

ŽILINSKÁ UNIVERZITA V ŽILINE

FAKULTA RIADENIA A INFORMATIKY

PODPORA RIADENIA PREVÁDZKY ŽELEZNIČNÝCH
DOPRAVNÝCH UZLOV V REÁLNO M ČASE S VYUŽITÍM
SIMULAČNÝCH MODELOV

Dizertačná práca

Študijný program: Aplikovaná informatika
Študijný odbor: Informatika
Pracovisko: Katedra matematických metód a operačnej analýzy
Fakulta: Fakulta riadenia a informatiky, Žilinská univerzita v Žiline
Školiteľ: doc. Ing. Norbert Adamko, PhD.

28360020243004

Žilina 2024

Ing. Andrea Galadíková

ČESTNÉ VYHLÁSENIE

Vyhlasujem, že som dizertačnú prácu s názvom *Podpora riadenia prevádzky železničných dopravných uzlov v reálnom čase s využitím simulačných modelov* vypracovala samostatne pod vedením školiteľa práce doc. Ing. Norberta Adamka, PhD., a uviedla v nej všetky použité literárne a iné odborné zdroje v súlade s právnymi predpismi a vnútornými predpismi Žilinskej univerzity v Žiline.

V Žiline, dňa 28.4.2024

.....

Andrea Galadíková

POĎAKOVANIE

Na tomto mieste by som sa chcela poďakovať školiteľovi mojej dizertačnej práce doc. Ing. Norbertovi Adamkovi, PhD. za cenné pripomienky a odborné rady, ktorými prispel k vypracovaniu tejto dizertačnej práce. Taktiež ďakujem môjmu manželovi Ladislavovi, mojim dcéram a celej rodine za nekonečnú trpezlivosť a podporu.

ABSTRAKT

Železničná doprava má dlhodobu stabilnú miesto v oblasti prepravy osôb a tovarov. Pre jej efektívne fungovanie sa ako jedna z mnohých dôležitých zložiek ukazuje kvalita operatívneho riadenia v železničných dopravných uzloch, preto je nevyhnutné zaoberať sa možnosťami poskytnutia podpory riadenia ich prevádzky. V súčasnosti sú v jednotlivých železničných dopravných uzloch využívané rôzne systémy, ktoré slúžia dispečerom ako podporný prostriedok pre realizáciu ich rozhodnutí. Vo veľkej miere sa jedná najmä o dispozičné systémy, v rámci ktorých zvyčajne nie sú plne využité všetky možnosti, ktoré by systém na podporu rozhodovania mohol poskytnúť. Pre zlepšenie kvality poskytnutej podpory sa ako vhodné ukazuje vytvorenie systému, ktorého základ je tvorený simulačným modelom a táto práca sa zaoberá návrhom takéhoto systému. Nevyhnutnou podmienkou existencie takéhoto systému je zabezpečenie kvalitných vstupných informácií z reálneho systému železničného uzla. V rámci práce sú popísané možnosti získavania takýchto informácií, či už o polohe jednotlivých vlakov, personálu, ale aj mnohých ďalších. V práci navrhujeme systém, v ktorého jadre stojí simulačný model, a preto uvádzame aj analýzu existujúcich simulačných nástrojov, ktoré umožňujú tvorbu simulačných modelov železničných dopravných uzlov. Ako vhodný sa ukázal simulačný softvér Villon, ktorý využíva agentovo orientovanú architektúru ABASim. Pre zabezpečenie použiteľnosti simulačných modelov vytvorených v tomto softvéri, boli realizované viaceré zmeny v implementácii tohto softvéru. Bol implementovaný systém, ktorý zabezpečuje aktuálnosť stavu simulačného modelu vzhľadom na reálny systém. Súčasťou práce je metodika, ktorá popisuje všetky kroky, ktoré je potrebné realizovať, pri tvorbe navrhovaného systému na podporu rozhodovania. Dôležitou časťou navrhovaného systému je aj použitie sofistikovaných metód, akými sú algoritmy postavené na exaktných matematických modeloch, heuristiky, neurónové siete a iné. V rámci overenia navrhovaného konceptu je v práci uvedené použitie dvoch rôznych optimalizačných prístupov na vybraných úlohách. Týmto prístupmi sú na simulácii založený optimalizačný algoritmus a vybraný algoritmus učenia posilňovaním. Dosiahnuté výsledky ukazujú zlepšenie v porovnaní so štandardnými metódami používanými v simulačných modeloch a potvrdzujú použiteľnosť navrhovaných prístupov.

Kľúčové slová: Simulácia, optimalizácia, železničná doprava, podpora rozhodovania

ABSTRACT

The railway transport has a long-standing and stable role in the transportation of people and goods. For its effective functioning, the quality of operational management in railway transport hubs appears to be one of the many important components, therefore it is necessary to address the possibilities of providing support for the management of their operation. Currently, several different systems are used in the railway transport nodes to support dispatchers in their decision-making. Most of those are disposition systems, which do not fully exploit all the possibilities that a decision support system could offer. The development of a system based on a simulation model appears to be an appropriate solution to improve the quality of decision support provided. This thesis deals with the proposal of such a system. A necessary prerequisite for the existence of such a system is ensuring quality input information from the real system of the railway node. The possibilities of obtaining such information, about the position of individual trains, personnel, and many other data are described in this thesis. We propose a system whose core is a simulation model, we also present an analysis of existing simulation tools that allow the creation of simulation models of railway nodes. The Villon simulation software, which is based on the agent-based ABAsim architecture, was found to be an appropriate choice. Several changes have been made in the implementation of this software to ensure the applicability of the simulation models created in it. A mechanism to ensure that the state of the simulation model is up-to-date with respect to the real system has been implemented. The thesis includes a methodology that describes all the steps necessary to implement the proposed decision support system. An important part of the proposed system is also the use of sophisticated methods, such as algorithms based on exact mathematical models, heuristics, neural networks, and others. In order to verify the decision support system with proposed architecture, the thesis presents the use of two different optimization approaches in selected tasks. These approaches are a simulation-based optimization algorithm and a selected reinforcement learning algorithm. The achieved results show an improvement over the standard methods used in simulation models and confirm the applicability of the proposed approaches.

