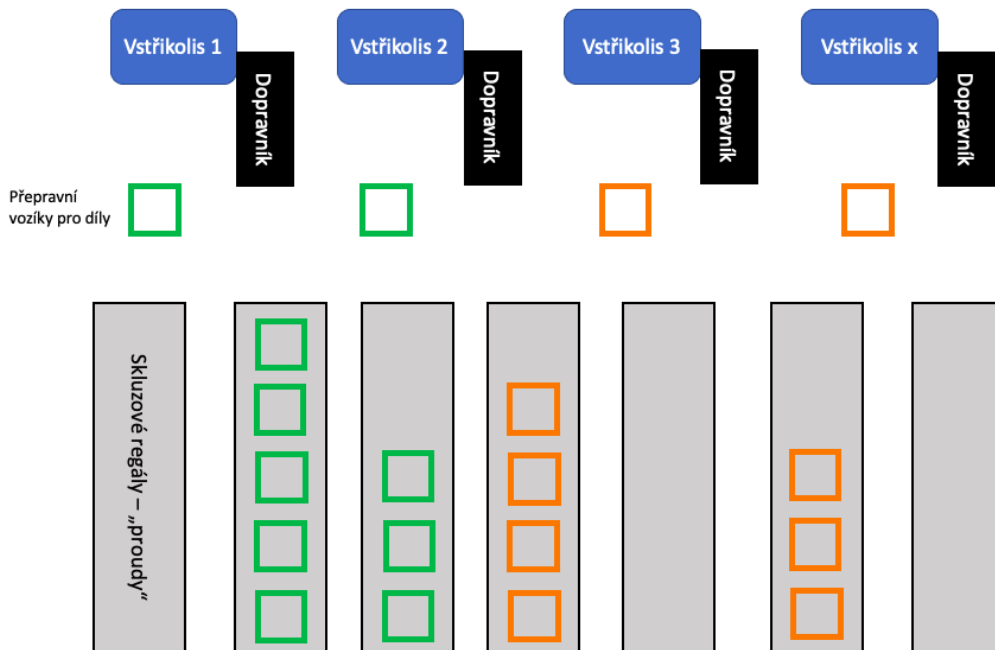
Dalšími pojmy jsou vstupní a výstupní signál. Jedná se o řádek v datové tabulce, který společně s časovým údajem udává informaci o změně množství kusů. Vstupní signál vychází ze vstříkolisů směrem ke gravitačním regálům. Výstupní signál vychází z gravitačních regálů do kaširovacích lisů.

6

Koncepční model a sběr dat

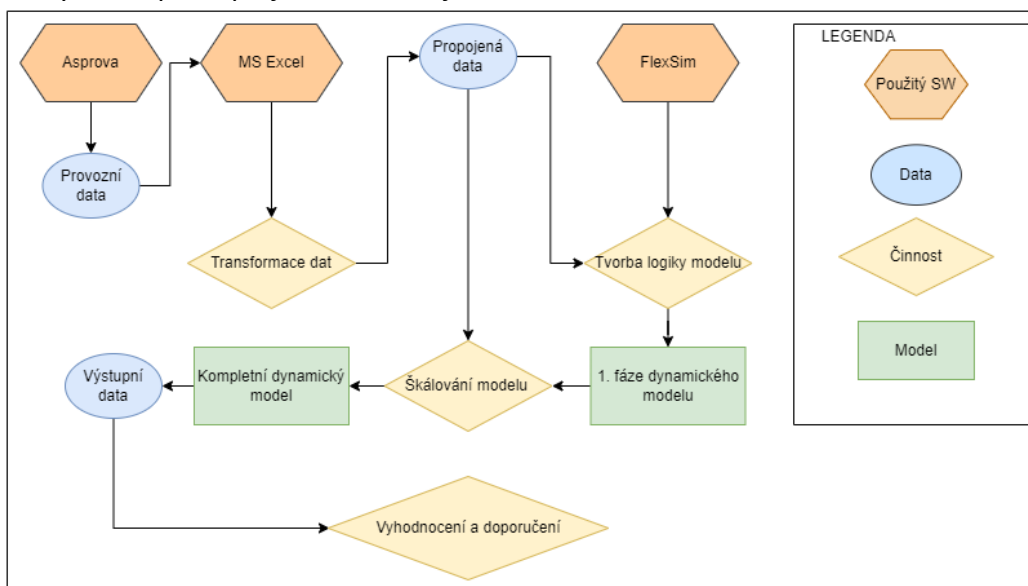
Počátkem simulace je výroba dílů na vstříkolisech, které se přes dopravník dostanou k operátorovi výroby. Operátoři poté postupně plní přepravní vozík (transport trolley) díly a v momentě jeho naplnění je připraven k naskladnění. Vyrábí se zde několik typů dílů, proto se tato doba plnění liší. Následuje naskladnění do gravitačních regálů (tzv. proudů), k tomuto účelu se využívá vysokozdvizný vozík. Logika tohoto procesu platí dle FIFO (first in, first out). Zároveň nesmí dojít k naskladnění různých typů dílů do stejného regálu. V momentě, kdy se naskladní první přepravní vozík do regálu, nesmí do něj vstoupit jiný typ

výrobku. Na druhé straně gravitačních regálů probíhá vyskladňování dle požadavků výroby v další části haly, u kaširovacích lisů. Schéma toku výroby je na následujícím obrázku.



Obrázek 30: Zjednodušené schéma toku výroby; vlastní zpracování

K tvorbě modelu byla využita reálná data, která získáme z implementovaných systémů pro plánování výroby (Asprova). Tato data jsou získána přímo z databáze, ale zároveň bude nutná i jejich úprava s využitím MS Excel ve smyslu propojení tabulek a transformace např. časových údajů pro to, aby s nimi FlexSim správně pracoval. Nástrojů bylo využito několik, schéma využití a postupu je na následujícím obrázku.



Obrázek 31: Nástroje a jejich využití; vlastní zpracování

Transformované (propojené) datové tabulky jsou na následujícím obrázku. Součástí každé tabulky vždy různá ID např. operace, výrobku nebo vozíku a dále množství, maximální kapacity a časy.

Tabulka: vstupní signál	Tabulka: výstupní signál	Tabulka: počáteční naskladnění																					
<table border="1"> <tr><td>ID_Operace</td></tr> <tr><td>ID_Výrobek</td></tr> <tr><td>Množství výrobku</td></tr> <tr><td>Čas počátku operace</td></tr> <tr><td>Čas konce operace</td></tr> <tr><td>ID_Vstřikolis</td></tr> <tr><td>ID_Paleta</td></tr> </table>	ID_Operace	ID_Výrobek	Množství výrobku	Čas počátku operace	Čas konce operace	ID_Vstřikolis	ID_Paleta	<table border="1"> <tr><td>ID_Operace</td></tr> <tr><td>ID_Výrobek</td></tr> <tr><td>Odebrané množství výrobku</td></tr> <tr><td>Čas počátku operace</td></tr> <tr><td>Čas konce operace</td></tr> <tr><td>ID_Vstřikolis</td></tr> <tr><td>ID_Paleta</td></tr> </table>	ID_Operace	ID_Výrobek	Odebrané množství výrobku	Čas počátku operace	Čas konce operace	ID_Vstřikolis	ID_Paleta	<table border="1"> <tr><td>ID_Regál</td></tr> <tr><td>ID_Zóna</td></tr> <tr><td>Maximální kapacita regálu</td></tr> <tr><td>Obsazená kapacita regálu</td></tr> <tr><td>ID_Paleta</td></tr> <tr><td>Množství výrobků na paletě</td></tr> <tr><td>ID_výrobek</td></tr> </table>	ID_Regál	ID_Zóna	Maximální kapacita regálu	Obsazená kapacita regálu	ID_Paleta	Množství výrobků na paletě	ID_výrobek
ID_Operace																							
ID_Výrobek																							
Množství výrobku																							
Čas počátku operace																							
Čas konce operace																							
ID_Vstřikolis																							
ID_Paleta																							
ID_Operace																							
ID_Výrobek																							
Odebrané množství výrobku																							
Čas počátku operace																							
Čas konce operace																							
ID_Vstřikolis																							
ID_Paleta																							
ID_Regál																							
ID_Zóna																							
Maximální kapacita regálu																							
Obsazená kapacita regálu																							
ID_Paleta																							
Množství výrobků na paletě																							
ID_výrobek																							

Obrázek 32: Datové tabulky; vlastní zpracování

Tvorba simulačního modelu je rozdělena do několika fází. První fáze se zabývá transformací dat a provázáním jejich logiky, aby data byla vhodná pro simulační model. Ve druhé fázi tento model škáluji do reálné podoby výrobní haly dle nákresu poskytnutého samotnou společností. V této druhé fázi je navíc řešena logika zpětného toku prázdných přepravních vozíků, které jsou poté opět plněny operátory u vstřikolisů. Třetí fáze tvorby modelu slouží k návrhu indikátorů, jejich analýze a případným doporučením.

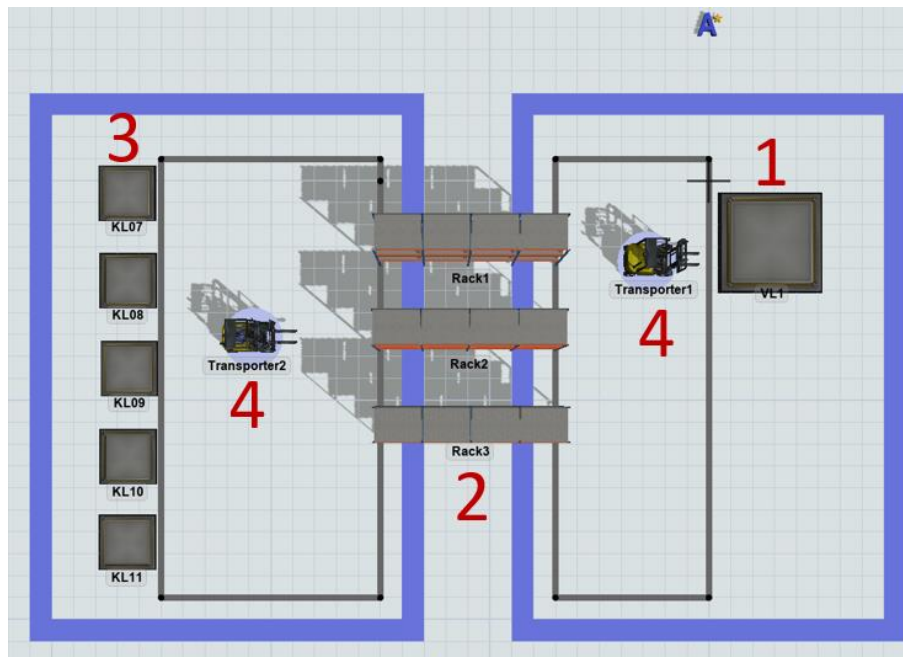
7

Simulační model – první fáze

Cílem první fáze řešeného simulačního modelu bylo kompletní nasimulování logiky propojující cestu ze vstřikolisů, přes gravitační regály až do kaširovacích lisů. K tomu byla vygenerována data související právě s jedním vstřikolisem jako zdrojem. Tento přístup byl zvolen vzhledem ke složitosti celého modelu, ať už jde o vymyšlení logiky rozhodování nebo množství objektů v oběhu.

Pro první fázi byl tedy zvolen jeden vstřikolis (1), 3 gravitační regály (2), několik kaširovacích lisů (3) a dva vysokozdvizné vozíky (4). Na následujícím obrázku se nachází pohled shora, čísla v obrázku a v závorce u objektů slouží jako reference. Tok materiálu začíná v pravé části, pokračuje přes regály do levé, kde se nachází kaširovací lisy. Dále můžeme vidět modré ohraničení, které určuje oblast, kam operátoři, nebo v našem případě vozíky, nemohou nebo kde je směr jízdy omezen.

Kromě modrého ohraničení se zde dále nachází i šedé, které funguje jako bariéra – neviditelná zeď. Toto ohraničení zde bylo přidáno, aby vysokozdvížeňové vozíky nenajížděly až do objektu.



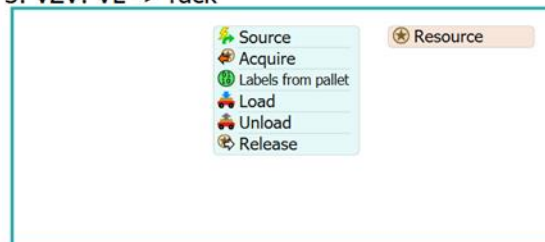
Obrázek 33: 3D model 1. fáze; vlastní zpracování

Po vytvoření funkčních objektů bylo nutné nastavit i logiku, a to pomocí procesního toku. Schéma vytvořeného procesního toku je na následujícím obrázku.

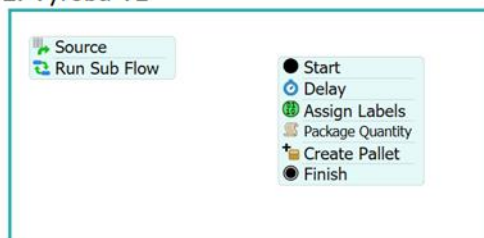
1. Počáteční naskladnění



3. VZV: VL -> rack



2. Výroba VL



4. VZV: rack -> KL



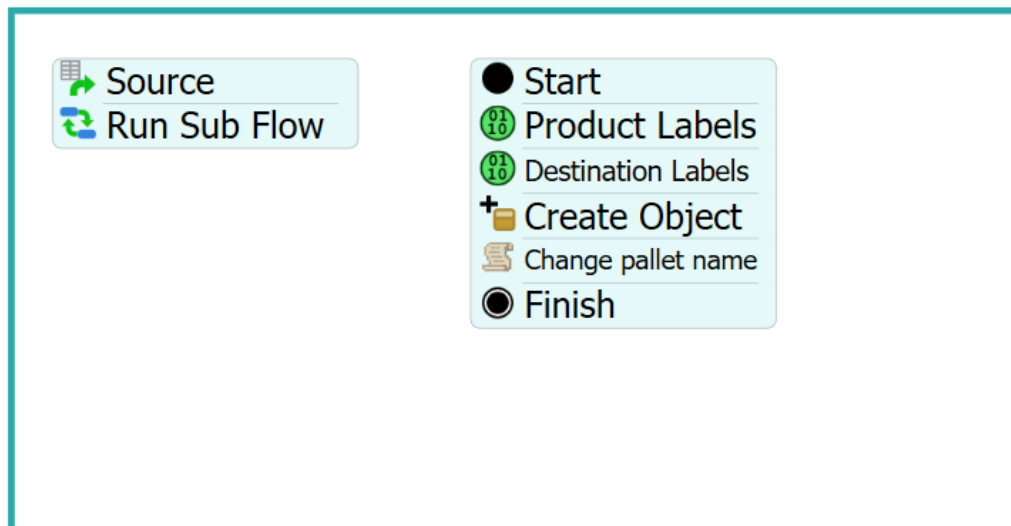
Obrázek 34: Procesní tok 1. fáze simulačního modelu; vlastní zpracování

Procesní tok tedy řídí logiku tohoto modelu, jednoduše řečeno zadává modelovým objektům, co mají vykonat a kdy. Jednotlivé části procesního toku jsou popsány v následujících podkapitolách.

7.1 Počáteční naskladnění

Prvním krokem je počáteční naskladnění palet v regálech – vytvoření počáteční zásoby. Tento údaj vychází z databázových dat, které nám říkají, jaké palety s jakými ID dílu byly na počátku dne již uloženy v regálech a zároveň i jejich počet – to znamená, že víme do jaké míry byly regály obsazené. Logiku naskladnění řídí první část procesního toku, která je na následujícím obrázku č. 35. Níže jsou potom v detailu popsány jednotlivé aktivity, které tuto část procesního toku tvoří.

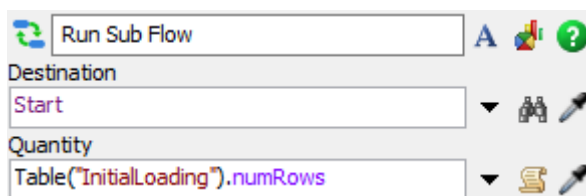
1. Počáteční naskladnění



Obrázek 35: Procesní tok počátečního naskladnění; vlastní zpracování

Zdroj (Source) – slouží k vytvoření tokenů, které postupně procházejí aktivitami.

Dílčí tok (Run Sub Flow) – má vlastnost, že token po vstoupení zůstane, dokud všechny jeho podřízené tokeny, vytvořené do aktivity Start, nedokončí svůj dílčí tok. Na následujícím obrázku můžeme vidět nastavení generování tokenů pro tento dílčí tok. Tento příkaz určí kolik tokenů má aktivita vytvořit, v našem případě to znamená podívat se do vybrané tabulky s daty (InitialLoading, tabulka č. 3) a vytvořit stejný počet tokenů, kolik má tabulka řádků.