Key words: Simulation, optimization, railways, decision support

Obsah

Zoznam obrázkov	2
Zoznam tabuliek	3
Zoznam skratiek	4
Úvod	6
1 Ciele dizertačnej práce	8
2 Analýza súčasnej situácie v železničných dopravných uzloch	10
2.1 Technologické postupy v železničných dopravných uzloch	10
2.2 Informačné systémy v železničnej doprave	12
2.2.1 Sietové informačné systémy	12
2.2.2 Prevádzkový informačný systém	15
2.2.3 Systémy využívané v železničnom dopravnom uzle	17
3 Návrh systému na podporu rozhodovania	20
3.1 Definícia požiadaviek na navrhovaný systém	20
3.2 Popis navrhovaného systému na podporu rozhodovania v reálnom čase	26
3.2.1 Interakčný modul	27
3.2.2 Hodnotiaci modul	28
3.2.3 Modul riešení	29
3.2.4 Spôsob nasadenia navrhovaného systému	29
4 Metodika tvorby systému na podporu rozhodovania s využitím simulačných modelov	31
4.1 Fáza 1 - Definovanie požiadaviek na systém	32
4.2 Fáza 2 - Definovanie dát, ktoré je potrebné čerpať z reálneho systému	32
4.3 Fáza 3 - Definovanie komunikačnej architektúry	33
4.4 Fáza 4 - Výber softvéru na tvorbu simulačného modelu	33
4.5 Fáza 5 - Integrácia spracovania dát z reálneho systému do softvéru na tvorbu simulačného modelu	34
4.6 Fáza 6 - Vytvorenie simulačného modelu	34
4.7 Fáza 7 - Zabezpečenie ukladania a opätovného načítania aktuálneho stavu simulačného modelu	35
4.8 Fáza 8 - Zabezpečenie prenosu dát	35

4.9	Fáza 9 - Overenie funkčnosti navrhovaného systému	36
5	Spôsoby získavania aktuálnych informácií o stave železničného dopravného uzla	37
5.1	Dáta využívané v systéme na podporu rozhodovania	37
5.1.1	Statické informácie	37
5.1.2	Dynamické informácie	38
5.2	Možnosti získavania informácií	38
5.2.1	Statické informácie	38
5.2.2	Dynamické informácie	39
5.3	Komunikačné štandardy	41
5.4	Definovanie komunikačného rozhrania	43
6	Simulačné modely železničných dopravných systémov	46
6.1	Softvér pre tvorbu simulačných modelov	46
6.1.1	Simulácia železničnej prevádzky - OpenTrack	47
6.1.2	RailSys	47
6.1.3	Rail Traffic Controller (RTC)	48
6.1.4	Simulačný softvér Villon	48
7	Rozšírenie architektúry ABAsim	51
7.1	Agentovo orientovaná simulácia - ABAsim	51
7.2	Návrh komunikačnej architektúry	54
7.2.1	Implementácia komunikačného agenta	58
7.3	Ukladanie aktuálneho stavu simulačného modelu	59
7.4	Zabezpečenie aktuálneho stavu vzhľadom na realitu	59
7.4.1	Zabezpečenie zmeny priradeného zdroja	61
7.4.2	Zabezpečenie zmeny polohy vlaku	61
7.4.3	Odobratie technologického kroku	62
7.4.4	Pridanie technologického kroku	62
7.4.5	Zmena času trvania technologického kroku	62
8	Implementácia navrhovaného systému bez sofistikovaných metód	63
9	Metodika použitia sofistikovaných metód v systémoch využívajúcich simulačné modely	67
9.1	Fáza 1 - Definovanie úlohy	67
9.2	Fáza 2 - Výber sofistikovanej metódy	68
9.3	Fáza 3 - Definovanie dát, potrebných pre použitie vybranej metódy	68
9.4	Fáza 4 - Implementácia sofistikovanej metódy	70

9.5	Fáza 5 - Definovanie komunikačného rozhrania	70
9.6	Fáza 6 - Úprava simulačného modelu	71
9.7	Fáza 7 - Prípravná fáza (trénovanie)	72
9.8	Fáza 8 - Nasadenie sofistikovanej metódy	72
9.9	Fáza 9 - Overenie funkčnosti systému	72
10	Integrácia optimalizačných metód do systému na podporu rozhodovania v reálnom čase	74
10.1	Aplikácia optimalizačných metód v železničnej doprave	74
10.1.1	Exaktné metódy	74
10.1.2	Heuristické metódy	76
10.1.3	Metódy strojového učenia	78
10.1.4	Na simulácii založené optimalizačné metódy	78
10.2	Možnosti integrácie výsledkov optimalizačných metód do systémov na podporu rozhodovania	80
10.2.1	Integrácia exaktného výpočtu pomocou matematického modelu	81
10.2.2	Integrácia heuristického prístupu	82
10.2.3	Integrácia učenia posilňovaním	82
11	Implementácia navrhovaného systému s využitím sofistikovaných metód	84
11.1	Úloha priradenia rušňovodičov	84
11.1.1	Použitie na simulácii založeného optimalizačného algoritmu	85
11.1.2	Použitie algoritmov učenia posilňovaním	92
11.2	Úloha určenia poradia úloh	95
11.2.1	Použitie na simulácii založeného optimalizačného algoritmu	96
11.2.2	Použitie algoritmov učenia posilňovaním - určenie poradia úloh	100
	Záver	107
	Zoznam použitej literatúry	109