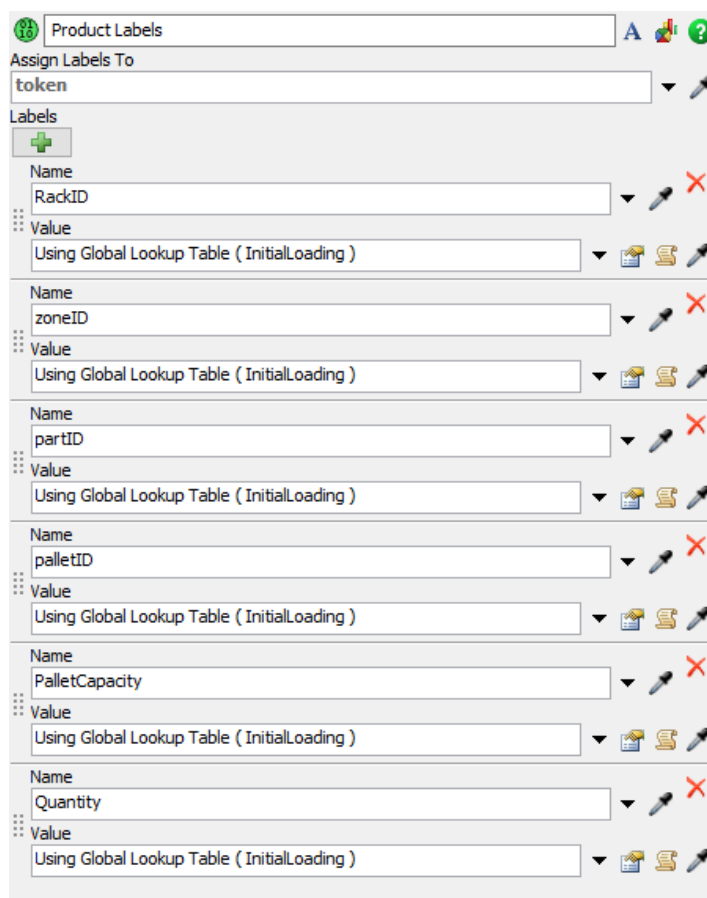
Obrázek 36: Dílčí tok naskladnění; vlastní zpracování

Tabulka 3: InitialLoading; vlastní zpracování

RackID	ZoneID	ProductID	Capacity	OccupiedCapacity	PalletID	PalletCapacity
331.A.024	IM front L	XVY16554	11.00000	3.00000	wmh152	33.00000
331.A.025	IM front L	XVY16555	11.00000	7.00000	wiml320	33.00000
331.A.025	IM front L	XVY16554	11.00000	10.00000	wmh152	33.00000
331.A.026	IM front L	XVY16555	11.00000	11.00000	wiml320	33.00000
331.A.026	IM front L	XVY16554	11.00000	0.00000	wmh152	33.00000
331.A.026	IM front L	XVY16555	11.00000	4.00000	wiml320	33.00000

Start – tato aktivita je obvykle první aktivitou v dílčím toku. Start však označuje pouze jeho začátek, neprovádí žádnou jinou logiku. Jakmile token vstoupí do aktivity Start, okamžitě se přesune na následující připojenou aktivitu v dílčím toku.

Přiřazení štítků 1 (Product Labels) – slouží k přiřazení štítků (atributů) každému tokenu, který do této aktivity vstoupí. V tomto případě token představuje přepravní vozík (v 3D modelu zobrazen jako paleta). Pro následující aktivity a správné řízení toku těchto vozíků napříč celým logistickým procesem vyžaduje token právě tyto štítky. Všechny informace, které jsou potřebné pro vozíky, se nacházejí v datové tabulce pro počáteční naskladnění. K vyhledání těchto informací slouží funkce Value Using Global Lookup Table napojená na tabulku č. 3 s názvem InitialLoading. Přiřazované atributy jsou ID gravitačního regálu, zóny, vyráběného dílu a ID vozíku. Další informací je maximální kapacita a také kolik těchto vozíků je na počátku směny již zaskladněno v regálu. Celý seznam je na následujícím obrázku.



Obrázek 37: Nastavení štítků pro tokeny; vlastní zpracování

Přiřazení štítků 2 (Destination Labels) – tento štítek určí, na který konkrétní regál, se má vozík vytvořit. Tato informace je nutná pro následující aktivitu Create object. Na obrázku č. 32 můžeme vidět, že přiřazení štítku není v tomto případě řešeno vyhledáním v datové tabulce, ale pomocí vlastního kódu (Custom Code). Štítek Kam vyhledá odpovídající gravitační regál, ve kterém má být vozík vytvořen. Kód na obrázku č. 33 bude pro každý token vyhledávat přiřazené ID gravitačního regálu.



Obrázek 38: Nastavení štítku pro umístění; vlastní zpracování

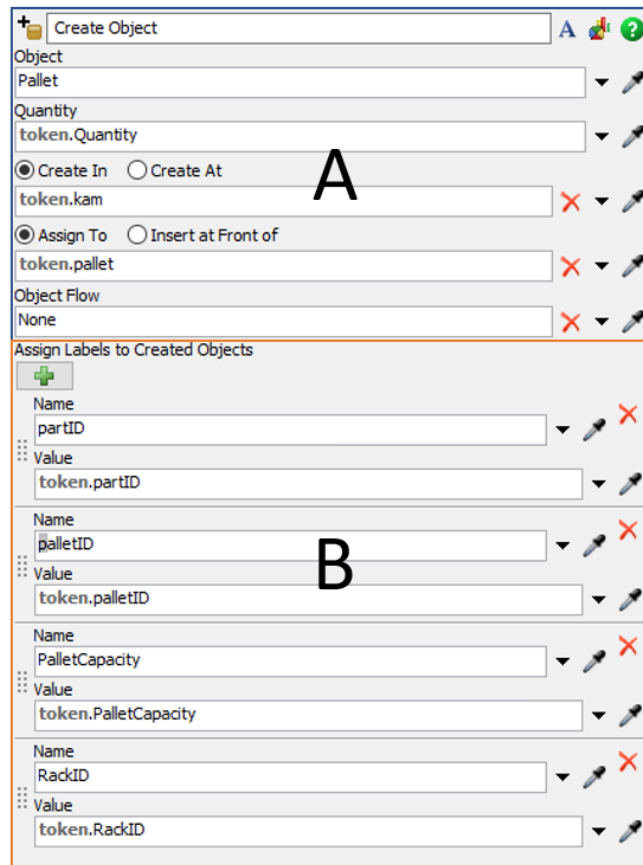
```

1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 Variant assignTo = param(4);
6 string labelName = param(5);
7 treenode processFlow = ownerobject(activity);
8
9 Object o = Model.find(token.RackID);
10 int a = 0;
11 return o;

```

Obrázek 39: Kód pro umístění palet; vlastní zpracování

Vytvoření objektu (Create Object) – tato aktivita generuje objekty v 3D modelu na základě atributů přiřazených v předchozích kroků v procesním toku. Vybraným 3D objektem je paleta a jejich počet je dán štítkem Quantity. Konkrétní gravitační regál určí štítek Kam (Přiřazení štítků 2 (Destination Labels)). Toto nastavení je na následujícím obrázku v části A. Těmto vytvořeným objektům jsou zároveň přiřazeny štítky z aktivity Přiřazení štítků 1. Bez tohoto kroku by se vygenerovaly prázdné palety v gravitačních regálech a tím by byl narušen celý logistický systém. Přiřazení štítků je na následujícím obrázku v oblasti B.



Obrázek 40: Generování objektů; vlastní zpracování

Vlastní kód (Change pallet name) – tato aktivita slouží k přejmenování vygenerovaných palet na základě štítku palletID. Celý kód je na následujícím obrázku.

```

1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 treenode processFlow = ownerobject(activity);
6
7 Object pal = Model.find(token.RackID + "/Pallet1");
8
9 if(token.Quantity == 0){
10
11 }
12 else {
13 pal.name = token.palletID;
14 }
15
16
17 int a = 0;

```

Obrázek 41: Kód pro pojmenování objektů; vlastní zpracování

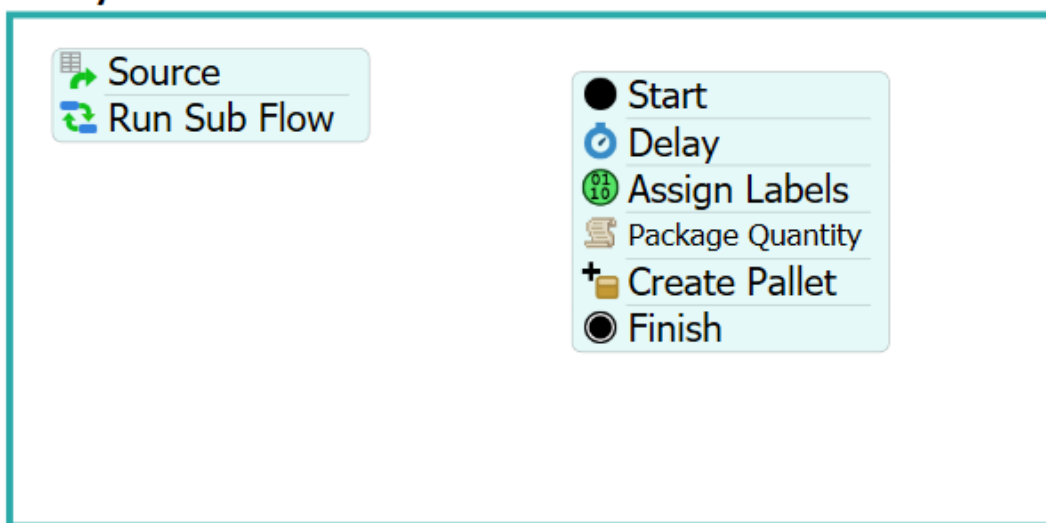
Dokončení (Finish) – tato aktivita uzavírá dílčí tok.

V první části procesního toku došlo k vygenerování konkrétních přepravních vozíků, které byly na počátku směny již zaskladněny. Tyto vozíky nesou všechny důležité informace, které umožní jejich zařazení do ostatních částí logistického systému a jejich další využití.

7.2 Výroba na vstřikolisech

Druhá část procesního toku řídí a generuje přepravní vozíky, které vstupují do logistického systému přímo z výroby. Přehled aktivit pro logiku generovaných vozíků je na následujícím obrázku. Níže jsou v detailu popsány jednotlivé aktivity, které řídí druhou část procesního toku. Aktivity, které se opakují pro každou část a nejsou jinak pozměněny, jsou vynechány. Jedná se o aktivity zdroj, dílčí tok, start a dokončení.

2. Výroba VL



Obrázek 42: Procesní tok palet z výroby; vlastní zpracování

Zpoždění (Delay) – tato aktivita drží vygenerovaný token po určenou dobu a tím vytváří časové rozestupy odpovídající času naplnění palety posledním dílem ze vstřikolisu. Zpoždění může být na základě pevně daných hodnot nebo např. statistického rozdělení. V datové tabulce (Vstup, tabulka č.4) jsou zaznamenány přesné časové údaje o vstupu a výstupu přepravního vozíku z pracovní stanice vstřikolisu. Tyto časy byly převedeny na sekundy a dále transformovány na interní čas simulačního běhu, a to pomocí vlastního

kódu. V kódu dojde k vyhledání času v sekundách a následného odečtení času simulačního běhu. Celý kód je na následujícím obrázku.

```

1 | **Custom Code**/
2 | Object current = param(1);
3 | treenode activity = param(2);
4 | Token token = param(3);
5 | treenode processFlow = ownerobject(activity);
6 |
7 | int tmp = token.row;
8 |
9 |
10 | Table tab = Table("Vstup");
11 | int cas = tab[tmp][10];
12 |
13 | int a = cas - Model.time;
14 | return a;
15 |

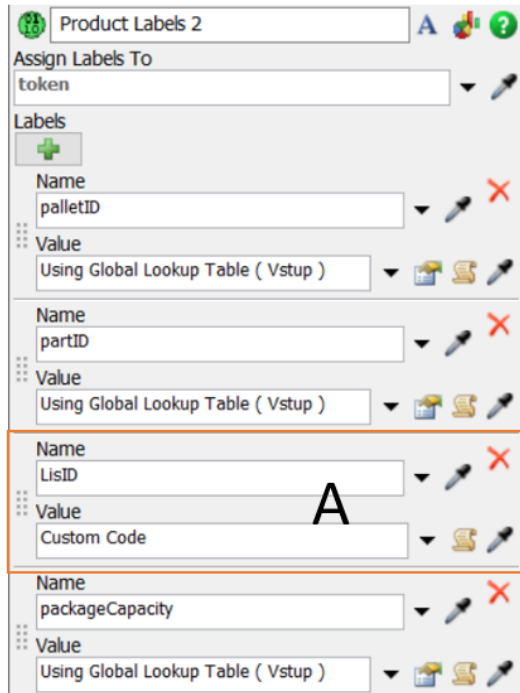
```

Obrázek 43: Kód aktivity zpoždění; vlastní zpracování

Tabulka 4: Vstup; vlastní zpracování

ID operace	ProductID	Quantity	StartTime	EndTime	SimTime	PalletID	PalletCapacity	LisID	RackID
I467794001	LRV851237	800.00000	06.02.2022 16:56	07.02.2022 2:04	84124	bsl115	1000.00000	IMM23	331.B.034
I466741001	LRV851237	800.00000	06.02.2022 16:56	07.02.2022 2:04	84124	bsl115	1000.00000	IMM23	331.B.034
I466747001	XVY16556	400.00000	06.02.2022 9:31	06.02.2022 14:19	36810	wchn0511	33.00000	IMM16	331.A.019
I466747007	XVY16551	400.00000	06.02.2022 9:31	06.02.2022 14:19	36810	vvm17172	33.00000	IMM16	331.A.019
I466664001	XVY16556	400.00000	06.02.2022 9:31	06.02.2022 14:19	36810	wchn0511	33.00000	IMM16	331.A.019
I466664007	XVY16551	400.00000	06.02.2022 9:31	06.02.2022 14:19	36810	vvm17172	33.00000	IMM16	331.A.019

Přiřazení štítků (Product Labels 2) – v této aktivitě jsou přiřazeny stejné štítky jako v aktivitě Product Labels 1 pro každý token generovaný v tomto dílčím toku. Je to z toho důvodu, že se opět jedná o přepravní vozíky, v tomto případě přicházející z výroby. Celý seznam štítků je na následujícím obrázku. V oblasti A je štítek navíc, jehož nastavení je řešeno pomocí vlastního kódu.



Obrázek 44: Přiřazení štítků pro vozíky z výroby; vlastní zpracování

Vlastní kód zde určí, jaký sloupec z datové tabulky má prohledat a vrátit hodnotu ID vstříkolisu.

```

1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 Variant assignTo = param(4);
6 string labelName = param(5);
7 treenode processFlow = ownerobject(activity);
8
9 Table scan = Table("Vstup");
10 int tmp = token.row; //ulozime do tmp místo psaní token.row
11
12 string LisID = scan[tmp][4];
13 return LisID;

```

Obrázek 45: Vlastní kód přiřazení ID vstříkolisu; vlastní zpracování

Vlastní kód (Package Quantity) – tento kód řeší vygenerování správného počtu přepravních vozíků. Signál v tabulce Vstup určuje celkové množství ks z výroby na jeden požadavek, ale musí se vzít v potaz maximální kapacita daného vozíku.

```

1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 treenode processFlow = ownerobject(activity);
6
7 Table scan = Table("Vstup");
8 int tmp = token.row; //ulozime do tmp místo psaní token.row
9
10 int packageQty = scan[tmp][3]; //množství v balení
11 int palletCapacity = scan [tmp][8];
12
13 int pocetPalet = packageQty / palletCapacity; //kolik palet chci vygenerovat pod vybranym jmenem
14
15 token.packageQty = pocetPalet;

```

Obrázek 46: Vlastní kód generování počtu vozíků; vlastní zpracování

Např. signál z výroby určí 1000 ks výrobku X, ale přepravní vozík má maximální kapacitu 250 ks, proto je nutné pomocí kódu vygenerovat 4 vozíky. Kód je na obrázku č. 46.

Vytvoření vozíků (Create Pallet) – tato aktivita generuje palety (představující vozíky) v 3D modelu na základě předchozích kroků v procesním toku. Nastavení aktivity na následujícím obrázku je opět rozděleno do dvou oblastí. Oblast A určuje typ objektu (paleta), počet palet (z předchozí aktivity vlastní kód) a místo, kde mají být generovány (štítek LisID z aktivity Product Labels 2). Přiřazení štítků z tokenů na vygenerované vozíky je na následujícím obrázku v oblasti B.

The screenshot shows the configuration for the 'Create Pallet' activity. It is divided into two main sections:

- Section A (General Object Creation):**
 - Object: Pallet
 - Quantity: token.packageQty
 - Create In (selected) / Create At
 - Model.find(token.LisID)
 - Assign To (selected) / Insert at Front of
 - token.pallet
 - Object Flow: None
- Section B (Assign Labels to Created Objects):**
 - Name: palletID, Value: token.palletID
 - Name: LisID, Value: token.LisID
 - Name: packageCapacity, Value: token.packageCapacity
 - Name: RackID, Value: token.RackID

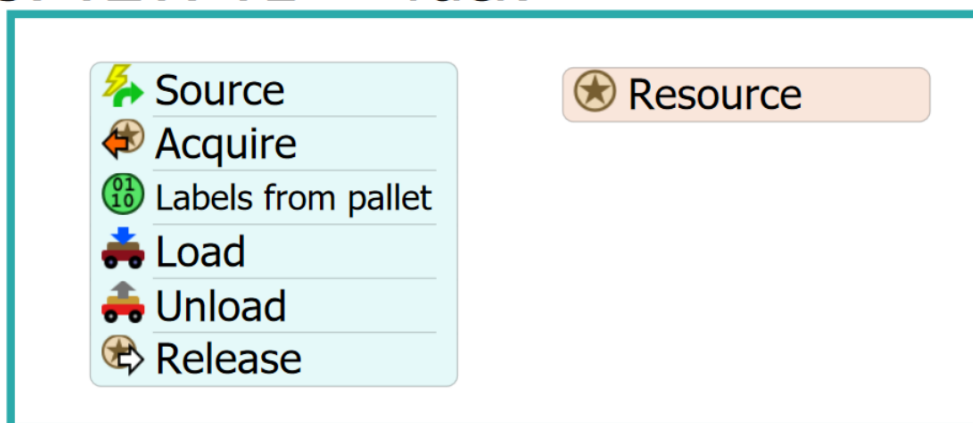
Obrázek 47: Generování objektů z výroby; vlastní zpracování

Druhou část procesního toku uzavírá aktivita dokončení. Tato část slouží k postupnému generování konkrétních přepravních vozíků na konkrétních vstřikolisech na základě signálů z výroby.

7.3 Odvoz vozíků od lisů do gravitačních regálů

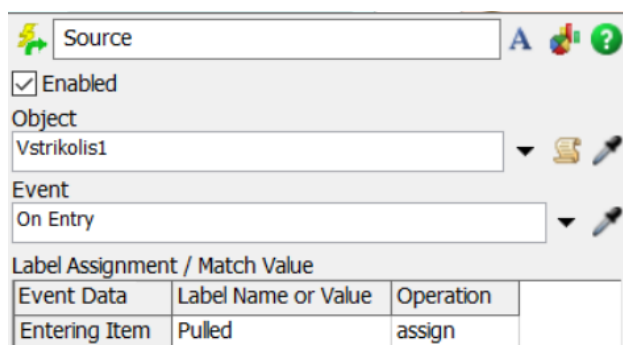
Třetí část procesního toku řídí zavážení naplněných přepravních vozíků od lisů do určených gravitačních regálů. Přehled aktivit pro logiku odvozu vozíků do gravitačních regálu je na následujícím obrázku. Níže jsou v detailu popsány jednotlivé aktivity, které řídí třetí část procesního toku.