Zoznam obrázkov

2.1	Schéma zriaďovacej stanice	11
2.2	Schéma údržbového depa	11
2.3	Dátové väzby GTN	14
2.4	VICOS	18
3.1	Use case diagram - používateľ	25
3.2	Architektúra systému na podporu rozhodovania založený na simulačnom modeli	26
3.3	Časová os použitia systému	28
6.1	Model v simulačnom softvéri Villon	49
7.1	Hierarchická štruktúra agentov	52
7.2	Dekompozícia agenta	53
7.3	Návrh komunikačnej architektúry	55
7.4	UML diagram tried, potrebných pre implementáciu komunikačného agenta	58
7.5	Zmena času trvania technologického kroku	62
11.1	Technológia obsluhy vlaku v simulačnom modeli údržbového depa	85
11.2	Simulačný model údržbového depa v softvéri Villon	86
11.3	Sekvenčný diagram inicializačného behu	89
11.4	Sekvenčný diagram optimalizačného algoritmu	89
11.5	Percentuálna zmena stavu „presúvajúci sa“	91
11.6	Percentuálna zmena stavu „vracajúci sa“	91
11.7	Tréningový proces algoritmu učenia posilňovaním	95
11.8	Simulačný model údržbového depa 2 v softvéri Villon	96
11.9	Percentuálna zmena súčtu oneskorení	100
11.10	Tréning PPO agenta v deterministickom prostredí	101
11.11	Tréning A2C agenta v deterministickom prostredí	102
11.12	Tréning PPO agenta v stochastickom prostredí	105

Zoznam tabuliek

3.1	Proces - zadávanie vstupov vyvolané používateľom	21
3.2	Proces - zadávanie vstupov vyvolané systémom	21
3.3	Proces - spúšťanie simulačných behov	22
3.4	Proces - využívanie sofistikovaných algoritmov	22
3.5	Proces - ovplyvnenie simulačného behu	23
3.6	Proces - vizualizácia stavu systému	23
3.7	Proces - Získavanie čiastkových výstupov	24
3.8	Proces - Získavanie celkových výstupov	24
8.1	Porovnanie ukazovateľov medzi simulačným modelom nahrádzajúcim realitu a systémom na podporu rozhodovania-testovací scenár 1	65
8.2	Porovnanie ukazovateľov medzi simulačným modelom nahrádzajúcim realitu a systémom na podporu rozhodovania-testovací scenár 2	65
8.3	Porovnanie ukazovateľov medzi simulačným modelom nahrádzajúcim realitu a systémom na podporu rozhodovania-testovací scenár 3	66
11.1	Porovnanie výsledkov experimentov s rôznym časovým výhľadom	92
11.2	Celkové meškanie pri použití rôznych časových horizontov	100
11.3	Porovnanie výsledkov experimentov s použitím algoritmu PPO pri určení poradia úloh . .	106

Zoznam skratiek

A2C	Simple Advantage Actor Critic
ASVC	Automatické stavanie vlakových ciest
CEVIS	Centrálny vozňový informačný systém
EDD	Elektronický dispečerský denník
ETCS	European Train Control System
FTP	File Transfer Protocol
GPS	Global Positioning System
GNSS	Global Navigation Satellite System
GSM	Global System for Mobile Communication
GTFS	General Transit Feed Specification
GTN	Graficko-technologická nadstavba zabezpečovacieho zariadenia
HKV	Hnacie koľajové vozidlo
HTTP	Hypertext Transfer Protokol
ISI	Informačný systém infraštruktúry ŽSR
IS MIP	Informačný systém Miestna práca s vozňom
JSON	JavaScript Object Notation
MIS	Miestny informačný systém 2
OMS	Operation Management System
PIS	Prevádzkový informačný systém
PPO	Proximal Policy Optimization
REST	Representational State Transfer
RFID	Radio Frequency IDentification
RTC	Rail Traffic Controller
RVD	Riadenie vlakovej dopravy
SGVD	Splnený grafikon vlakovej dopravy
SMTP	Simple Mail Transfer Protocol
TAF	Telematic Applications for Freight
TAP	Telematic Applications for Passenger
TCP	Transmission Control Protocol
TPV	Traťová poloha vlakov
TSI	Technical Specification for Interoperability
VDS	Vlakový dispečerský systém
VICOS	Vehicle and Infrastructure Control and Operating System

WSCI Web Service Choreography Interface
WSDL Web Services Description Language
XML eXtensible Markup Language
YMS Yard Management System
ZSSK Železničná spoločnosť Slovensko
ŽSR Železnice Slovenskej republiky

Úvod

Železničná doprava je považovaná za jeden z kľúčových prostriedkov na prepravu osôb aj tovaru, najmä vzhľadom na svoje ekologické vlastnosti. Železnica počas svojej dlhoročnej existencie prešla vývojom vo všetkých oblastiach svojej činnosti – technickej, prevádzkovej, personálnej i organizačnej. Napriek tomu nie sú v plnej miere využívané všetky možnosti, ktoré tento spôsob dopravy ponúka. Rovnako ako v iných oblastiach života, aj tu je potrebné zabezpečiť dostatočný stupeň informatizácie a automatizácie mnohých zložitých procesov.

Železničná sieť sa skladá zo železničných uzlov a zo železničných tratí, ktoré tieto uzly spájajú. Železničné dopravné uzly sú najmä stanice pre osobnú a nákladnú dopravu, zriaďovacie a odstavné stanice, dielne, rušňové a vozňové depo. Takýto uzol sa dá chápať ako komplexný obslužný systém, kde zákazníkmi sú vlaky a títo zákazníci potrebujú naplňať svoje požiadavky s využitím obmedzených zdrojov (zvyčajne koľaje, personál). Riadenie takého zložitého systému musí prebiehať na všetkých troch úrovniach a to na strategickej, taktickej aj operatívnej úrovni. Na strategickej úrovni vznikajú rozsiahle plány vo forme cestovných poriadkov, či plánovania pracovných zmien personálu. Tieto dlhodobé plány však nie sú funkčné na operatívnej úrovni a to najmä vzhľadom k tomu, že nedostatočne zahŕňajú náhodné vplyvy, ktoré môžu železničný uzol ovplyvniť. Takýmto vplyvom môže byť napríklad meškanie vlaku spôsobené neočakávanou situáciou na trati, neprítomnosť personálu, predĺženie času trvania jednotlivých technologických krokov z rôznych nepredvídateľných technických príčin a mnohé iné. Ak vlak príde do uzla s meškaním, môže sa stať, že koľaj, ktorá mu mala byť pridelená je už obsadená iným vlakom. Iný problém nastáva, ak zmeškaný vlak ovplyvní ďalšie naň naviazané vlakové spojenie. Predĺženie času trvania niektorého z technologických krokov môže spôsobiť, že na danej koľaji už mal v danom čase stáť iný vlak. Pri takýchto konfliktných situáciách je kľúčová práca dispečera. Tento musí na základe informácií o celom železničnom systéme, s využitím svojich vlastných vedomostí a skúseností vhodne rozhodnúť o prerozdelení jednotlivých obmedzených zdrojov. Toto môže byť veľmi komplikované a najmä takýto systém je plne závislý na ľudskom faktore, čo sa z hľadiska efektívnosti riadenia (najmä na operatívnej úrovni) nejaví ako dostatočne efektívne. Je preto potrebné zabezpečiť dispečerovi systém na podporu rozhodovania, ktorý by mu bol nápomocný pri hľadaní riešení jednotlivých problémov vyplývajúcich z kolíznych situácií.

Hoci v súčasnosti existuje mnoho informačných systémov, ktoré dispečeri využívajú pri svojej práci, žiadny z nich nie je dostatočne zameraný na problémy vznikajúce v železničnom dopravnom uzle, akým môže byť napríklad komplikovaná zriaďovacia stanica a tieto systémy sa nezaoberajú takýmito problémami komplexne. Riešením tohto problému by mohlo byť použitie počítačovej simulácie a simulačných modelov železničných dopravných uzlov a to najmä vzhľadom na mnohé dobré vlastnosti počítačovej simulácie (napr. možnosť analyzovať fungovanie systému detailne, bez potreby experimentovania so skutočným systémom). Táto dizertačná práca sa preto zaoberá možnosťami zahrnutia simulačných modelov železničných

dopravných uzlov do systému na podporu rozhodovania.

Táto práca sa zaoberá návrhom metodiky, v rámci ktorej sú popísané jednotlivé kroky, ktoré je potrebné zabezpečiť, aby bolo možné takýto systém na podporu rozhodovania vytvoriť a zároveň sú v tejto práci navrhnuté spôsoby, ako vytvorenie takéhoto systému zabezpečiť, čím poskytuje komplexný rámec vytvorenia systému na podporu rozhodovania dispečerov v železničnom dopravnom uzle.

V kapitole 1 tejto práce sú definované ciele dizertačnej práce. V kapitole 2 sú popísané jednotlivé systémy, ktoré sú v súčasnosti využívané dispečermi a ktoré im slúžia ako podporný prostriedok pre realizáciu ich rozhodnutí. V kapitole 3 sú špecifikované požiadavky na systém na podporu rozhodovania a následne sú popísané jednotlivé komponenty navrhovaného systému. Kapitola 4 je tvorená popisom metodiky, ktorá definuje postup pre vytvorenie daného systému. Aby mohol byť takýto systém realizovaný, je potrebné zabezpečiť kvalitné vstupné informácie z reálneho systému železnice. Možnosti získavania takýchto informácií, či už o polohe jednotlivých vlakov, personálu, ale aj mnohých ďalších sú popísané v kapitole 5. Keďže sa táto práca zaoberá systémom, v ktorého jadre je simulačný model, kapitola 6 sa zaoberá existujúcimi simulačnými nástrojmi, ktoré umožňujú tvorbu simulačných modelov železničných dopravných uzlov. Ako vhodný sa ukázal simulačný softvér Villon, ktorého jadro tvorí agentovo orientovaná architektúra ABASim. Kapitola 7 sa zaoberá implementačnými zmenami, ktoré je potrebné zabezpečiť, aby simulačné modely postavené na architektúre ABASim mohli stáť v jadre navrhovaného systému. V kapitole 8 je popísaná implementácia navrhovaného systému bez sofistikovaných metód a to najmä z hľadiska zabezpečenia aktuálneho stavu simulačného modelu vzhľadom na reálny systém. Kapitola 9 obsahuje metodiku integrácie sofistikovaných metód do navrhovaného systému. Kapitola 10 popisuje, aké sofistikované metódy je vhodné v danej oblasti použiť. Kapitola 11 popisuje implementáciu vybraných metód, konkrétne exaktného prístupu založeného na rozdelení celého datasetu na čiastkové úlohy a algoritmus učenia posilňovaním na vybranej úlohe priradenia personálu.