3. VZV: VL -> rack



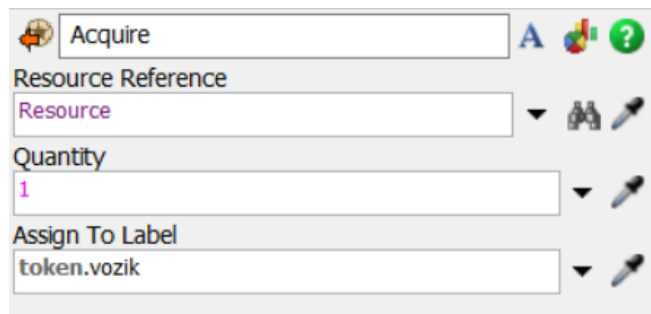
Obrázek 48: Procesní tok odvozu palet z výroby; vlastní zpracování

Zdroj (Event-triggered Source) – tento zdroj tokenů se liší od předchozích použitých zdrojů. Jelikož se jedná o zjednodušený model bylo vhodnější použít tuto variantu, protože jsou vozíky generovány pouze na jednom vstřikolis (viz obrázek č. 27). Pro každý vstupující token je přiřazen štítek Pulled, aby došlo k rozlišení, z jaké části logistického systému vozík pochází. Nastavení aktivity je na následujícím obrázku.



Obrázek 49: Aktivita zdroj; vlastní zpracování

Získání (Acquire) – tato aktivita slouží pro získání zdroje pro další aktivity a činnosti v procesním toku a 3D modelu. Nejedná se však o zdroj tokenů jako v předchozí aktivitě. V tomto případě se jedná o vysokozdvihný vozík, který bude odvážen generované palety ze vstřikolisu. V nastavení je nutný odkaz na tento zdroj pomocí níže popsané aktivity (Zdroj VZV (Resource)). Dále je nutné přiřadit štítek k vozíku a jejich požadovaný počet. Nastavení aktivity je na následujícím obrázku.



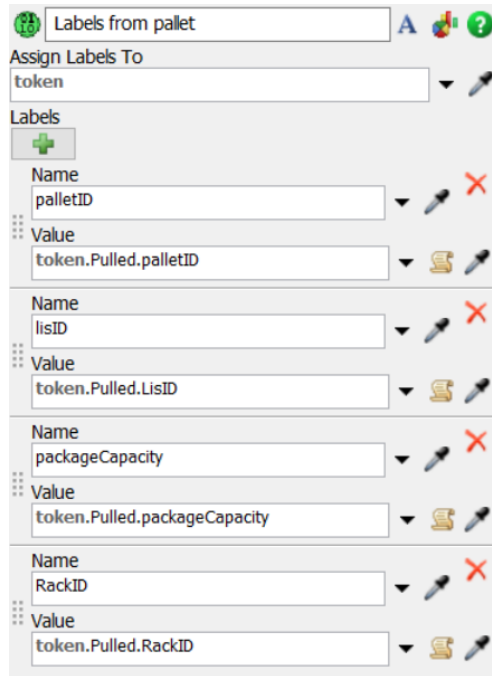
Obrázek 50: Aktivita získání zdroje; vlastní zpracování

Zdroj VZV (Resource) – aktivita, která vytváří vazbu na získání a uvolnění zdroje (VZV) pro manipulaci s tokeny (v 3D modelu s paletami). Pro vazbu je nutné vybrat konkrétní objekt v modelu, což je VZV s názvem Transporter1. Nastavení je na následujícím obrázku.



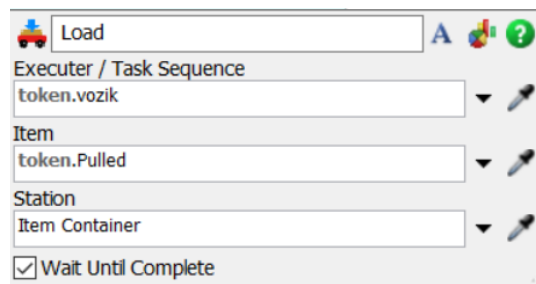
Obrázek 51: Zdroj VZV; vlastní zpracování

Přiřazení štítků (Labels from pallet) – v první aktivitě zdroj byl tokenu přiřazen štítek Pulled. Pro tyto tokeny Pulled byly dále přiřazeny štítky se stejným typem informací jako v předchozích aktivitách přiřazení štítku. Detailní seznam je na následujícím obrázku.



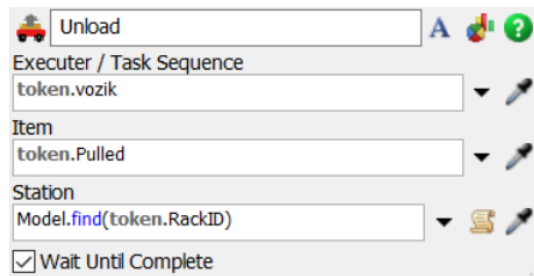
Obrázek 52: Přiřazení štítku tokenům; vlastní zpracování

Naložení (Load) – aktivita, která slouží k provedení akce naložení přepravního vozíku. Je nutné určit jaký zdroj tuto akci má provést a co má naložit. V tomto případě akci provede VZV Transporter1 (token.vozik) a naloží vozíky (tokeny) se štítkem Pulled. Detail nastavení je na následujícím obrázku.



Obrázek 53: Aktivita naložení; vlastní zpracování

Vyložení (Unload) – aktivita, která slouží k provedení akce vyložení přepravního vozíku do gravitačního regálu. Je nutné opět určit jaký zdroj tuto akci má provést a co má vyložit. V tomto případě akci provede VZV Transporter1 (token.vozik.) a vyloží vozíky (tokeny) se štítkem Pulled. Zároveň je nutné vyložit přepravní vozík do správného regálu a k tomu slouží štítek RackID a příkaz Model.find, který nalezne správný regál. Detail této aktivity je na následujícím obrázku.



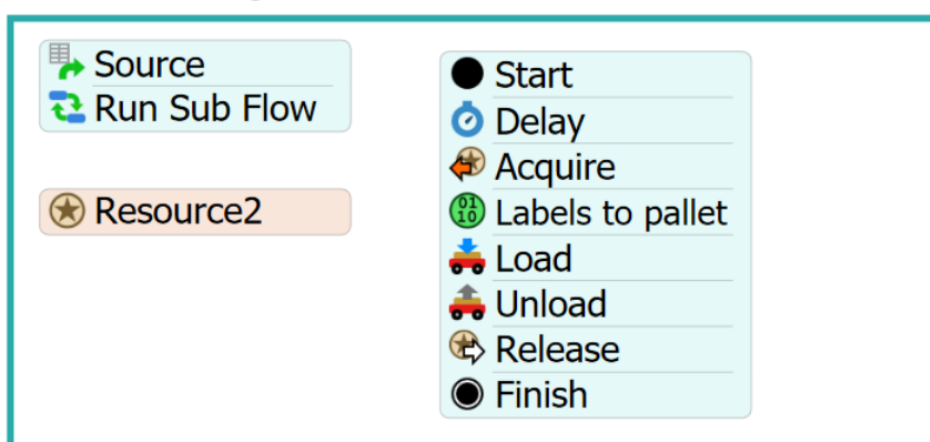
Obrázek 54: Aktivita vyložení; vlastní zpracování

Uvolnění (Release) – aktivita, která po dokončení předchozích aktivit zdroj VZV uvolní. Po uvolnění se tento tok opakuje a VZV dostane další signál pro naložení a vyložení dalšího přepravního vozíku.

7.4 Odvoz vozíků z regálu ke kaširovacím lisům

Čtvrtá a poslední část procesního toku řídí odvoz přepravních vozíků z gravitačních regálů na kaširovací lisy. Přehled aktivit pro řízení odvozu vozíků do kaširovacích lisů je na následujícím obrázku. Jednotlivé aktivity jsou popsány níže. Aktivity, které se opakují z předchozích podkapitol a nejsou výrazně pozměněny, jsou vynechány. Jedná se o aktivity zpoždění (Delay), získání (Acquire), zdroj VZV (Resource) a uvolnění (Release). Jedinou změnou u těchto aktivit je využití jiné datové tabulky se stejnou podobou dat a využití druhého VZV s názvem Transporter2.

4. VZV: rack -> KL



Obrázek 55: Procesní tok pro odvoz palet z regálů; vlastní zpracování

```
1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 Variant assignTo = param(4);
6 string labelName = param(5);
7 treenode processFlow = ownerobject(activity);
8
9 string Paleta = token.palletID;
10 string Rack = token.RackID;
11 Object Pallet = Model.find(Rack+"/"+Paleta);
12 return Pallet;
```

Obrázek 56: Vlastní kód vyhledání vozíku v regálu; vlastní zpracování

Přiřazení štítků (Labels to Pallet) – tato aktivita je z velké části stejná jako předchozí aktivity přiřazení štítků. Opět se jedná o ID regálů, součástí, kapacit apod. Obsahuje však navíc štítek sloužící k identifikaci cílového kaširovacího lisu. Tyto štítky jsou přiřazeny přímo z datové tabulky (Výstup, tabulka č.5). Má stejný formát jako tabulka 2. části procesního toku. Je zde přidán ještě štítek whichPallet, na jehož základě může druhý VZV vyhledat požadovaný přepravní vozík v gravitačním regálu. Řešení pomocí vlastního kódu je na následujícím obrázku.

Tabulka 5: Výstup; vlastní zpracování

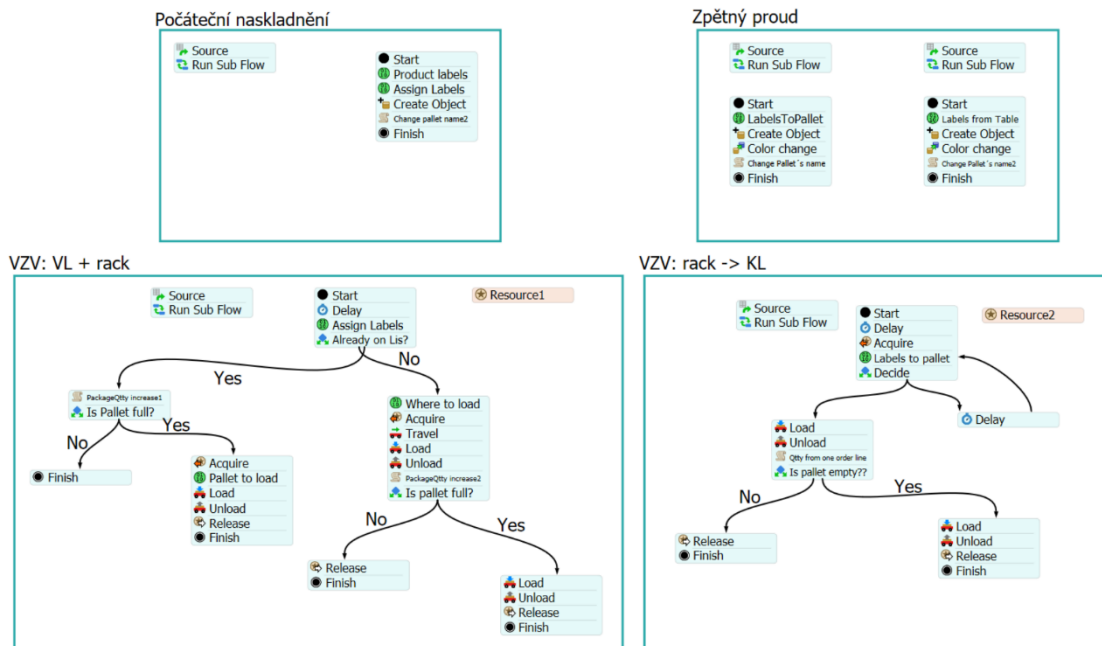
ID operace	ProductID	Quantity	StartTime	EndTime	SimTime	PalletID	LisID	RackID
L413394001	RSE84421	-16	06.02.2022 14:56	07.02.2022 15:56	2364	ARN0014	KL07	331.A.023
L411341001	RSE84421	-16	06.02.2022 14:56	07.02.2022 15:56	2364	ARN0014	KL07	331.A.023
L411343001	XVZ16554	-16	06.02.2022 7:31	06.02.2022 14:19	5584	OPL123	KL08	331.A.023
L411343003	XVZ16552	-16	06.02.2022 7:31	06.02.2022 14:19	5584	OPL123	KL08	331.A.023
L411114001	XVZ16554	-16	06.02.2022 7:31	06.02.2022 14:19	5584	wchn0511	KL08	331.A.023
L411114003	XVZ16552	-16	06.02.2022 7:31	06.02.2022 14:19	5584	vvml7172	KL08	331.A.023

Princip této poslední části procesního toku je stejný jako u předchozí části. Postupně jsou posílány signály vysokozdviznému vozíku, který vyhledá požadovaný přepravní vozík v regálu, zaveze ho na požadovanou pozici a čeká na další signál. Vždy musí nejprve dokončit jednu akci, než začne zpracovávat další signál.

První verze simulačního modelu představuje prvotní implementaci koncepčního modelu, přenesení poznatků z návštěvy společnosti, zpracování dat a převážně vytvoření procesního toku. Procesní tok je dále rozšířen v druhé fázi simulačního modelu, a to konkrétně o zpětný proud.

8 Simulační model – druhá fáze

Cílem druhé fáze simulačního modelu je rozšíření modelu a využití kompletního souboru dat. Je zde několik úprav procesního toku, který je rozšířen o rozhodovací aktivity a hlavně zpětný proud. Zpětný proud zpracovaných palet pokrývá tok přepravních vozíků, které jsou již zpracovány na kaširovacích lisech a vrací se zpět do oběhu pro nové výrobní požadavky na vstřikolisech. Zpětný proud také zahrnuje počáteční naskladnění prázdných vozíků (palet), což je principiálně stejné jako počáteční naskladnění naplněných vozíků v gravitačních regálech. Ve výrobní hale jsou k tomuto zpětnému proudu vyhrazeny gravitační regály, odkud proudí prázdné palety zpět. V první fázi modelu byl tento proces vynechán a palety byly dle potřeby generovány přímo na vstřikolisech. Z tohoto důvodu je nyní nutné vymodelovat systém detailněji.



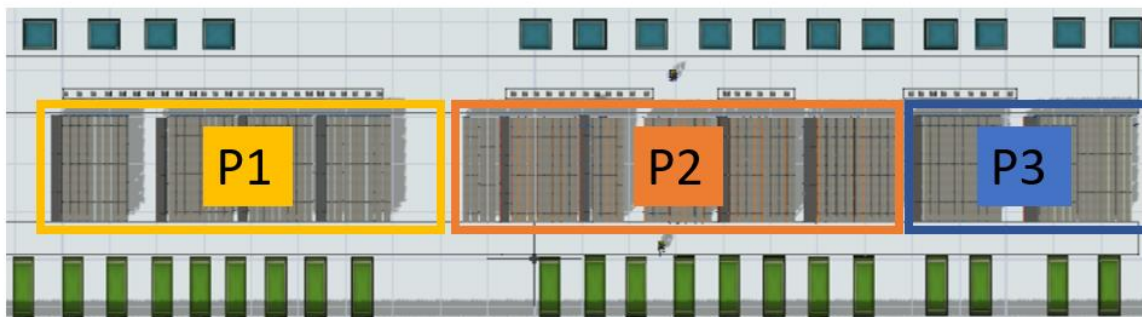
Obrázek 57: Procesní tok 2. fáze simulačního modelu; vlastní zpracování

Na obrázku č. 58 je pohled shora. Oblast A zahrnuje vstřikolisy (celkem 30) a jeden VZV, v oblasti B jsou gravitační regály pro zaskladnění (celkem 95) a gravitační regály pro zpětný proud (celkem 10). Oblast C zahrnuje kaširovací lisy (celkem 20) a druhý VZV.

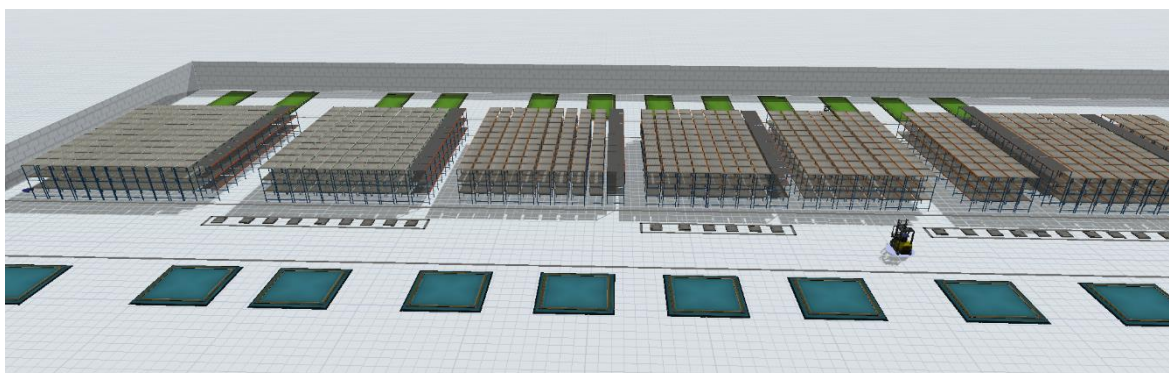
Samotné regály jsou rozděleny na 3 části. Toto rozdělení je na základě zpracovávaných projektů a jim odpovídajících výrobků. Výrobky mohou být naskladněny pouze do předem určených zón, které seskupují několik regálů. Toto rozdělení je na obrázku č. 59. Na obrázku č. 60 je část haly v 3D modelu.



Obrázek 58: Rozložení haly; vlastní zpracování



Obrázek 59: Rozložení regálů na projekty; vlastní zpracování



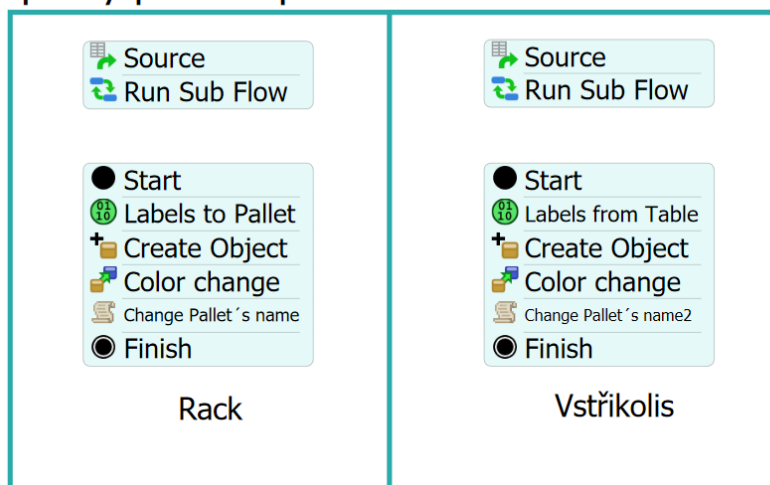
Obrázek 60: 3D model haly – pohled od kaširovacích lisů; vlastní zpracování

Procesní tok pro finální verzi simulačního modelu je na následujícím obrázku. Byla přidána část řídicí zpětný proud a dále byly upraveny původní části procesního toku. Část procesního toku, která řídí počáteční naskladnění naplněných přepravních vozíků nebyla nijak pozměněna, proto bude v této části vynechána. V detailu budou v podkapitolách níže popsány ostatní části procesního toku, které byly upraveny.

8.1 Počáteční naskladnění prázdných vozíků

Logika zpětného proudu je rozdělena do dvou částí. Procesní tok pro prázdné vozíky je na následujícím obrázku. Pravá část slouží k vytvoření počátečních prázdných přepravních vozíků u vstřikolisů. Na počátku simulačního běhu jsou totiž vozíky pouze na některých vstřikolisech a nejprve musí být naplněny. Tímto se liší simulační model v první fázi, kde byly generovány jeden za druhým. Levá část slouží k vygenerování prázdných vozíků ve vybraných gravitačních regálech, které jsou součástí zpětného proudu.

Zpětný proud - počáteční naskladnění



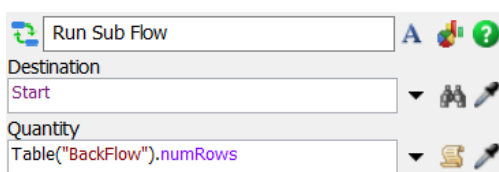
Obrázek 61: Procesní tok pro generování prázdných vozíků; vlastní zpracování

8.1.1 Zpětný proud – levá část

Níže jsou v detailu popsány jednotlivé aktivity, které tvoří levou část procesního toku.

Zdroj (Source) – slouží k vytvoření tokenů, které postupně procházejí aktivitami.

Dílčí tok (Run Sub Flow) – sjednocuje aktivity, kterými tokeny musí projít. Nastavení počtu tokenů, které mají být vygenerovány, je na následujícím obrázku. Princip je stejný jako u předchozích dílčích toků, což znamená, že bude vygenerováno tolik tokenů, kolik je řádků ve vybrané tabulce (BackFlow, tabulka č.6).



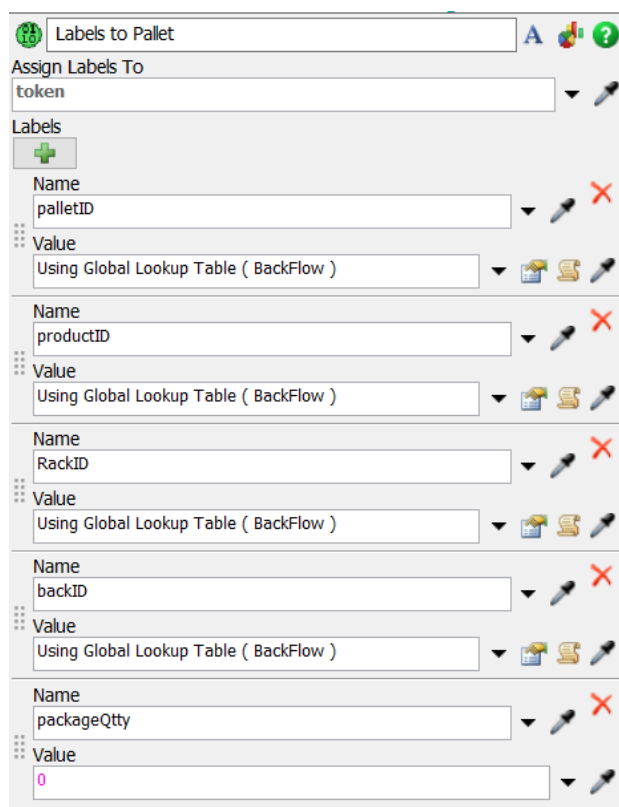
Obrázek 62: Nastavení dílčího toku 1; vlastní zpracování

Tabulka 6: BackFlow; vlastní zpracování

PalletID	ProductID	RackID	BackRack
uxhl4161	LLP472146	331.A.025	/Z1_331A
lam1114	XHG2861	331.A.007	/Z1_331B
VVNL1358	WER46002	331.B.022	/Z1_331A
uxhl4161	LLP472146	331.A.025	/Z1_331A
VVNL1358	WER46002	331.A.025	/Z1_331A
uxhl4161	LLP472146	331.A.025	/Z1_331A

Start – počátek bloku aktivit.

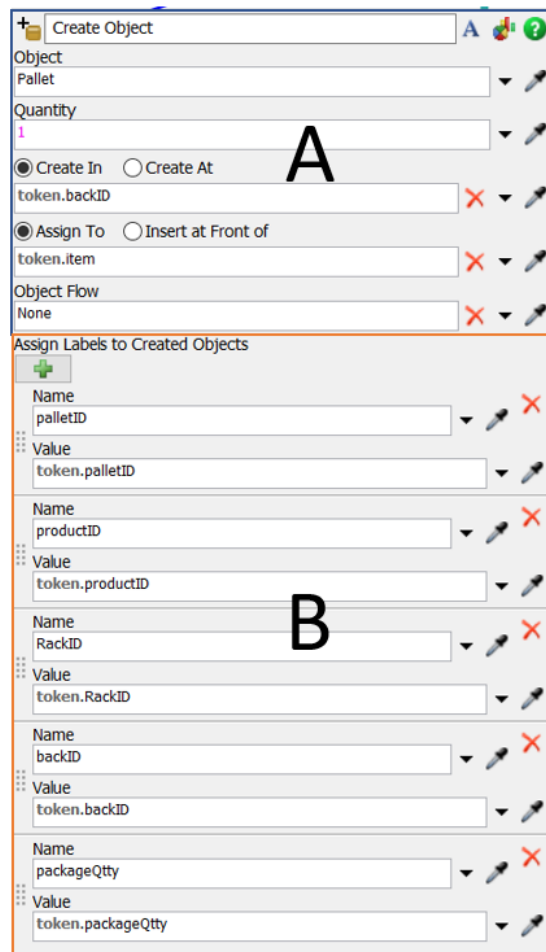
Přiřazení štítků (Labels to Pallet) – aktivita, ve které jsou tokenům přiřazeny štítky. K vyhledání hodnot pro všechny štítky v této aktivitě slouží Value Using Global Lookup Table s názvem BackFlow. Jedná se o ID gravitačního regálu, vyráběného dílu, ID vozíku a také gravitačního regálu pro zpětný proud. Poslední štítek nesoucí informaci o počtu vyrobených ks ve vozíku má pevně stanovenou hodnotu na nulu, jelikož jsou vozíky v tomto procesu prázdné. Všechny štítky jsou na následujícím obrázku.



Obrázek 63: Nastavení štítků pro vozíky - zpětný regál; vlastní zpracování

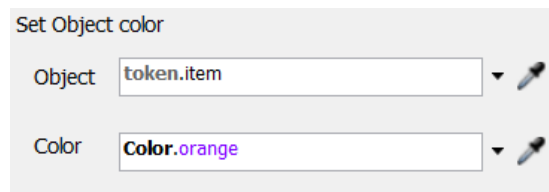
Vytvoření objektu (Create Object) – tato aktivita generuje palety, které představují přepravní vozíky. Jeden token znamená vygenerování jedné palety. Umístění ve zpětném k gravitačním regálu určí štítek backID. Toto nastavení je na následujícím obrázku v části A.

Těmto vytvořeným objektům jsou zároveň přiřazeny štítky z předchozí aktivity Přiřazení štítků (Labels to Pallet). Přiřazení štítků je na následujícím obrázku v oblasti B.



Obrázek 64: Generování palet ve zpětných regálech; vlastní zpracování

Změna barvy (Color change) – aktivita, která slouží ke změně barvy objektu. Do procesního toku byla přidána z důvodu lepšího přehledu v 3D modelu. Barvu palety ve zpětném gravitačním regálu změní na oranžovou.



Obrázek 65: Změna barvy 1; vlastní zpracování

Vlastní kód (Change pallets name 1) – tato aktivita slouží k pojmenování vygenerovaných palet na základě štítku palletID. Celý kód je na následujícím obrázku.

```

1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 treenode processFlow = ownerobject(activity);
6
7 string back = token.backID.name;
8
9 Object pal = Model.find(back + "/Pallet1");
10 pal.name = token.palletID;

```

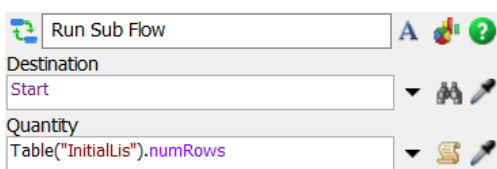
Obrázek 64: Vlastní kód pro pojmenování palet 1; vlastní zpracování

Dokončení (Finish) – uzavření dílčího toku pro generování prázdných přepravních vozíků (palet) v gravitačních regálech pro zpětný proud.

8.1.2 Zpětný proud – pravá část

Níže jsou v detailu popsány jednotlivé aktivity, které tvoří pravou část procesního toku (obrázek č. 55). Aktivity, které jsou totožné s předchozí částí budou vynechány. Jedná se o aktivity zdroj (Source), start (Start), dokončení (Finish).

Dílčí tok (Run Sub Flow) – aktivita, která řídí vygenerování tolika tokenů, kolik je řádků ve vybrané tabulce (InitialLis, tabulka č.7).



Obrázek 65: Nastavení dílčího toku 2; vlastní zpracování

Tabulka 7: InitialLis; vlastní zpracování

PalletID	Capacity	RackID	LisID
fno961	1000	331.B.021	IMM22
lam110	33	331.A.019	IMM17
wmkl58	25	331.B.011	IMM20
fno961	20	331.B.017	IMM28
wmkl58	25	331.B.011	IMM20
fno961	33	331.A.019	IMM16

Přiřazení štítků (Labels from Table) – aktivita, ve které jsou tokenům přiřazeny štítky. Štítek BackID z předchozí části procesního toku je nahrazen za štítek LisID a navíc přibyl štítek Where. Tento štítek slouží v další aktivitě k určení místa, kde má být paleta vygenerována. Hodnota štítku je řešena pomocí vlastního kódu, který je na obrázku č. 67. Přehled všech štítků aktivity je na obrázku č. 68.

```

1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 Variant assignTo = param(4);
6 string labelName = param(5);
7 treenode processFlow = ownerobject(activity);
8
9
10 Object lis = Model.find(token.LisID);
11 return lis;

```

Obrázek 68: Vlastní kód pro štítek Where; vlastní zpracování

The screenshot shows the configuration for the 'Labels from Table' activity. At the top, the title is 'Labels from Table'. Below it, the 'Assign Labels To' field is set to 'token'. There are several 'Labels' listed, each with a 'Name' and a 'Value' field. The labels are:

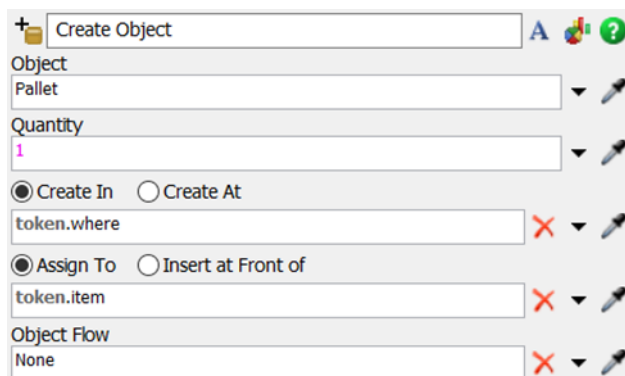
- Name: palletID, Value: Using Global Lookup Table (InitialLis)
- Name: Capacity, Value: Using Global Lookup Table (InitialLis)
- Name: RackID, Value: Using Global Lookup Table (InitialLis)
- Name: LisID, Value: Using Global Lookup Table (InitialLis)
- Name: where, Value: Custom Code
- Name: packageQty, Value: 0

Each label entry has a red 'X' icon to its right, indicating it can be removed. There are also icons for adding, editing, and deleting labels.

Obrázek 69: Nastavení štítků pro vozíky - VL; vlastní zpracování

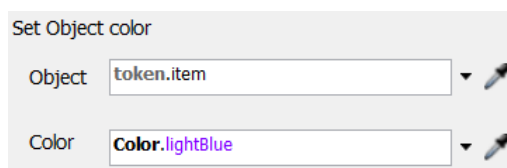
Vytvoření objektu (Create Object) – tato aktivita generuje palety, které představují přepravní vozíky. Jeden token znamená vygenerování jedné palety. Na jakém vstřikolisu se má vygenerovat určuje štítek where z předchozí aktivity. To je zároveň jediná změna u této

aktivity. Místo generování palet na zpětných regálech dojde ke generování palet přímo u vstřikolisů



Obrázek 66: Generování palet u vstřikolisů; vlastní zpracování

Změna barvy (Color change) – aktivita, která slouží ke změně barvy objektu. Do procesního toku byla přidána z důvodu lepšího přehledu v 3D modelu. Prázdné palety u vstřikolisů mají světle modrou barvu. Nastavení je na následujícím obrázku.



Obrázek 67: Změna barvy 2; vlastní zpracování

Vlastní kód (Change pallets name 2) – tato aktivita slouží k pojmenování vygenerovaných palet na základě štítku palletID. Celý kód je na následujícím obrázku.

```
1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 treenode processFlow = ownerobject(activity);
6
7
8 Object pal = Model.find(token.LisID + "/Pallet1");
9 pal.name = token.palletID;
```

Obrázek 68: Vlastní kód pro pojmenování palet 2; vlastní zpracování

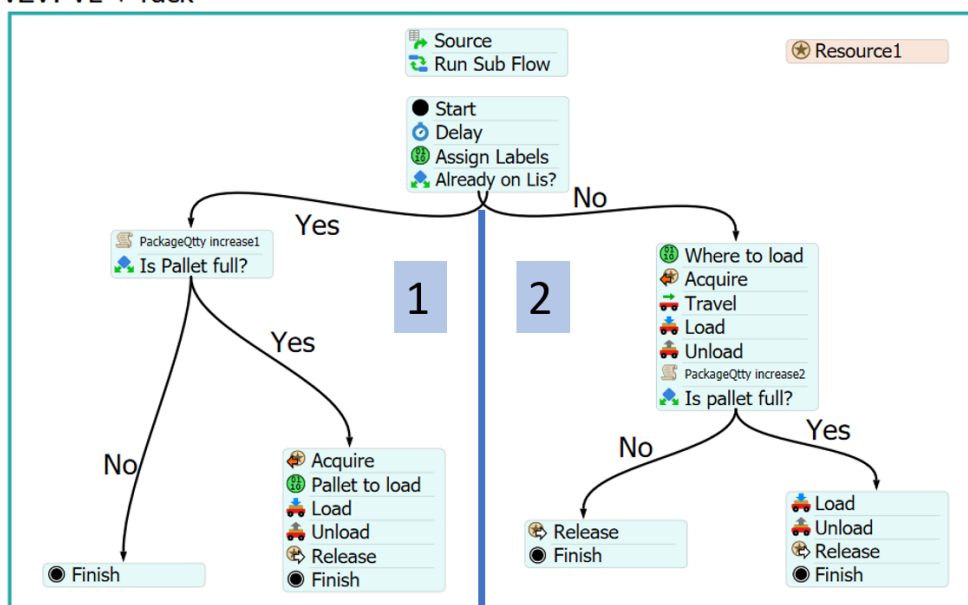
Dokončení (Finish) – uzavření dílčího toku pro generování prázdných přepravních vozíků (palet) u vstřikolisů.

Levá i pravá část procesního toku pro zpětný proud řeší generování prázdných palet na předem určených místech v hale. Jedná se o část vstřikolisů a o zpětné gravitační regály.

8.2 VZV1: vstřikolisy a gravitační regály

Tato část procesního toku řídí VZV1 a logiku oblasti A+B z obrázku č. 57. Tok je rozšířen především o několik rozhodovacích bodů. Pro lepší přehled a popis je tato část procesního toku rozdělena. Nejprve bude popsána společná část, poté část 1 a v poslední řadě část 2. Celý procesní tok pro tuto logiku je na následujícím obrázku.

VZV: VL + rack



Obrázek 69: Procesní tok pro řízení VZV1; vlastní zpracování

8.2.1 VZV1 – společná část

Zdroj (Source) – aktivita, která generuje tokeny.

Dílčí tok (Run Sub Flow) – aktivita, která generuje tolik dílčích tokenů, kolik je řádků ve vybrané tabulce. V tomto případě se jedná o tabulku Vstup.

Start – aktivita, která značí počátek dílčího toku.

Zpoždění (Delay) – tato aktivita drží vygenerovaný token po určenou dobu. Aktivita řešena pomocí vlastního kódu. V kódu dojde k vyhledání času v sekundách a následného odečtení času simulačního běhu.

Přiřazení štítků 1 (Assign Labels) – přiřazení štítků tokenům. Tyto štítky obsahují důležité informace pro správné řízení simulačního modelu. Jedná se ID palety, regálu, dílů, maximální kapacity atd.

Výše zmíněné aktivity (zdroj až přiřazení štítků 1) jsou identické aktivitám z první fáze simulačního modelu viz. Kapitola 7.1. Z toho důvodu již nejsou v této části rozepsány do detailu. Následující aktivity byly přidány v rámci rozšíření a upřesnění simulačního modelu z první fáze a proto jsou popsány detailněji.