1 Ciele dizertačnej práce

Rozhodovanie v komplexnom systéme železničného dopravného uzla so sebou prináša mnoho komplikácií. Aby mohlo byť toto rozhodovanie dostatočne efektívne, je potrebné zaoberať sa podporou rozhodovania a to najmä na operatívnej úrovni. Na tejto úrovni je potrebné rozhodovať v reálnom čase. Vzhľadom na vlastnosti simulačných modelov sa tieto javia ako vhodné na poskytnutie podpory pri tvorbe rozhodnutí v reálnom čase. Systém obsahujúci simulačný model by mohol umožniť dispečerovi sledovať aktuálny stav reálneho systému a zároveň s týmto systémom experimentovať. Okrem týchto benefitov je potrebné zaoberať sa možnosťami integrácie sofistikovaných metód do systémov na podporu rozhodovania, nakoľko tieto metódy môžu priniesť návrhy, ako realizovať rozhodovanie. V súčasnosti však neexistuje jednoznačný postup akým realizovať spojenie simulačného modelu a optimalizačnej metódy do jedného systému, ktorý by vedel poskytnúť dostatočnú podporu v oblasti rozhodovania.

Hlavným cieľom dizertačnej práce bude navrhnúť a vypracovať metodiku tvorby systémov slúžiacich na poskytovanie podpory rozhodovania dispečerov železničného dopravného uzla v reálnom čase využívajúcich počítačovú simuláciu.

Pre realizáciu hlavného cieľa je potrebné naplniť niekoľko čiastkových cieľov:

- Návrh architektúry systému na podporu rozhodovania v reálnom čase, pričom významnou súčasťou tohto návrhu je návrh komunikačného rozhrania medzi jednotlivými komponentami tohto systému zohľadňujúci časovú efektívnosť dátového prenosu, rovnako aj objem prenášaných dát.
- Analýza a následné využitie možností simulačných modelov, medzi ktoré patrí poskytovanie aktualizovaného obrazu modelovaného systému, realizácia experimentov a dosahovanie celkových aj čiastkových výsledkov v reálnom čase, dostupnosť výstupov zo simulačného modelu v používateľsky prívetivej podobe a mnohé ďalšie.
- Analýza možností poskytovania údajov potrebných pre systém na podporu rozhodovania v reálnom čase z aktuálnych dispozičných systémov s dôrazom na identifikáciu problematických oblastí a následný návrh ich riešenia.
- Návrh a implementácia rozšírenia ABASim architektúry tak, aby bola umožnená jednoduchá integrácia podporných metód (vrátane podpory učenia sa) do existujúcich i budúcich simulačných modelov s cieľom ich využitia v systémoch na podporu rozhodovania.
- Overenie navrhovaného postupu na reálnom vybranom železničnom dopravnom uzle, pričom naplnenie tohto cieľa sa skláda z nasledujúcich čiastkových cieľov a to:
 - identifikácia a špecifikácia úlohy, v ktorej je vhodné poskytnúť podporu rozhodovania v železničnom dopravnom uzle v reálnom čase, pričom je dôležité klásť dôraz na komplexnosť prevádzky železničného uzla a tým aj na rozsah špecifikovanej úlohy.

- návrh a implementácia optimalizačných metód pre vybranú úlohu v železničnom dopravnom uzle. Vzhľadom na snahu zabezpečiť podporu v reálnom čase sa javí ako vhodné využitie heuristických metód, ktorých návrh štandardne vychádza z matematického modelu.
- dôsledné vyhodnotenie výsledkov dosiahnutých aplikáciou navrhovaného postupu a ich porovnanie s predchádzajúcim stavom.

2 Analýza súčasnej situácie v železničných dopravných uzloch

Železničná sieť je tvorená jednotlivými traťovými úsekmi a železničnými dopravnými uzlami, ktoré v rámci systému železničnej dopravy zabezpečujú určité technické zázemie zabezpečujúce plynulú prevádzku železnice. Rozlišujeme mnoho typov železničných dopravných uzlov, ako napríklad stanica pre osobnú, nákladnú alebo kombinovanú dopravu, rušňové a vozňové depo, zriaďovacia stanica, odstavná stanica, terminál, údržbové depo a iné.

Vo všetkých takýchto dopravných uzloch je dôležitá práca dispečera, ktorý zabezpečuje operatívne riadenie. Aby bolo možné lepšie pochopiť prácu dispečera Pre lepšie pochopenie práce dispečera je potrebné dôkladne popísať procesy v železničných dopravných uzloch.

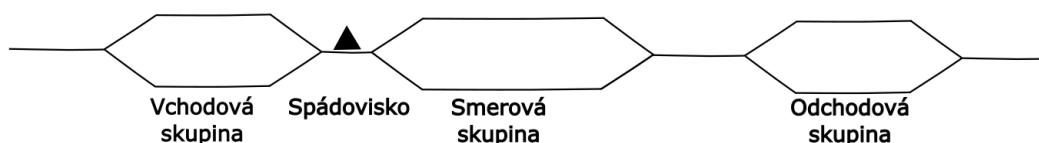
2.1 Technologické postupy v železničných dopravných uzloch

Stanice pre osobnú a nákladnú dopravu môžu byť medzistahové alebo úsekové. V medzistahovej stanici sú zabezpečované úkony ako vchody, odchody, prechody, križovanie a prechádzanie vlakov osobnej a nákladnej dopravy, posuny manipulačných vlakov, obsluha vlečiek a manipulačných koľají, v niektorých prípadoch odvesovanie vozňov od tranzitných vlakov, nástup a výstup cestujúcich. V úsekových stanicach sa okrem spomínaných činností vykonávajú technologické úkony s tranzitnými vlakmi, spravovanie cieľových vlakov, zostava východiskových vlakov, manipulácia so súpravami osobných vlakov vo východiskových alebo vratných stanicach, umývanie a dezinfekcia osobných a nákladných vozňov. Úsekové stanice disponujú odstavnými koľajami pre osobné vozne, rušňovým depom, vchodovými a odchodovými skupinami pre jednotlivé vlaky. [46]

Špecifickým typom železničnej stanice je zriaďovacia stanica. Hlavnou úlohou zriaďovacej (vlakotvornej) stanice je tvorba nových nákladných vlakov, ktoré je zabezpečovaná siedmymi hlavnými technologickými procesmi. Týmito procesmi sú:

- príjem cieľových vlakov zo železničnej siete - väčšinou prichádzajú na vchodové koľaje;
- príprava súprav cieľových vlakov na rozradenie. Táto príprava sa skladá z viacerých činností a to odpojenie a odstúpenie vlakového hnacieho koľajového vozidla a nastúpenie a pripojenie posunovacieho rušňa k rozradovanej súprave, odbrzdzenie brzd a uvoľnenie(rozpojenie) skrutkoviek, technická a prepravná prehliadka vozňov;
- rozradovanie súprav vlakov, pričom vznikajú z vlakov odvesy a tieto sa posunom dostanú na smerové koľaje. Posunom sa vo väčšine zriaďovacích staníc rozumie spúšťanie. Spúšťanie je proces zahŕňajúci prísun rozradovanej súpravy na zvažný pahorok pomocou posunovacieho rušňa, samostatná jazda

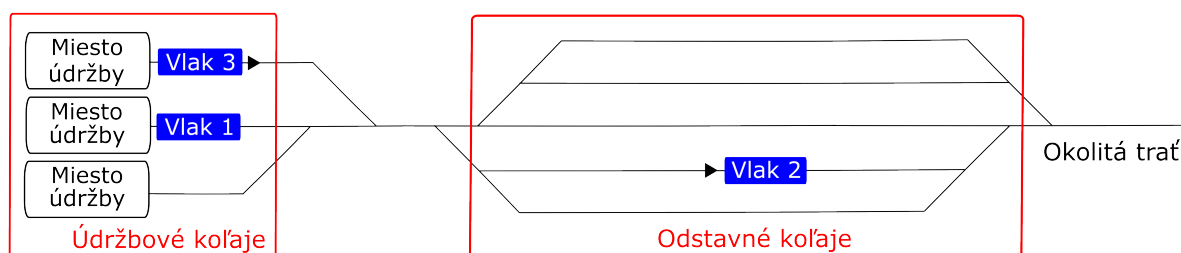
- odvesov z vrcholu pahorku na smerovú koľaj a zabrzdzenie odvesov na smerovej koľaji;
- zhromažďovanie vozňov na smerových koľajach;
- zostava súprav skupinových vlakov;
- obsluha vozňov zhromaždených na smerových koľajach - stláčanie odvesov, spájanie vozňov, spájanie hadíc priebežnej vzduchovej brzdy;
- príprava súprav východiskových vlakov, ktorá sa môže začať po prestavení vozňov zo smerových koľají na odchodové. Táto príprava zahŕňa nástup rušňa, úplnú skúšku brzdy, označenie konca vlaku, technickú a prepravnú prehliadku, odovzdanie sprievodných listín vlakvedúcemu. [48]



Obr. 2.1: Schéma zriaďovacej stanice [48]

Rôzne typy terminálov zabezpečujú najmä prekládku tovaru v rámci železničnej dopravy, prípadne na iné typy dopravy (nákladné autá, lode). V kontajnerovom termináli sa realizuje prekládka naplánovaných zásielok. Prichádzajúcemu vlaku je potrebné prideliť koľaj tak, aby bolo možné realizovať vykládku podľa typu zásielky. Po príchode vlaku je vykonaná kontrola jednotlivých vozňov a údaje o prijatej zásielke sú evidované do informačného systému. Následne sú tovary preložené z kontajnerov na tahač a do kontajnerov sú naložené ďalšie zásielky. Po skontrolovaní jednotlivých vozňov na základe plánu, môže vlak odísť.

V rámci údržbového depa je potrebné zabezpečiť údržbu prichádzajúcich vlakov. Pod údržbou sa myslí čistenie vlaku, kontrola jednotlivých komponentov a ich prípadná oprava. Vlaky prichádzajú do údržbového depa podľa vopred určeného harmonogramu a majú dané, aký typ údržby je potrebný. Aby mohla byť realizovaná takáto údržba, vlaky sa potrebujú dostať na jednotlivé miesta údržby (hala, servisná jama a ďalšie). Ak je toto miesto obsadené, vlak prichádzajúci z trate zastane v skupine koľají určenej na čakanie. Na takejto koľaji zastane aj v prípade, že jeho údržba bola dokončená, ale ešte nenastal čas jeho



Obr. 2.2: Schéma údržbového depa

plánovaného odchodu na trať. V niektorých prípadoch môže v rámci údržbového depa dochádzať k tvorbe nových vlakov, rozpojením prichádzajúcej súpravy, avšak v mnohých údržbových depách sa toto nedeje a údržba na vlaku sa vykonáva bez rozpojenia súpravy. V rámci údržbového depa sú dôležitým personálom najmä rušnovodiči zabezpečujúci presun vlakov medzi jednotlivými miestami údržby a personál zabezpečujúci čistenie, kontroly a opravy vlakov.

2.2 Informačné systémy v železničnej doprave

Operatívne riadenie v tak komplexnom systéme ako je železničný dopravný uzol je kľúčové pre efektívne fungovanie systému. Napriek tomu, že na strategickej úrovni sú zostavené robustné plány, ktorými sa prevádzka má riadiť, dispečer počas prevádzky v reálnom čase často rieši situácie, na ktoré sa takýto plán nedá aplikovať. V takýchto prípadoch sa rozhodnutia dispečera zvyčajne zakladajú na informáciách dostupných v aktuálnom čase (zhromaždených napríklad z dispečerských informačných systémov), na obmedzených znalostiach očakávaného budúceho vývoja (napríklad plánovaných príchodoch vlakov) a na osobných skúsenostiach dispečera. Pre zlepšenie kvality rozhodnutí dispečera je vhodné využívať informačné technológie ako podporný prostriedok pre realizáciu riadenia. Práve preto existuje viacero rôznych informačných systémov pre podporu rozhodovania dispečerov.