Rozhodnutí 1 (Already on Lis?) – v této aktivitě obecně dochází k rozhodnutí, kam má být token poslán. Logika může být řešena parametrem, statistickým rozdělením, vyhledáním v tabulce nebo vlastním kódem. Na základě této logiky je pak pomocí konektoru token poslán do další aktivity. Logika této konkrétní aktivity je řešena pomocí vlastního kódu na následujícím obrázku. Tento kód slouží k vyhledání konkrétních palety a příslušejícího lisu v 3D modelu. V případě, že tuto paletu nenalezne, je token poslán přes konektor 1 (NE). V případě, že paletu na lisu najde, je token poslán konektorem 2 (ANO).

```

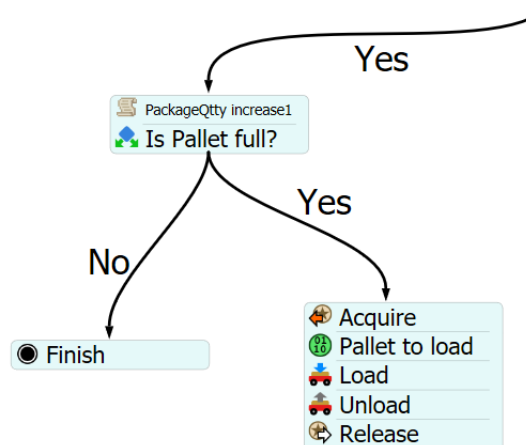
1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 treenode processFlow = ownerobject(activity);
6
7 string pallet = token.palletID;
8 string lis = token.LisID;
9
10 Object konkretni = Model.find(lis + "/" + pallet);
11
12 if (konkretni == NULL){
13     return 1;
14 }
15 else return 2;

```

Obrázek 70: Vlastní kód rozhodnutí 1; vlastní zpracování

8.2.2 VZV1 – levá část

Následuje popis aktivit v případě odeslání tokenu přes konektor ANO. Tyto aktivity jsou na následujícím obrázku.



Obrázek 71: Aktivity v bloku rozhodnutí 1 - ANO; vlastní zpracování

Vlastní kód (PackageQty increase1) – tento kód řeší zvýšení množství ks výrobků na paletě na základě datové tabulky a příslušného údaje, který určuje, kolik bylo vyrobeno ks. Celý kód s komentáři je na následujícím obrázku.

Rozhodnutí 2 (Is Pallet full?) – na základě vlastního kódu dojde k rozhodnutí, zda paleta dosáhla plné kapacity. Celý kód s komentáři je na následujícím obrázku.

V případě, že kapacita nedosáhla maxima, tak je tento token poslán do aktivity dokončení a je dále zpracován další token (obrázek č. 72 – vlevo). Tento proces se v toku opakuje, dokud paleta nemá naplněnou kapacitu. V případě, že je tato kapacita naplněna, je token odeslán druhým konektorem do následujícího bloku aktivit (obrázek č. 72 – vpravo).

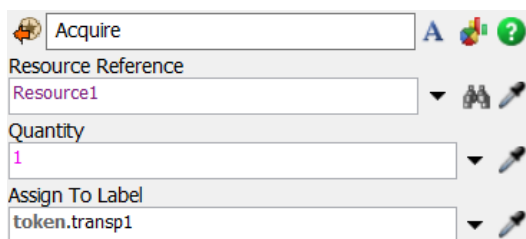
```
1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 treenode processFlow = ownerobject(activity);
6
7 Object pal = Model.find(token.LisID + "/" + token.palletID); // konkrétní paleta
8 // pro povysení množství balení
9 int capacity = token.packageCapacity; // kapacita na paletě
10 int suma = token.amount; // množství na jedné radce orderu, které chci na danou paletu přidat
11
12 pal.packageQty += suma; // přidávám množství
```

Obrázek 72: Vlastní kód pro navýšení množství na paletě; vlastní zpracování

```
1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 treenode processFlow = ownerobject(activity);
6
7
8 Object pal = Model.find(token.LisID + "/" + token.palletID); // konkrétní paleta
9 if(pal.packageQty == pal.Capacity){ // naplnila se celá paleta
10     return 1; // mohu ji zavést
11 }
12 else return 2; // ještě potřebuji další orderý na zaplnění palety
```

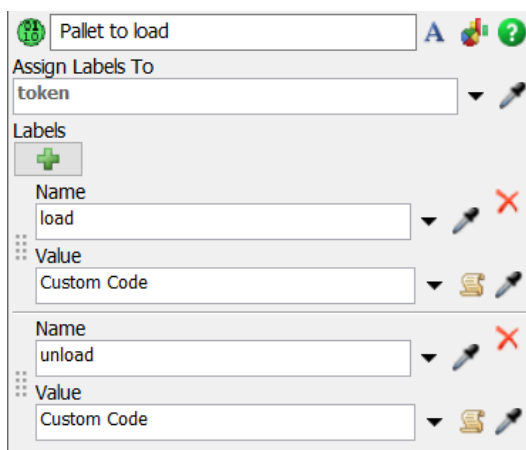
Obrázek 73: Vlastní kód pro kontrolu kapacity palety; vlastní zpracování

Získání (Acquire) – získání zdroje. V tomto případě se jedná o vysokozdvizný vozík, který bude odvážen generované palety. V nastavení je nutný odkaz na tento zdroj pomocí níže popsané aktivity (Resource1). Dále je nutné přiřadit štítek k vozíku (token.transp1) a jejich potřebný počet, který je 1. Nastavení aktivity je na následujícím obrázku.



Obrázek 74: Aktivita získání zdroje VZV1; vlastní zpracování

Přiřazení štítků 2 (Pallet to load) – aktivita, ve které jsou přiřazeny tokenu dva štítky. Jedná se o load a unload, které budou využity ve dvou následujících aktivitách. Na obrázku č. 78 je celkové nastavení aktivity. Hodnoty obou štítků jsou řešeny pomocí vlastního kódu. Štítek load slouží k vyhledání určené palety na daném lisu v modelu. Celý kód je na obrázku č 79.



Obrázek 75: Přiřazení štítků load a unload; vlastní zpracování

Štítek unload slouží k určení gravitační regálu, do kterého bude paleta odvezena. Kód je na obrázku č. 80.

```

1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 Variant assignTo = param(4);
6 string labelName = param(5);
7 treenode processFlow = ownerobject(activity);
8
9 Object pal = Model.find(token.LisID + "/" + token.palletID);
10 return pal;

```

Obrázek 76: Vlastní kód štítku load; vlastní zpracování

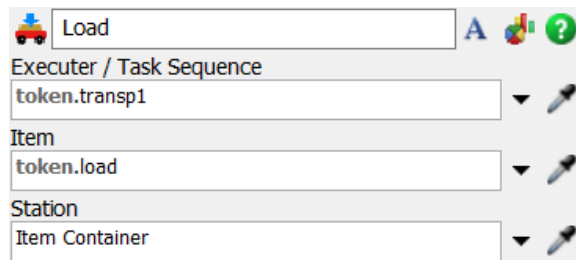
```

1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 Variant assignTo = param(4);
6 string labelName = param(5);
7 treenode processFlow = ownerobject(activity);
8
9
10 Object pal = Model.find(token.LisID + "/" + token.palletID);
11 string where = pal.rackID; // na palete mam informaci, do jakého racku patri
12 return where;

```

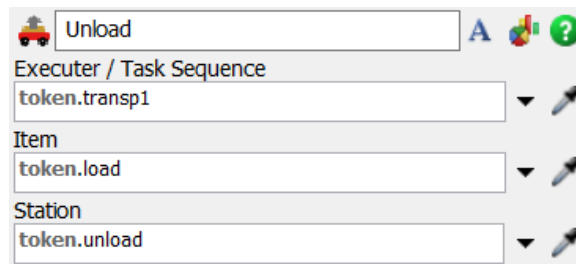
Obrázek 77: Vlastní kód štítku unload; vlastní zpracování

Naložení 1 (Load) – aktivita, která zadá VZV1 úkol naložit vybranou paletu z lisu na základě štítku load. Nastavení je na následujícím obrázku.



Obrázek 78: Aktivita naložit pro VZV1; vlastní zpracování

Vyložení 1 (Unload) – aktivita, která zadá VZV1 úkol vyložit naloženou paletu na vybraný gravitační regál na základě štítku unload. Nastavení je na následujícím obrázku.



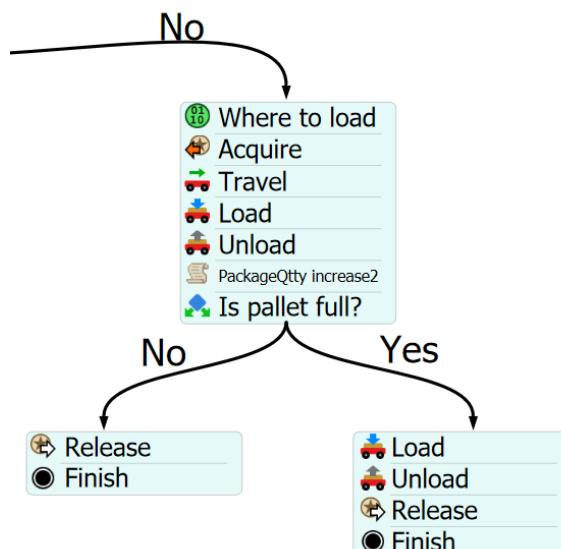
Obrázek 79: Aktivita vyložit pro VZV1; vlastní zpracování

Uvolnění (Release) – uvolnění zdroje VZV1.

Dokončení (Finish) – závěrečná aktivita výše popsaného bloku.

8.2.3 VZV1 – pravá část

Následuje popis aktivit v případě odeslání tokenu přes konektor NE (obrázek č. 78). Tyto aktivity řeší logiku, kdy paleta není na vstřikolislu a VZV1 pro ni musí zajet. Aktivity jsou na následujícím obrázku. Kompletní přehled aktivit je na obrázku č. 83.

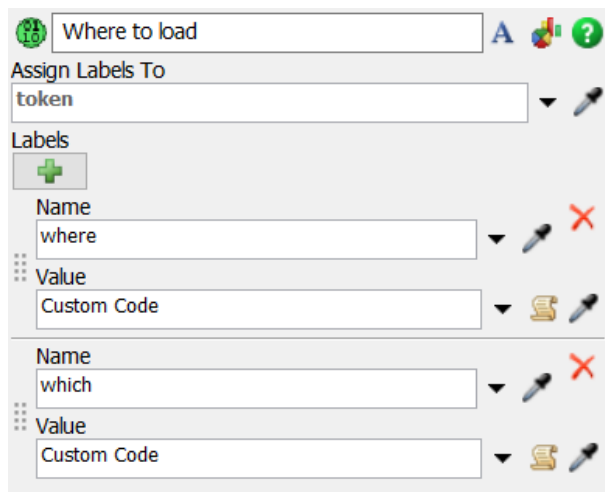


Obrázek 80: Aktivity v bloku rozhodnutí 2 - NE; vlastní zpracování

Přiřazení štítků 3 (Where to load) – aktivita, ve které jsou přiřazeny tokenu dva štítky. Jedná se o where a which, které budou využity v následujících aktivitách. Na obrázku č. 84 je celkové nastavení aktivity. Hodnoty obou štítků jsou řešeny pomocí vlastního kódu.

Štítek where slouží k vyhledání prázdné palety v 3D modelu. Hodnota tohoto štítku je řešena vlastní kódem, který je na obrázku č. 85 včetně komentářů. Štítek je dále využit VZV1, kterému umožní najít požadovanou prázdnou paletu ve zpětném gravitačním regálu.

Štítek which slouží k identifikaci a vrácení konkrétního ID palety, pro kterou VZV1 zajede a naloží. Celý kód štítku včetně komentářů je na obrázku č. 86.



Obrázek 81: Přiřazení štítků 3; vlastní zpracování

```

1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 Variant assignTo = param(4);
6 string labelName = param(5);
7 treenode processFlow = ownerobject(activity);
8
9
10 string pal = token.palletID; // paleta, kterou chci naložit
11
12 Table tab = Table("BackRacks"); // tabulka s racky pro zpetny proud, u kterych vim, ze neco
13 // od zacatku je
14 for(int i = 1; i <= tab.numRows; i++){
15     Object tmp = tab[i][1];
16     string conv = tmp.name;
17     Object pal = Model.find(rack + "/" + pal); // zda je dana paleta v tomto racku
18     if(pal == NULL){
19         continue;
20     }
21     else { // danou paletu jsem jiz nasel ve zpetnem proudu
22         Object zpet = tmp;
23         return tmp;
24     }
25 }
26

```

Obrázek 82: Vlastní kód štítku where; vlastní zpracování

```

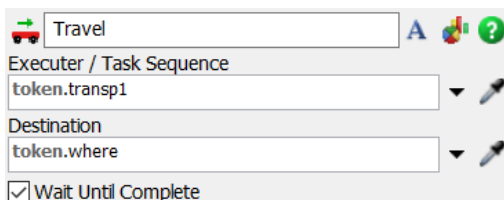
1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 Variant assignTo = param(4);
6 string labelName = param(5);
7 treenode processFlow = ownerobject(activity);
8
9
10 string back = token.where.name; // zpetny proud s paletou
11 string pal = token.palletID; // dana paleta
12
13 Object konkretni = Model.find(back + "/" + pal); // konkretni paleta, pro kterou chci zajet
14 return konkretni;

```

Obrázek 83: Vlastní kód štítku which; vlastní zpracování

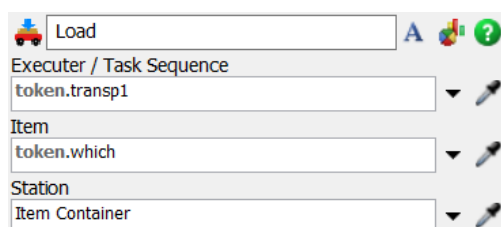
Získání (Acquire) – získání zdroje VZV1, nastavení je identické předchozímu bloku aktivit.

Cesta (Travel) – aktivita, která vybranému zdroji (v tomto případě VZV1) zadá, kam má zajet. Zde je využito štítku where, který udává cíl této aktivity. Celé nastavení je na následujícím obrázku.



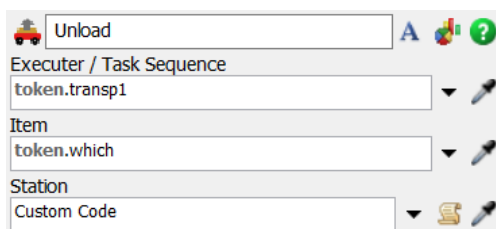
Obrázek 84: Aktivita cesta; vlastní zpracování

Naložení 2 (Load) – aktivita, která zadá VZV1 úkol naložit vybranou paletu ze zpětného regálu na základě štítku which. Nastavení je na následujícím obrázku.



Obrázek 85: Aktivita naložení prázdné palety; vlastní zpracování

Vyložení 2 (Unload) – aktivita, která zadá VZV1 úkol vyložit naloženou paletu na vybraný vstřikolis na základě štítku which. Celkové nastavení aktivity je na následujícím obrázku.



Obrázek 86: Aktivita vyložení prázdné palety; vlastní zpracování

Místo (Station – nastavení v rámci Vyložení 2), kde má být vyložena je vyřešeno pomocí vlastního kódu, který je na následujícím obrázku.

```
1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 treenode executer = param(4); //Or Task Sequence
6 treenode processFlow = ownerobject(activity);
7
8 Object lis = Model.find(token.LisID);
9 return lis; // vracim konkretni objekt lisu
```

Obrázek 87: Vlastní kód pro určení místa vyložení; vlastní zpracování

Vlastní kód (PackageQty increase2) – tento kód řeší zvýšení množství ks výrobků na paletě na základě datové tabulky a příslušného údaje, který určuje, kolik bylo vyrobeno ks. Celý kód je identický aktivitě Vlastní kód (PackageQty increase1).

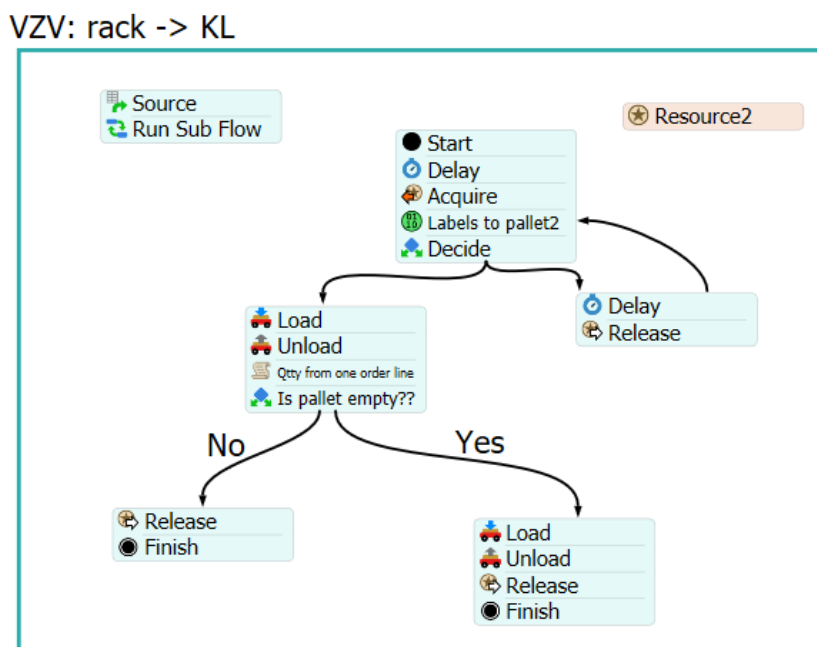
Rozhodnutí 3 (Is Pallet full?) – na základě vlastního kódu dojde k rozhodnutí, zda paleta dosáhla plné kapacity. Celý kód s komentáři je na následujícím obrázku. Nastavení je totožné aktivitě rozhodnutí 2 (Is Pallet full?).

V případě, že kapacita nedosáhla maxima, tak je zdroj VZV1 uvolněn (Release) a token poslán do aktivity dokončení (Finish). Tento proces se v toku opakuje, dokud paleta nemá naplněnou kapacitu. V případě, že je tato kapacita naplněna, je token odeslán druhým konektorem do následujícího bloku aktivit. Tento blok aktivity obsahuje naložení (Load), vyložení (Unload), uvolnění zdroje VZV1 (Release) a dokončení (Finish). Jsou principiálně nastaveny stejně jako v kapitole 8.2.2. Tyto aktivity využívají opět štítků which, vyložení dále využívá vlastního kódu, který vyhledá gravitační regál v modelu.

Tímto je uzavřena část procesního toku, kterému je věnována tato kapitola 8.2.

8.3 VZV2: gravitační regály a kaširovací lisy

Tato část procesního toku řídí VZV2 a logiku oblasti B+C z obrázku č. 57. Tok je rozšířen především o několik rozhodovacích bodů. Celý procesní tok pro tuto logiku je na následujícím obrázku. Jednotlivé aktivity jsou detailněji popsány pod obrázkem.



Obrázek 88: Procesní tok pro řízení VZV2; vlastní zpracování

Následující aktivity jsou principiálně nastaveny jako v kapitole 8.2. Jedná se o aktivitu zdroj (Source), dílčí tok (Run Sub Flow), start, zpoždění (Delay).

Získání (Acquire) – získání zdroje VZV2 nastavení odpovídá předchozím aktivitám získání. V tomto případě se odkazuje na VZV2 místo VZV1.

Přiřazení štítků 2 (Labels to Pallet2) – aktivita, ve které jsou tokenům přiřazeny štítky. K vyhledání hodnot pro všechny štítky v této aktivitě slouží Value Using Global Lookup Table s názvem Výstup. Jedná se o ID gravitačního regálu, vyráběného dílu, ID vozíku a také kaširovacího lisu. Poslední štítek whichPallet pomocí vlastního kódu vyhledá kombinaci palety a gravitačního regálu. Tuto informaci pak dostane VZV2 a v rámci dalších aktivit může

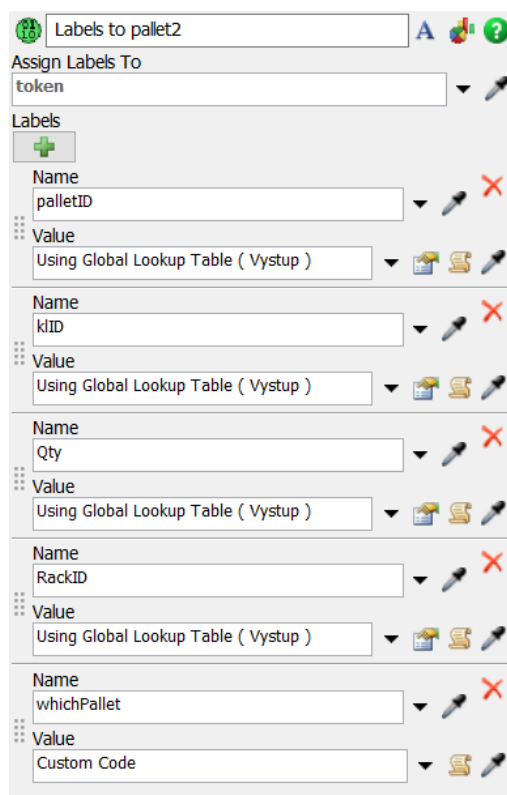
pro paletu zajeť. Vlastní kód pro šítík whichPallet je na obrázku č. 93. Celý seznam šítíků je na obrázku č. 94.

```

1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 Variant assignTo = param(4);
6 string labelName = param(5);
7 treenode processFlow = ownerobject(activity);
8
9 string Paleta = token.palletID;
10 string Rac = token.RackID;
11 Object Pallette = Model.find(Rac+"/"+Paleta);
12 return Pallette;

```

Obrázek 89: Vlastní kód šítíku whichPallet; vlastní zpracování



Obrázek 904: Aktivita přiřazení šítíků 2; vlastní zpracování

Rozhodnutí (Decide) – v této aktivitě dochází k rozhodnutí, jestli je již vybraná naplněná paleta v gravitačním regálu. Toto rozhodnutí je řešeno pomocí jednoduchého vlastního kódu, který je na následujícím obrázku.

```

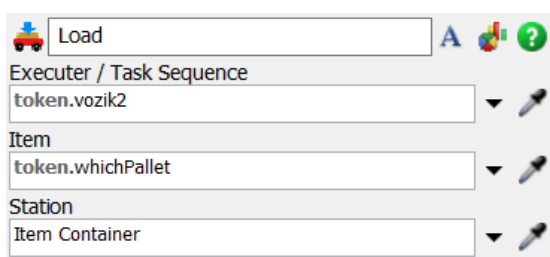
1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 treenode processFlow = ownerobject(activity);
6
7 if(token.whichPallet == NULL){
8     return 1;
9 }
10 else return 2;

```

Obrázek 915: Vlastní kód pro aktivitu decide; vlastní zpracování

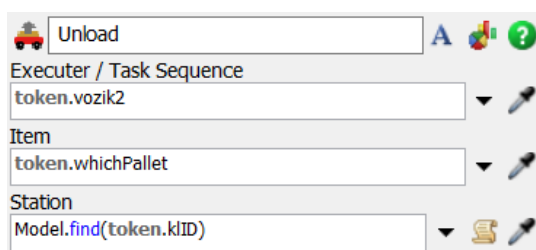
V případě, že paleta ještě není ve vybraném gravitačním regálu, dojde k odeslání tokenu do aktivity zpoždění (Delay). Aktivita zpoždění je zde nastavena na pevnou hodnotu 10 s. Token se poté vrací přes stejný blok aktivit opět do aktivity rozhodnutí 3 (Decide). V této smyčce je do té doby, než je požadovaná paleta založena v regálu a VZV2 tak pro ni může zajet. To řeší následující aktivity.

Naložení 3 (Load) – aktivita, která zadá VZV2 úkol naložit vybranou paletu z gravitačního regálu na základě štítku whichPallet. Nastavení je na následujícím obrázku.



Obrázek 92: Aktivita naložit pro VZV2; vlastní zpracování

Vyložení 3 (Unload) – aktivita, která zadá VZV2 úkol vyložit naloženou paletu k vybranému kaširovacímu lisu na základě příkazu Model.find, který na základě štítku ID kaširovacího lisu vyhledá odpovídající objekt v modelu. Nastavení je na následujícím obrázku.



Obrázek 93: Aktivita vyložit pro VZV2; vlastní zpracování

Vlastní kód (Qtty from one order line) – tento vlastní kód pracuje s daty z tabulky Výstup, kde je na každém řádku udáno, kolik ks výrobků bylo zpracováno. Celý kód včetně komentářů je na následujícím obrázku.

```

1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 treenode processFlow = ownerobject(activity);
6
7
8 Table tab = Table("Vystup");
9 int sum = tab[token.row][3];
10 sum = (-1)*sum; // mnozstvi, ktere chci z palety ponizit
11
12 Object pal = token.whichPallet; // konkretni paleta na ponizeni
13
14 if(pal.packageQtty >= sum){ // zde jsem schopen odecist (zpracovat) celou radku (mnozstvi)
15     pal.packageQtty -= sum;
16 }
17 else { // zde mohu odecist jen cast a zbytek musim z dalsi palety
18     int rozdil = sum - pal.packageQtty; // co mi zbyde
19     pal.packageQtty = 0; // vynulovani
20
21 }

```

Obrázek 94: Vlastní kód pro odečet množství ks; vlastní zpracování

Rozhodnutí (Is pallet empty?) – na základě vlastního kódu dojde k rozhodnutí, zda byla paleta kompletně zpracována (vyprázdněna). Celý kód s komentáři je na následujícím obrázku.

```
1 /**Custom Code*/
2 Object current = param(1);
3 treenode activity = param(2);
4 Token token = param(3);
5 treenode processFlow = ownerobject(activity);
6
7
8 Object pal = token.whichPallet; // paleta, se kterou pracuji
9 if(pal.packageQty == 0){ // paleta je pripravena na odvoz do zpetneho proudu
10     return 1;
11 }
12 else return 2; // paleta na sobe ma jeste nejake mnozstvi
```

Obrázek 95: Vlastní kód rozhodnutí 4; vlastní zpracování

V případě, že množství ks nebylo ještě vynulováno, je tento token poslán do aktivity dokončení a zdroj VZV2 je uvolněn. Tento proces se v toku opakuje, dokud paleta nemá nulové množství výrobků. V případě, že je kapacita vyprázdněna, je token odeslán druhým konektorem do následujícího bloku aktivit, který slouží k založení do zpětných gravitačních regálů. Tento blok aktivit obsahuje naložení (Load), vyložení (Unload), uvolnění zdroje VZV2 (Release) a dokončení (Finish). Tyto aktivity využívají opět štítků whichPallet, vyložení dále využívá příkazu Model.find, který vyhledá zpětný gravitační regál v modelu.

Tímto je uzavřena část procesního toku, kterému je věnována tato kapitola 8.3.

Druhá a finální verze simulačního modelu slouží k rozšíření předchozího modelu první fáze, přiblížení procesů reálnému stavu, přidání zpětného proudu a několika rozhodovacích bodů. Zároveň byly přidány nové datové tabulky pro zpětný proud a ostatní tabulky byly rozšířeny o celý soubor dat. V předchozí fázi bylo využito pouze několika řádků dat pro ověření procesního toku. Rozšířený model pracuje již se všemi daty vyexportovanými pro daný časový interval.

Verifikace a validace

Verifikace a validace simulačního modelu slouží k ověření toho, že model funguje dle očekávání a výstupy ze simulačního modelu odpovídají

9 9.1 Verifikace

Ve spolupráci se zadavatelem proběhla verifikace modelu, během které byly odhaleny problémy s chybějícími daty. Konkrétně se jednalo o umístění prázdných přepravních vozíků ve zpětných gravitačních regálech. Chyběla informace o přesné pozici těchto přepravních vozíků. Na základě tohoto zjištění byla část zpětného proudu zjednodušena a přepravním vozíkům byly pevně přiřazeny zpětné regály. Výběr těchto regálů byl na základě daného projektu, pod který vozík spadal. Jiné výraznější problémy napovrch nevyvstaly jelikož byl simulační model budován po jednotlivých fázích.

9.2 Validace

Validace probíhala spouštěním simulačních běhů a ověření každé části procesního toku a tokenů po jednotlivých krocích. Na základě tohoto simulačního běhu bylo kontrolováno, zda model odpovídá realitě. V části procesního toku pro přepravu vozíků z regálů na kaširovací lisy, byla objevena nesrovnalost s vlastním kódem, který hledá požadovaný vozík na odpovídajícím gravitačním regálu. V případě, že tento vozík nebyl nalezen na první pokus, došlo k zablokování tokenů. Tento problém byl vyřešen pomocí správného uvolnění zdroje a odblokování tokenů.

Návrh klíčových indikátorů výkonnosti a jejich analýza

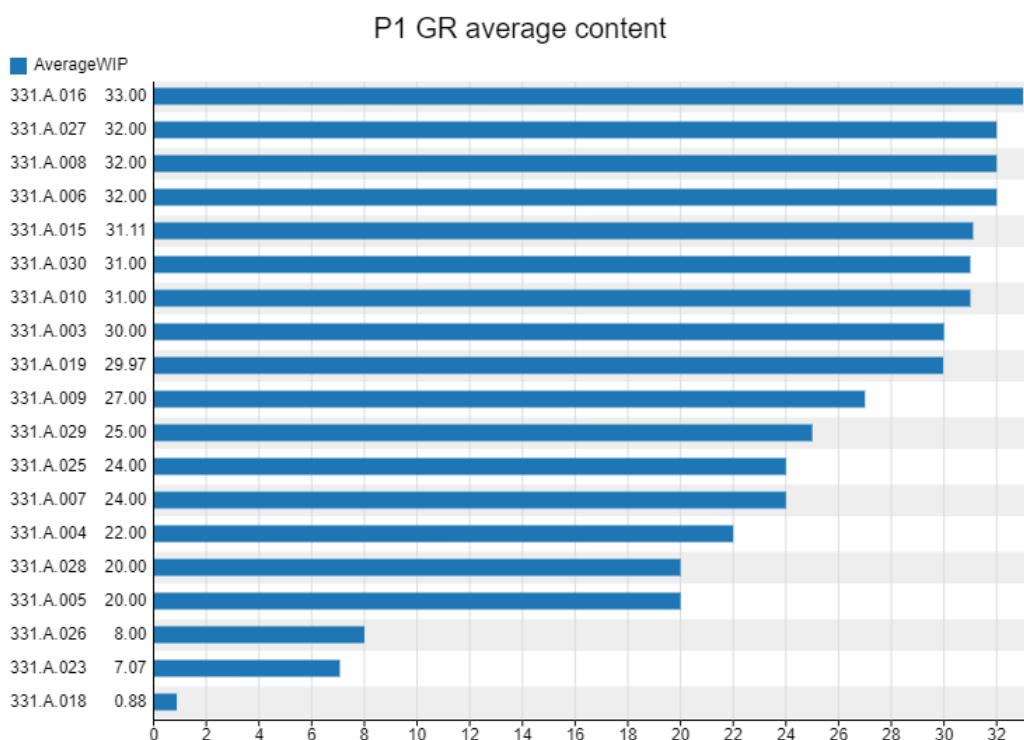
Cílem třetí a zároveň poslední fáze je návrh klíčových indikátorů výkonnosti, které je třeba sledovat v návaznosti na charakteristiku řešeného problému. Proběhne jejich vyhodnocení a doporučení za využití provozních dat. Simulační model včetně indikátorů lze využít i pro další provozní data např. jiné časové období, projekty apod. logistického systému za využití simulačního modelu a celého souboru dat. Jedná se o indikátor vytíženosti gravitačních regálů, indikátor vytíženosti zpětných regálů, počet naplněných přepravních vozíků vystupujících ze vstřikolisů a potřebné množství přepravních vozíků pro tento běh.

10.1.1 Vytíženost gravitačních regálů

Cílem tohoto indikátoru je zobrazení průměrného zaplnění gravitačních regálů v rámci každého projektu. Tento indikátor je důležitý, protože toto vytížení je pro jeden pracovní den (3 směny), proto se může lišit v rámci jiných dlouhodobých projektů a zároveň i jiných pracovních dnů. Každý gravitační regál má maximální kapacitu 33 přepravních vozíků.

Vytíženost gravitačních regálů pro projekt 1

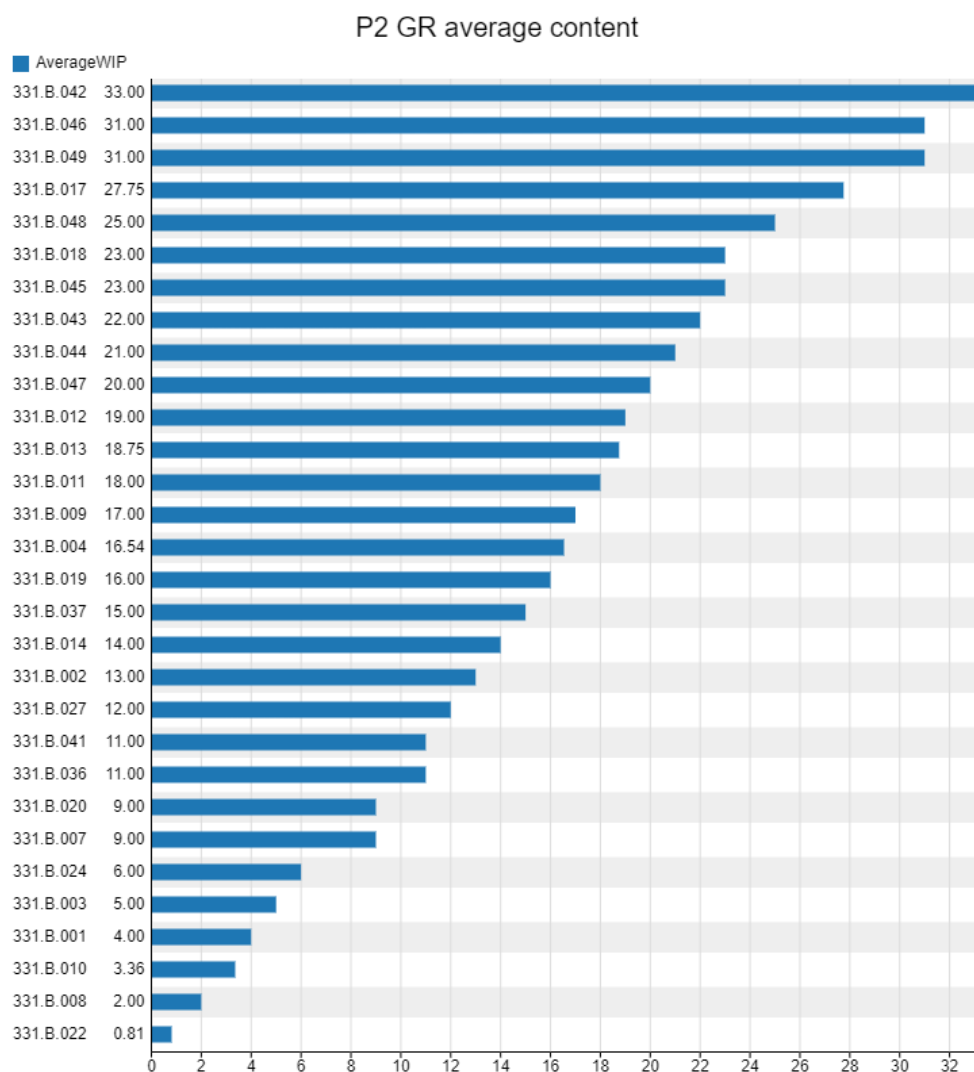
Projekt 1 má vyhrazených 33 gravitačních regálů. Z toho zbývá ještě 14 regálů kompletně volných. Když nebereme v potaz pravidlo, které určí, že do stejné úrovně gravitačního regálu mohou pouze přepravní vozíky se stejným typem výrobku, tak v rámci tohoto projektu to znamená prostor pro dalších 462 přepravních vozíků. V následujícím grafu je průměrné zaplnění využitých gravitačních regálů projektu 1. Regál 331.A.016 je zaplněn po celou dobu simulačního běhu. Regálů, které dosahují průměrného zaplnění 30–32 vozíky, je celkem 7. Gravitačních regálů s průměrným zaplněním 20–30 je 7. Zbývající využitá regály jsou celkem 3 a dosahují pouze jednotek ks.



Graf 1: Průměrné vytížení GR pro projekt 1; vlastní zpracování

Vytíženost gravitačních regálů pro projekt 2

Projekt 2 má vyhrazených 40 gravitačních regálů. Z toho zbývá ještě 10 regálů kompletně volných, což rámci tohoto projektu znamená prostor pro dalších 330 přepravních vozíků. V následujícím grafu je průměrné zaplnění využitých gravitačních regálů projektu 2.