Hoci sa táto dizertačná práca zaoberá najmä prevádzkou v železničných dopravných uzloch, systémy, ktoré sa používajú pre sledovanie a riadenie situácie v železničnej sieti sú takisto dôležité, keďže efektívne fungovanie železničného dopravného uzla je závislé aj na situácii mimo tento uzol a takého informačné systémy môžu byť cenným zdrojom informácií pre dispečerské riadenie v železničnom dopravnom uzle.

2.2.1 Sieťové informačné systémy

Sieťové informačné systémy sa zaoberajú sledovaním stavu na trati, medzi jednotlivými železničnými dopravnými uzlami. Medzi dôležité systémy, ktoré v súčasnosti poskytujú dispečerom podporu pri riadení situácie v železničnej sieti radíme Vlakový dispečerský systém, Traťovú polohu vlakov, Elektronický dopravný denník, Riadenie vlakovej dopravy a Graficko technologickú nadstavbu zabezpečovacieho zariadenia. Tieto informačné systémy sú poskytované spoločnosťou Oltis Group [23]. Tieto systémy podporujú tvorbu ročných aj krátkodobých cestovných poriadkov, optimálne využitie zdrojov a ich priradenie k jednotlivým dopravným výkonom, vrátane sledovania jász vlakov, ale aj iných elementov (lokomotív, vlakového personálu) v reálnom čase. Systémy okrem toho poskytujú dispečerovi vizualizáciu aktuálnej a očakávanej situácie a tým podporu pri korekcii vzniknutých odchýlok. Systémy zabezpečujú štandardizovanú dátovú komunikáciu s manažérom infraštruktúry, ako aj medzi dispečermi či personálom vo vlaku.

Vlakový dispečerský systém (VDS) je systém poskytujúci podporu riadenia najmä grafickým zobrazením aktuálnej situácie (stav HDV, vozňa, personálu vo vzťahu k plánu GVD, plánu HDV, súprav, personálu a reálnej jazdy vlaku) a prognózy vývoja dopravnej situácie (splnený GVD). Umožňuje filtrovanie zobrazeného traťového úseku aj množiny vlakov a pomáha zabezpečiť včasnú a efektívnu podporu

pre riešenie prevádzkových mimoriadností a nehôd. VDS poskytuje relevantné informácie manažérovi infraštruktúry, ostatným dopravcom a zákazníkom dopravcu v zmysle platnej legislatívy a noriem Telematic Applications for Freight (TAF), Technical Specification for Interoperability (TSI), Zozbierané informácie (napr. automatický zber a import dát z EDD) sú zhromažďované do jednotného úložiska údajov a toto úložisko umožňuje získavať požadované výstupy aj ďalším zložkám dopravcu. Systém VDS má priame komunikačné spojenie s PIS, ILTIS, GTN, EDD a SGVD.

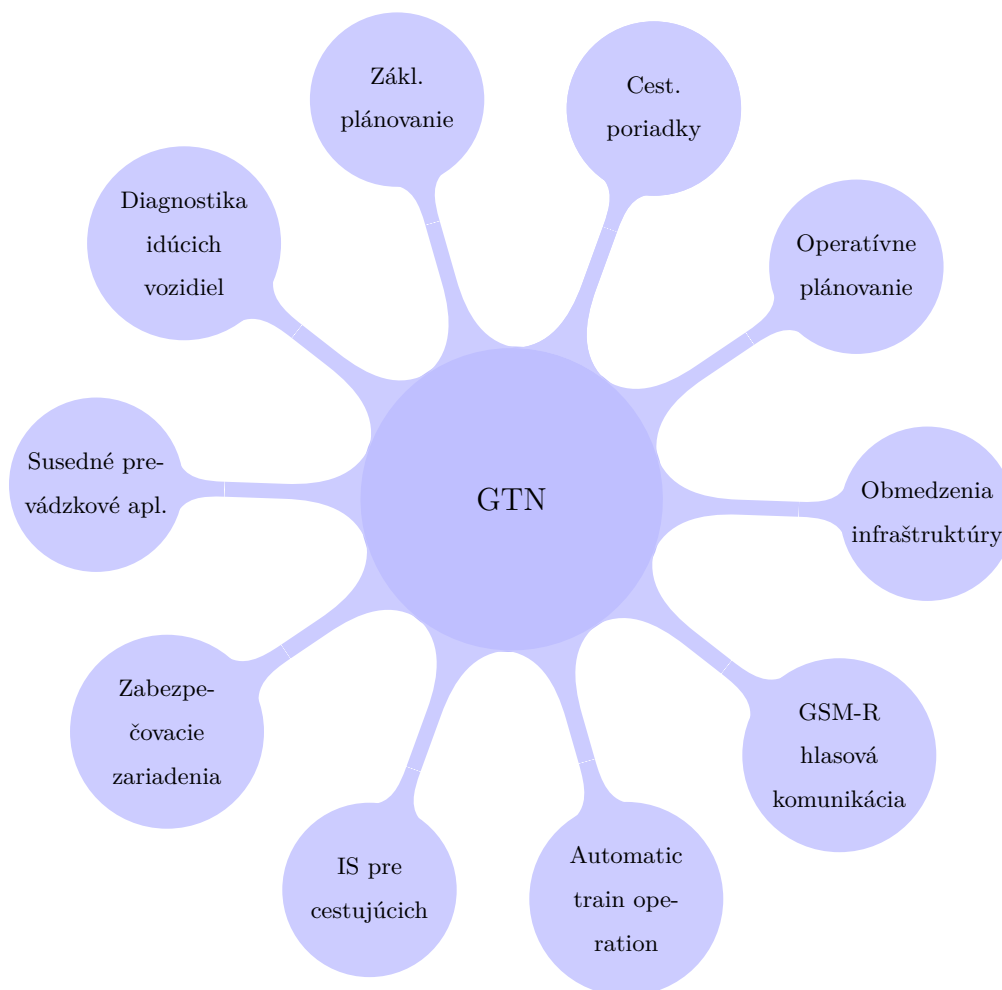
Trafová poloha vlakov (TPV) je aplikácia slúžiaca na podporu operatívneho riadenia dopravného procesu. Aplikácia grafickou formou znázorňuje jazdy vlakov a priebehy výluk vo vybranom traťovom úseku a dopravných bodoch. Dopravným zamestnancom v stanici poskytuje prehľad o prevádzkovej situácii na vybraných tratiach. V prípade mimoriadností a výluk umožňuje operatívne korigovať dopravné rozhodnutia a tým zabezpečiť plynulosť dopravy aj počas výlukových, alebo inak obmedzujúcich, aktivít už na najnižšej úrovni riadenia. [69]