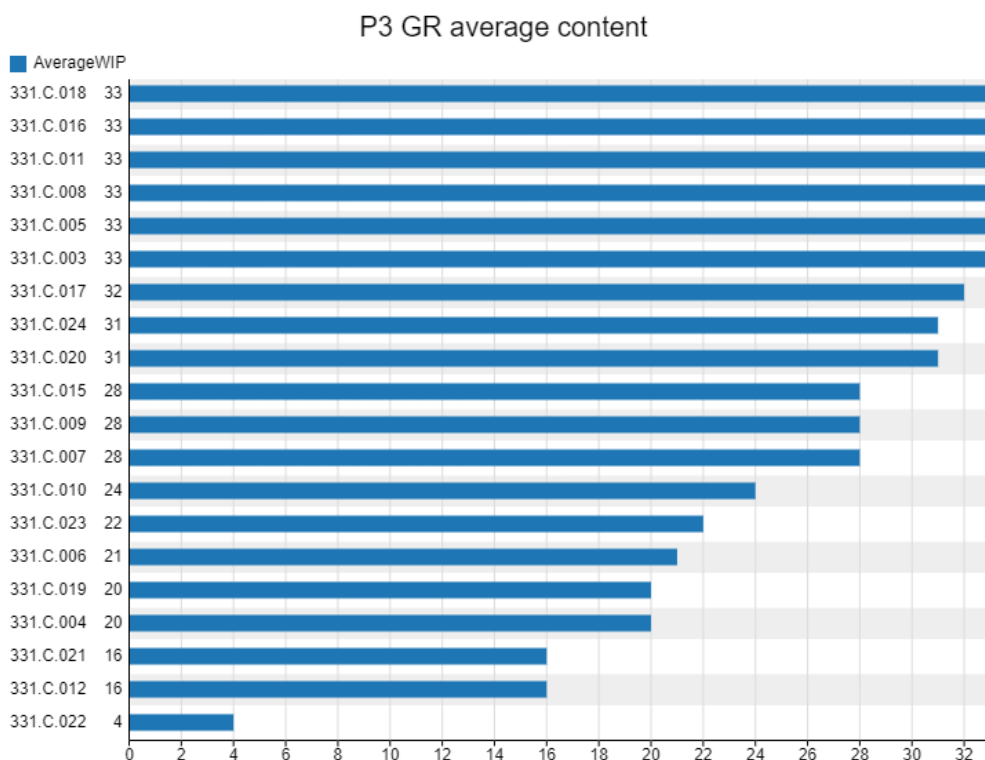


Graf 2: Průměrné vytížení GR pro projekt 2; vlastní zpracování

Regál 331.B.042 je zaplněn po celou dobu simulačního běhu. Regály, které dosahují průměrného zaplnění 30 – 32 vozíky, jsou 2. Regálů s průměrným zaplněním 20 – 30 je 7. Regálů s průměrným zaplněním 10 – 20 vozíků je 12. Zbýlých 8 regálů dosahuje pouze jednotek ks.

Vytíženost gravitačních regálů pro projekt 3

Projekt 3 má vyhrazených 22 gravitačních regálů. Z toho zbývají ještě 2 regály kompletně prázdné. Když nebereme v potaz pravidlo, které určí že do stejné úrovně gravitačního regálu mohou pouze přepravní vozíky se stejným typem výrobku, tak v rámci tohoto projektu to znamená prostor pro dalších 66 přepravních vozíků. V následujícím grafu je průměrné zaplnění využitých gravitačních regálů projektu 3.



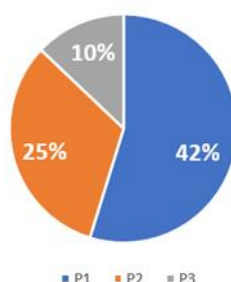
Graf 3: Průměrné vytížení GR pro projekt 3; vlastní zpracování

Regálů, které jsou zaplněny po celou dobu simulačního běhu, je 6. Regály, které dosahují průměrného zaplnění 30 – 32 vozíky, jsou 3. Regálů s průměrným zaplněním 20 – 30 je 8. Regály s průměrným zaplněním 10 – 20 vozíků jsou 2. Zbýlý regál dosahuje pouze jednotek ks.

Doporučení a zhodnocení

Nejvíce zaplněným místem je tedy oblast pro projekt 3 a naopak nejvíce volné kapacity má projekt 1. Porovnání celkového a volného počtu gravitačních regálů je na následujícím grafu.

% množství volných regálů dle projektu



Graf 4: Procento volných GR dle projektu; vlastní zpracování

V případě většího příchozího objemu z výroby by mohl projekt 3 rychle dosáhnout plné kapacity. Zbývají volné pouze 2 gravitační regály, to znamená 6 proudů a maximálně 6 různých typů výrobku. Na první pohled nejjednodušším řešením je ubrat gravitační regály z projektu 1 a přidat je projektu 3, problémem je však to, že jsou na opačných koncích haly. Jelikož projekt 1 má téměř polovinu regálů volných, tak by řešením mohlo být přeskládání zón a s tím souvisejícího umístování přepravních vozíků.

10.1.2 Vytíženost zpětných regálů

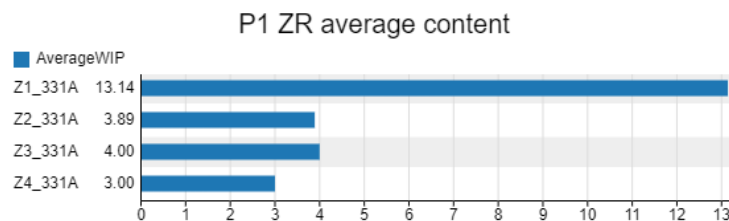
Cílem tohoto indikátoru je zobrazení průměrného vytížení zpětných gravitačních regálů v rámci každého projektu. Tento indikátor je důležitý, protože v případě zaplnění těchto zpětných proudů dojde ke zpomalení celého toku a případnému odkládání vozíků mimo určené oblasti. Toto vytížení je pro jeden pracovní den (3 směny), proto se může lišit v rámci jiných dlouhodobých projektů a zároveň i jiných pracovních dnů. Každý gravitační regál má maximální kapacitu 33 přepravních vozíků. Z těchto zpětných regálů odebírají vysokozdvizné vozíky prázdné vozíky a odvázejí je zpět ke vstřikolisům, kde jsou opět plněny výrobky.

Tato část byla v simulačním modelu zjednodušena, protože nikde v databázi nejsou pevně dané údaje o pozicích prázdných přepravních vozíků a jim odpovídajících regálů. Pomocí filtrování a úpravě v excelu byl získán seznam prázdných vozíků, které jsou již založeny, ale nešlo již dohledat na jakém zpětném regálu. Proto byly tyto vozíky (v 3D modelu palety)

rozmístěny na základě nejbližších zpětných regálů. To bylo umožněno především díky ID vstříkolisu a ID gravitačního regálu pro založení zaplněného vozíku, poté byl již vybrán nejbližší zpětný regál. Celkem je v hale 10 zpětných gravitačních regálů. Vzhledem k velikosti projektů jsou vyhrazeny 4 zpětné regály pro projekt 1 a 2. Pro projekt 3 jsou vyhrazeny 2 zpětné regály.

Vytíženost zpětných regálů pro projekt 1

Celková kapacita zpětných regálů projektu 1 je 12 proudů, tedy 132 přepravních vozíků. Na následujícím grafu je přehled průměrného zatížení těchto zpětných regálů těchto zpětných regálů.

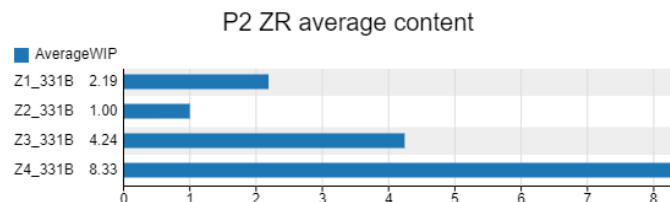


Graf 5: Průměrné vytížení pro projekt 1; vlastní zpracování

Nejvyššího zaplnění dosáhl Z1_331A, a to celkem 13 vozíků, přesto nedosáhl ani 50% celkové kapacity. Ostatní zpětné regály dosáhly pouze jednotek ks.

Vytíženost zpětných regálů pro projekt 2

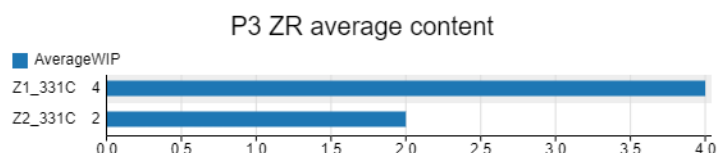
Celková kapacita projektu 2 je také 12 proudů, tedy 132 přepravních vozíků. Na následujícím grafu je přehled průměrného zatížení těchto zpětných regálů. Nejvyššího zaplnění dosáhl regál Z4_331B, a to celkem 8 vozíků, ale i tak nepřesáhl 25% své celkové kapacity. Zbylé zpětné regály dosáhly pouze jednotek ks.



Graf 6: Průměrné vytížení pro projekt 2; vlastní zpracování

Vytíženost zpětných regálů pro projekt 3

Celková kapacita projektu 3 je rovna 6 proudům, tedy 66 přepravním vozíkům. Na následujícím grafu je přehled průměrného zatížení těchto zpětných regálů. Nejvyššího zaplnění dosáhl regál Z1_331C, ale ve výsledku to byly pouze 4 palety.



Graf 7: Průměrné vytížení pro projekt 3; vlastní zpracování

Doporučení a zhodnocení

Na základě simulačního běhu nedošlo k přetížení zpětného proudu. Pro tato vstupní data nebyla kapacita problém. Přesný stav zpětných proudů nelze přesně určit z důvodu nepřesných vstupních údajů. Doporučením je dodržování štítků u prázdných vozíků a netisknout nové v případě, že požadovaný vozík není hned k nalezení. Další hodnotnou informací by mohla být evidence prázdných vozíků ve zpětných prouděch. V současném stavu není přesná informace o konkrétním vozíku na konkrétním zpětném proudu, s těmito informacemi bude výstup simulačního modelu mnohem přesnější.

10.1.3 Výstup ze vstříkolisů

Tento indikátor slouží k analýze zpracovaných signálů na vstříkolisech. V tomto simulačním běhu a na základě vstupního signálu bylo zpracováno přes 200 požadavků. Přesněji řečeno, byly plněny vyráběnými díly.



Graf 8: Výstup ze vstříkolisů; vlastní zpracování

V rámci tohoto simulačního běhu (24 hodin) bylo využíváno pouze 9 vstřikolisů z celkových 30. Celkem bylo zpracováno 202 signálů přepravních vozíků. Celkový počet zpracovaných vstupních signálů na vstřikolisech je na grafu č. 8.

Nejvíce zpracovaných signálů bylo na následujících lisech: lis IMM16 – 50 signálů , lis IMM20 – 48 signálů, lis IMM13 – 46 signálů. Naopak převážná většina lisů nebyla v tomto simulačním běhu využita.

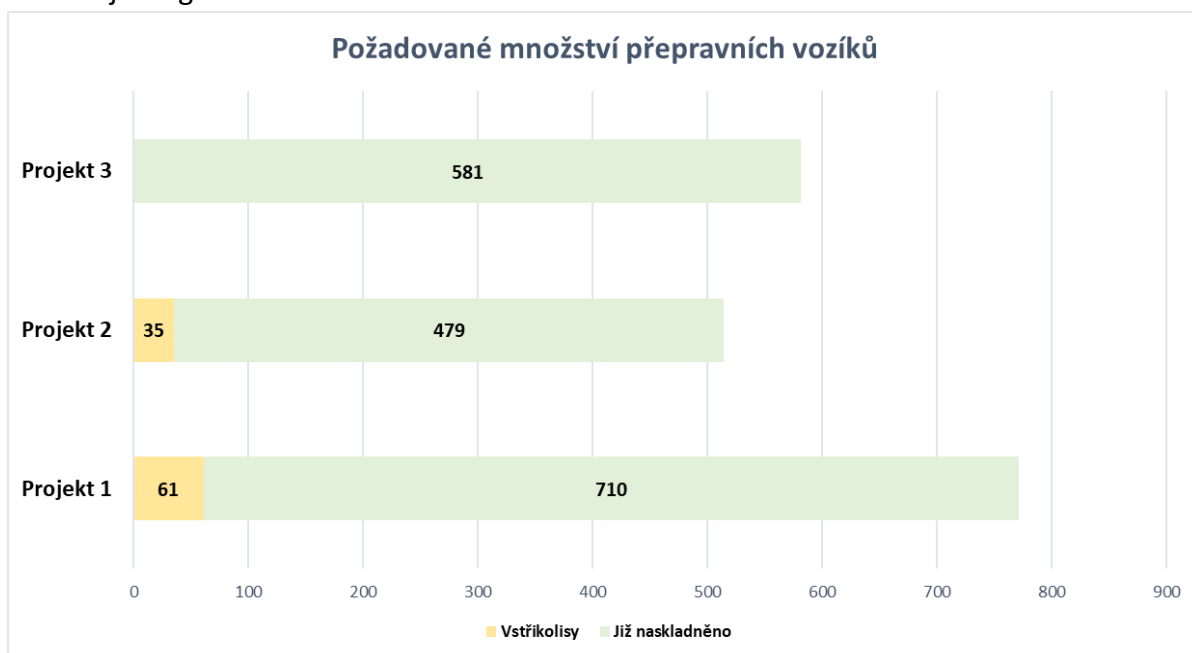
Doporučení a zhodnocení

V rámci tohoto simulačního běhu je využíváno pouze zlomku kapacity vstřikolisů, týká se to projektů 1 a 2, pro projekt 3 nebyl zpracován žádný signál. Důvodů může být několik. Součástí souboru dat nejsou projekty využívající zbylé vstřikolisy nebo bylo v tento den vyráběno pouze pro projekty 1 až 2. Může zde být i omezení z hlediska dostupných forem, kdy nelze vyrábět stejné varianty požadovaných dílů na zmiňovaných nevyužitých vstřikolisech.

10.1.4 Využité přepravní vozíky

Tento indikátor slouží k analýze požadovaného množství přepravních vozíků pro aktuální projekty. Jedná se o kombinaci vstupního signálu z pohledu využitých vozíků, čímž se liší od předchozího indikátoru, který pracoval se signály, a již naskladněných vozíků (kapitola 7.1).

V projektech 1 až 3 je vyžadováno 1866 přepravních vozíků. V rámci výstupu ze vstřikolisů bylo potřeba 96 přepravních vozíků a k tomu bylo již naskladněno 1770. Přehled je na následujícím grafu.



Graf 9: Množství požadovaných přepravních vozíků; vlastní zpracování

Doporučení a zhodnocení

Ze zpracovaných dat vychází požadavek na 1866 přepravních vozíků v rámci všech tří projektů – vstupní signály ze vstřikolisů a již naskladněné vozíky. Pro projekt 3 nebyly zpracovány žádné vstupní signály, pouze bylo naskladněno 581 vozíků. Pro projekt 2 bylo zpracováno 35 vozíků a již naskladněno 479. Pro projekt 1 bylo zpracováno 61 vozíků a již naskladněno 710. V rámci doporučení z kapitoly 10.1.2 a to konkrétně přesná evidence vozíků ve zpětných proudech může mít vliv i na tuto oblast. Tím, že se zefektivní zpětný proud prázdných vozíků, dojde k plynulejšímu přísunu požadovaných vozíků ke vstřikolisům. S efektivnějším tokem mohou být využívány prázdné vozíky přesně v požadovaný čas a tím ubude množství vozíků, které nejsou využívány na 100%.

Závěr

Kvůli prudkému technologickému růstu a globální konkurenci je stále větší tlak na efektivitu a snižování nákladů. Nejenom u výrobních podniků dochází k analýze pomocí simulačních softwarů. V kombinaci s již zavedenými softwary pro plánování a skladování je možné získat velmi cenné informace pro vstup analýzy. Tyto cenné vstupy z plánovacích softwarů je možné transformovat na výstupy ze simulačního modelu, které nám dají možnost pohlížet na systém z různých úhlů. Na základě historických dat lze vytvářet pomocí simulačního běhu prognózy, zkoušet změny a celkově hledat optimální řešení pro mnoho problémů.

Cílem této práce bylo vytvoření takového modelu pro skladovací systém podniku z automobilového průmyslu a jeho ověření pomocí souboru provozních dat. Data musela být nejprve transformována, jelikož se jedná o export přímo z databáze Asprovy z několika tabulek. Na základě charakteristiky problému byl postaven koncepční model. Tento model byl dále přenesen do simulačního softwaru FlexSim, kde musela být kompletně vytvořena logika pomocí procesního toku. Tento procesní tok byl rozdělen na několik částí – počáteční naskladnění, výroba na vstřikolisech, manipulace s vozíky od vstřikolisů do regálů, manipulace s vozíky z regálů do kaširovacích lisů. Součástí tohoto prvotního modelu bylo i ověření funkcionality na úzkém množství dat. Tento model byl v další fázi rozšířen o zpětný proud, který obsahuje prázdné přepravní vozíky, které se vrací zpět do výroby. Dalšími kroky bylo upřesnění části logiky z první fáze modelu. Soubor dat, se kterým se v tomto rozšířeném modelu pracovalo, byl už kompletní.

Pro finální verzi modelu a na základě ostrých dat v tomto modelu obsažených byly nadefinovány indikátory výkonnosti a byla provedena jejich analýza včetně následného doporučení. Tyto indikátory slouží jako výstup simulačního běhu. Mezi definované indikátory patří vytížení gravitačních regálů, vytížení zpětného proudu, zpracované požadavky na vstřikolisech a využití přepravní vozíky pro současné projekty. Z analýzy vytíženosti gravitačních regálů vyšlo, že projekt 1 má ještě 14 prázdných regálů, projekt 2 má ještě 10 prázdných regálů a projekt 3 má ještě 2 prázdné regály. Tento indikátor slouží k přehledu vytížení regálů napříč projekty. Může sloužit jako podnět k přiřazení gravitačních regálů k projektu, který je nejvytíženější a naopak ubrání z projektu, kde jsou rezervy zbytečně velké. Dalším analyzovaným indikátorem je požadavek na přepravní vozíky

v rámci projektů. Zde je analyzován požadavek na přepravní vozíky. Tento požadavek vychází ze vstupních signálů na vstřikolisech, kdy dochází k plnění těchto vozíků výrobky a zároveň vozíky, které jsou již naskladněny. Pro projekt 1 se jedná celkem o 771 vozíků, pro projekt 2 se jedná o 514 vozíků a pro projekt 3 se jedná o 581 vozíků. Celkem je tedy vyžadováno 1866 přepravních vozíků.

Vytvořený model bude nyní možné rozšířit o další stupeň výroby a s tím spojené výrobní logistiky a tím získat možnost dalších výstupů simulace a analýz. Tento model bude tedy jednak dále rozšiřován a zpřesňován za účelem získání co nejpřesnějšího simulačního modelu a jednak bude využíván pro řízení a vyhodnocování současného provozu skladu.

Seznam obrázků

Obrázek 1: Klíčové prvky logistiky a distribuce [1].....	13
Obrázek 2: Klíčové aktivity logistiky [31].....	14
Obrázek 3: Základní transformační diagram vstupů a výstupů; [1]	15
Obrázek 4: Sankeyův diagram [33].....	18
Obrázek 5: Spaghetti diagram [32]	18
Obrázek 6: Vývojový diagram [34]	19
Obrázek 7: Value stream mapping [35].....	20
Obrázek 8: Policový regál; [4].....	25
Obrázek 9: Paletový regál; [4]	26
Obrázek 10: Průjezdový regál; [4]	27
Obrázek 11: Spádový regál; [4]	27
Obrázek 12: Typy skladových rozvržení; [6]	29
Obrázek 13: Typy směrování; [6]	31
Obrázek 14: Princip počítačové simulace [9]; vlastní zpracování	34
Obrázek 15: Chování diskretních simulací [14]	36
Obrázek 16: Model výrobní linky [18]	39
Obrázek 17: Řešení logistiky skladu [22].....	40
Obrázek 18: Pokrytí oblasti mobilní sítě [23]	41
Obrázek 19: Řízení software projektu [25]	41
Obrázek 20: Simulace pacientů na pohotovosti [23]	42
Obrázek 21: Životní cyklus simulačního projektu [27]	44
Obrázek 22: Uživatelské rozhraní 3D modelu [29].....	47
Obrázek 23: Uživatelské rozhraní procesního toku [29]	48
Obrázek 24: Logika průběhu procesů skladu [22].....	49
Obrázek 25: Token [29]	50
Obrázek 26: Aktivita zdroj [29].....	51
Obrázek 27: Sdílené zdroje [29]	51
Obrázek 28: Konektor [29]	52
Obrázek 29: Složený blok [29].....	53
Obrázek 30: Zjednodušené schéma toku výroby; vlastní zpracování	55
Obrázek 31: Nástroje a jejich využití; vlastní zpracování	55
Obrázek 32: Datové tabulky; vlastní zpracování	56
Obrázek 33: 3D model 1. fáze; vlastní zpracování	57
Obrázek 34: Procesní tok počátečního naskladnění; vlastní zpracování	58
Obrázek 35: Dílčí tok naskladnění; vlastní zpracování	59
Obrázek 36: Nastavení štítků pro tokeny; vlastní zpracování	60
Obrázek 38: Nastavení štítku pro umístění; vlastní zpracování	Chyba! Záložka není definována.
Obrázek 37: Kód pro umístění palet; vlastní zpracování.....	Chyba! Záložka není definována.
Obrázek 39: Generování objektů; vlastní zpracování	62
Obrázek 40: Kód pro pojmenování objektů; vlastní zpracování	62
Obrázek 41: Procesní tok palet z výroby; vlastní zpracování	63
Obrázek 42: Kód aktivity zpoždění; vlastní zpracování	64

Obrázek 43: Přiřazení štítků pro vozíky z výroby; vlastní zpracování.....	65
Obrázek 44: Vlastní kód přiřazení ID vstřikolisů; vlastní zpracování.....	65
Obrázek 45: Vlastní kód generování počtu vozíků; vlastní zpracování	65
Obrázek 46: Generování objektů z výroby; vlastní zpracování	66
Obrázek 47: Procesní tok odvozu palet z výroby; vlastní zpracování	67
Obrázek 48: Aktivita zdroj; vlastní zpracování	67
Obrázek 49: Aktivita získání zdroje; vlastní zpracování.....	68
Obrázek 50: Zdroj VZV; vlastní zpracování.....	68
Obrázek 51: Přiřazení štítku tokenům; vlastní zpracování.....	69
Obrázek 52: Aktivita vyložení; vlastní zpracování.....	70
Obrázek 53: Aktivita naložení; vlastní zpracování.....	69
Obrázek 54: Procesní tok pro odvoz palet z regálů; vlastní zpracování.....	71
Obrázek 55: Vlastní kód vyhledání vozíku v regálu; vlastní zpracování	71
Obrázek 56: Rozložení regálů na projekty; vlastní zpracování.....	74
Obrázek 57: Rozložení haly; vlastní zpracování	Chyba! Záložka není definována.
Obrázek 58: 3D model haly – pohled od kaširovacích lisů; vlastní zpracování	74
Obrázek 59: Procesní tok 2. fáze simulačního modelu; vlastní zpracování	73
Obrázek 60: Procesní tok pro generování prázdných vozíků; vlastní zpracování	75
Obrázek 61: Nastavení dílčího toku 1; vlastní zpracování.....	75
Obrázek 62: Nastavení štítků pro vozíky - zpětný regál; vlastní zpracování	76
Obrázek 63: Změna barvy 1; vlastní zpracování.....	77
Obrázek 64: Generování palet ve zpětných regálech; vlastní zpracování.....	77
Obrázek 65: Vlastní kód pro pojmenování palet 1; vlastní zpracování	78
Obrázek 66: Nastavení dílčího toku 2; vlastní zpracování.....	78
Obrázek 67: Vlastní kód pro štítek where; vlastní zpracování	79
Obrázek 68: Nastavení štítků pro vozíky - VL; vlastní zpracování	79
Obrázek 69: Generování palet u vstřikolisů; vlastní zpracování	80
Obrázek 70: Změna barvy 2; vlastní zpracování.....	80
Obrázek 71: Vlastní kód pro pojmenování palet 2; vlastní zpracování	80
Obrázek 72: Procesní tok pro řízení VZV1; vlastní zpracování	81
Obrázek 73: Vlastní kód rozhodnutí 1; vlastní zpracování.....	82
Obrázek 74: Aktivity v bloku rozhodnutí 1 - ANO; vlastní zpracování.....	82
Obrázek 75: Vlastní kód pro navýšení množství na paletě; vlastní zpracování.....	83
Obrázek 76: Vlastní kód pro kontrolu kapacity palety; vlastní zpracování	83
Obrázek 77: Aktivita získání zdroje VZV1; vlastní zpracování	84
Obrázek 78: Vlastní kód štítku unload; vlastní zpracování.....	85
Obrázek 79: Vlastní kód štítku load; vlastní zpracování.....	84
Obrázek 80: Přiřazení štítků load a unload; vlastní zpracování	84
Obrázek 81: Aktivita naložit pro VZV1; vlastní zpracování.....	85
Obrázek 82: Aktivita vyložit pro VZV1; vlastní zpracování	85
Obrázek 83: Aktivity v bloku rozhodnutí 2 - NE; vlastní zpracování.....	86
Obrázek 84: Přiřazení štítků 3; vlastní zpracování	87
Obrázek 85: Vlastní kód štítku where; vlastní zpracování	87
Obrázek 86: Vlastní kód štítku which; vlastní zpracování	87
Obrázek 87: Aktivita cesta; vlastní zpracování.....	88

Obrázek 88: Aktivita naložení prázdné palety; vlastní zpracování.....	88
Obrázek 89: Aktivita vyložení prázdné palety; vlastní zpracování	88
Obrázek 90: Vlastní kód pro určení místa vyložení; vlastní zpracování	88
Obrázek 91: Procesní tok pro řízení VZV2; vlastní zpracování	90
Obrázek 92: Vlastní kód štítku whichPallet; vlastní zpracování	91
Obrázek 93: Vlastní kód pro aktivitu decide; vlastní zpracování	91
Obrázek 94: Aktivita přiřazení štítků 2; vlastní zpracování	91
Obrázek 95: Aktivita naložit pro VZV2; vlastní zpracování.....	92
Obrázek 96: Aktivita vyložit pro VZV2; vlastní zpracování	92
Obrázek 97: Vlastní kód pro odečet množství ks; vlastní zpracování	92
Obrázek 98: Vlastní kód rozhodnutí 4; vlastní zpracování	93

Seznam grafů

Graf 1: Průměrné vytížení GR pro projekt 1; vlastní zpracování.....	95
Graf 2: Průměrné vytížení GR pro projekt 2; vlastní zpracování.....	96
Graf 3: Průměrné vytížení GR pro projekt 3; vlastní zpracování.....	97
Graf 4: Procento volných GR dle projektu; vlastní zpracování.....	98
Graf 5: Průměrné vytížení pro projekt 1; vlastní zpracování	99
Graf 6: Průměrné vytížení pro projekt 2; vlastní zpracování	99
Graf 7: Průměrné vytížení pro projekt 3; vlastní zpracování	100
Graf 8: Výstup ze vstřikolisů; vlastní zpracování.....	100
Graf 9: Množství požadovaných přepravních vozíků; vlastní zpracování	102

Seznam tabulek

Tabulka 1: Simulace v různých fázích životního cyklu výrobního a logistického systému; [7].....	33
Tabulka 2: Historie simulačních programů [9]; vlastní zpracování	34
Tabulka 3: InitialLoading	59
Tabulka 4: Vstup; vlastní zpracování	64
Tabulka 5: Výstup	72
Tabulka 6: BackFlow; vlastní zpracování.....	76
Tabulka 7: InitialLis; vlastní zpracování.....	78

Seznam použité literatury

- [1] RUSHTON, Alan, Phil CROUCHER a Peter BAKER. The Handbook of Logistics & Distribution Management [online]. 5. London, UK: Kogan Page Limited, 2014 . ISBN 978-0-7494-6628-2.
- [2] JUROVÁ, Marie. Výrobní a logistické procesy v podnikání [online]. Praha: Grada Publishing, 2016. Expert (Grada). ISBN 978-80-271-9330-1.
- [3] GROS, Ivan. Velká kniha logistiky. Praha: Vysoká škola chemicko-technologická v Praze, 2016. ISBN 978-80-7080-952-5.
- [4] *Jungheinrich: Regálové systémy* [online]. Dostupné z: <https://www.jungheinrich.cz/produkty/regaly>
- [5] RICHARDS, Gwynne. Warehouse Management: A complete guide to improving efficiency and minimizing costs in the modern warehouse [online]. 3. London, UK: Kogan Page Limited, 2018. ISBN 978-0-7494-7978-7.
- [6] An overview of warehouse optimization. *International journal of advances in telecommunications, elektrotechnics, signals and systems*. 2013, 2(3), 111-117. Dostupné z: doi:<http://dx.doi.org/10.11601/ijates.v2i3.61>
- [7] MUJICA MOTA, Miguel a Idalia FLORES DE LA MOTA, ed. Applied Simulation and Optimization 2: New Applications in Logistics, Industrial and Aeronautical Practice [online]. Switzerland: Springer International Publishing, 2017. ISBN 978-3-319-55810-3.
- [8] NANCE, Richard E. a C. Michael OVERSTREET. History of computer simulation software: An initial perspective. In: *2017 Winter Simulation Conference (WSC)* [online]. IEEE, 2017, 2017, s. 243-261 [cit. 2019-11-30]. DOI: 10.1109/WSC.2017.8247792. ISBN 978-1-5386-3428-8. Dostupné z: <http://ieeexplore.ieee.org/document/8247792/>
- [9] MANLIG, František. *Využití počítačové simulace výrobních systémů*. Liberec: Technická univerzita v Liberci, 2014. Knihovnička Katedry výrobních systémů TU v Liberci. ISBN 978-80-7494-162-7.
- [10] Schriber T J, 1987, "The Nature and Role of Simulation in the Design of Manufacturing Systems". *Simulation in CIM and Artificial Intelligence Techniques*, Ed. Retti J & Wichmann K E, Society of Computer Simulation, pp. 5- 18
- [11] Altiok, Tayfur, and Benjamin Melamed. *Simulation Modeling and Analysis with Arena*, Elsevier Science & Technology, 2007. ProQuest Ebook Central, <https://ebookcentral.proquest.com/lib/cvut/detail.action?docID=305656>.

- [12] Rossetti, Manuel D.. *Simulation Modeling and Arena*, John Wiley & Sons, Incorporated, 2015. ProQuest Ebook Central, <https://ebookcentral.proquest.com/lib/cvut/detail.action?docID=1895506>.
- [13] HOSSEINPOUR, F. a H. HAJIHOSSEINI. Importance of Simulation in Manufacturing. *World Academy of Science, Engineering and Technology International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering Vol:3, No:3, 2009* [online]. 2009, **3**(2009), 229-231. Dostupné z: <https://pdfs.semanticscholar.org/26e7/0f93e08f15b5b47af7500a994ef99dbdd562.pdf>
- [14] *Behavior of a discrete system simulation* [online]. Dostupné z: https://www.tutorialspoint.com/modelling_and_simulation/modelling_and_simulation_discrete_system_simulation.htm
- [15] SHARMA, Prateek. Discrete-Event Simulation. *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH* [online]. 2015, **4**(4), 1-5. Dostupné z: <https://www.ijstr.org/final-print/apr2015/Discrete-event-Simulation.pdf>
- [16] Price, A. (2014). Discrete-event haulage simulation making better decisions with reduced uncertainty. Runge Pincock Minarco, Perspectives Newsletter, Issue No. 123, April 2014, 1-3.
- [17] Upadhyay, Askari-Nasab, Tabesh & Badiozamani (2015) – Simulation & Optimization in Open Pit Mining. In Bandopadhyay (Ed.): Application of computers and operations in research in the mining industry. Englewood, Colo., Society for Mining Metallurgy & Exploration Inc. (SME).
- [18] Kampa, Adrian & Gołda, Grzegorz & Paprocka, Iwona. (2017). Discrete Event Simulation Method as a Tool for Improvement of Manufacturing Systems. *Computers*. 6. 10. 10.3390/computers6010010.
- [19] SUN B, G.Q. a T.A. SPEDDING. Application of discrete event simulation to the activity based costing of manufacturing systems. *International Journal of Production Economics*. 1999, Issue 3(Volume 58), 289-301. DOI: [https://doi.org/10.1016/S0925-5273\(98\)00204-7](https://doi.org/10.1016/S0925-5273(98)00204-7).
- [20] ULRICH, Ph.D, Doc. Ing. Zdeněk a Ing. Pavel RAŠKA, Ph.D. Simulace výrobních linek. In: Systemonline [online]. Dostupné z: <https://www.systemonline.cz/rizeni-vyroby/simulace-vyrobnych-linek.htm>
- [21] ULRICH, Ph.D, Doc. Ing. Zdeněk a Ing. Pavel RAŠKA, Ph.D. Optimalizace logistických procesů a materiálových toků. In: Systemonline [online]. Dostupné z: <https://www.systemonline.cz/it-pro-logistiku/optimalizace-logistickyh-procesu-a-materialovych-toku.htm>

- [22] Making Critical Warehouse Decisions. *FlexSim* [online]. Dostupné z: <https://www.flexsim.com/case-studies/critical-warehouse-decisions/>
- [23] ZOLA, Enrica, Israel MARTIN-ESCALONA a Francisco BARCELO-ARROYO. Discrete Event Simulation of Wireless Cellular Networks. GOTI, Aitor, ed. Discrete Event Simulations [online]. Sciyo, 2010, 2010-08-18. DOI: 10.5772/9903. ISBN 978-953-307-115-2. Dostupné z: <http://www.intechopen.com/books/discrete-event-simulations/discrete-event-simulation-of-wireless-cellular-networks>
- [24] Better Decisions in the Emergency Department [online]. Dostupné z: <https://healthcare.flexsim.com/case-studies/better-decisions-emergency-department/>
- [25] Kouskouras, Konstantinos & Georgiou, Andreas. (2007). A discrete event simulation model in the case of managing a software project. *European Journal of Operational Research*. 181. 374-389. 10.1016/j.ejor.2006.05.031.
- [26] ZHANG, X. Application of discrete event simulation in health care: a systematic review. *BMC Health Serv Res* [online]. 2018, 18(618). DOI: <https://doi.org/10.1186/s12913-018-3456-4>. ISSN 1472-6963. Dostupné z: <https://bmchealthservres.biomedcentral.com/articles/10.1186/s12913-018-3456-4#citeas>
- [27] Životní cyklus simulačního projektu [online]. Dostupné z: <http://school.kjn.cz/modelovani-simulace/>
- [28] ALTIOK, Tayfur a Benjamin MELAMED. *Simulation modeling and analysis with Arena* [online]. 2007-08-13. Boston: Academic Press, c2007. ISBN 9780123705235. Dostupné z: <https://ebookcentral.proquest.com/lib/cvut/reader.action?docID=305656>
- [29] FlexSim: User guide [online]. Dostupné z: <https://docs.flexsim.com/>
- [30] TOMEK, Jan. *POČÍTAČOVÁ SIMULACE PODNIKOVÝCH PROCESŮ A SYSTÉMŮ*. Praha, 2020. Bakalářská práce. České vysoké učení technické v Praze.
- [31] ŠTŮSEK, Jaromír. Řízení provozu v logistických řetězcích. Praha : C.H. Beck, 2007. ISBN 978-80-7179-534-6.
- [32] Štíhlá logistika [online]. Dostupné z: <https://www.systemonline.cz/it-pro-logistiku/stihla-logistika.htm>
- [33] *Centre for Industrial Engineering: Sankeyův diagram* [online]. Plzeň, 2013. Dostupné z: <http://www.cie-plzen.cz/index.php/cz/lexikonmetod/sankeyuv-diagram>

[34] NENADÁL, Jaroslav. *Moderní management jakost – principy, postupy, metody*. Vyd. 1. Praha: Management Press, 2011, 377 s., ISBN 978-80-7261-186-7

[35] PLUTORA. *What is Value Stream Mapping (VSM), Benefits, Process and Value* [online]. Dostupné z: <https://www.plutora.com/blog/value-stream-mapping>

[36] BANKS, Jerry, John CARSON, Barry NELSON a David NICOL. *Discrete-Event System Simulation* [online]. 5. Harlow, UK: Pearson Education Limited, 2014. ISBN 978-1-292-02437-0.

[37] Bakešová, Miroslava a Křestán, Vladimír . 2008. *Základy Logistiky*. Jihlava : Vysoká škola polytechnická Jihlava, 2008. ISBN 978-80-87035-08-5.