Elektronický dispečerský denník (EDD) slúži na zaznamenávanie dopravných činností spojených s riadením dopravy a jeho úlohou je podpora sledovania jazdy vlakov a posunu medzi dopravňami, v dopravnom bode a príslušných úsekoch, slúži aj na zber dynamických informácií. [11] Činnosťami, ktoré sú uchovávané v EDD sú zápis vlaku, zápis posunu medzi dopravňami, začatie výluky, ukončenie výluky, prehliadanie začatých výluk, plán výluk, práce s varovným štítkom, zmena stavu traťového zabezpečovacieho zariadenia, výluka dopravnej služby vrátane odovzdávky, nemožné dorozumenie, komunikácia k jazde vlaku, kvitancia, poloha vlaku, pripravenosť k odchodu, odstavenie vlaku, mimoriadnosti na vlaku a iné.

Riadenie vlakovej dopravy (RVD) je aplikáciou umožňujúcou automatické plánovanie jazd vlakov vrátane odchýlok, automatické plánovanie nasadenia hnacích vozidiel a strojvedúcich na jednotlivé výkony. Okrem toho systém upozorňuje na potrebný zásah dispečera, najmä tým, že realizuje zber dát o jazde vlaku a ďalších udalostí v železničnej stanici. Dôležitou funkcionalitou je automatická distribúcia plánov a predhlások polôh vlaku, hnacích vozidiel a strojvedúcich na okolité systémy (GTN, VDS, VLASTA a iné). [23]

Graficko-technologická nadstavba zabezpečovacieho zariadenia (GTN) je riadiaci systém na priamu úroveň riadenia, pre dispečerov a výpravcov. [41] Tento systém v reálnom čase monitoruje činnosť zabezpečovacích zariadení a pomocou zozbieraných údajov informuje o aktuálnom stave vlakovej dopravy v pridelennej oblasti (na traťovom úseku aj v jednotlivých dopravniciach). Okrem toho vedie dopravnú štatistiku pre danú oblasť. Umožňuje aj tvorbu prognostického modelu a tým čiastočne zabezpečuje plánovanie dopravy. Medzi dôležité funkcie patrí aj detekcia a indikácia infraštruktúrnych konfliktov, konkrétne konfliktov vlakov s infraštruktúrou. Traťoví dispečeri sú teda vopred informovaní na hroziacu situáciu, keď je vlak určený na prepravu osôb predelený na kolaj bez nástupišťa, pri nákladných vlakoch sa zas porovnáva dĺžka vlaku a dĺžka plánovanej kolaje v stanici. [68] Súčasťou GTN je aj Automatické stavanie vlakových ciest (ASVC) s integráciou algoritmov zaisťujúcich inicializáciu stavania vlakovej cesty prostredníctvom geografickej metódy, ktorá spočíva v sledovaní reálnej polohy vlaku v zabezpečovacom zariadení. [67] [42]

GTN je súčasťou zložitého systému na podporu riadenia dispečerov a obrázok 2.3 zobrazuje dátové väzby s inými systémami.



Obr. 2.3: Dátové väzby GTN [59]

Dôležitými sú najmä väzby na iné prevádzkové aplikácie. V poslednom období vznikla potreba zefektívniť komunikáciu medzi ISOŘ (na Slovensku PIS) a GTN. Trasa každého vlaku (určená GVD) je v GTN pevne zviazaná s konkrétnou trasou daného dňa v ISOŘ pomocou jednoznačnej európskej identifikácie vlaku v zmysle TSI TAF/ Telematic Applications for Passenger (TAP). Technicky sú prehlášky z ISOŘ do GTN štruktúrované do troch samostatných typov (správa 7261, 7263 a 7265). Jednotlivé správy majú presne definované elementy (vlak, trasa, mimoriadna udalosť a ďalšie) a tieto elementy sú ďalej definované vlastnými elementami (vlak - TrainTransportID, názov, stav vlaku, východisková stanica, cieľová stanica, číslo vlaku, dopravca, dopravca EU, deň jazdy, meškanie, verzia trasy, verzia parametrov).[59] Komunikačné rozhranie s ostatnými IS je realizované cez Web Service Choreography Interface (WSCI) pomocou protokolu SOAP, je používaný jazyk Web Services Description Language (WSDL) a sú dodržiavané eXtensible Markup Language (XML) štandardy. Toto komunikačné rozhranie používa podobné

pravidlá ako komunikační server Common Interface. Je to striktne definovaná webová služba s jedinou funkciou s jedným parametrom typu String. Volanie prebieha v oboch smeroch symetricky, čo znamená, že keď je odovzdávaná správa z klienta na server CI, je volaná aj webová služba vystavená serverom. Pri odovzdávaní správy zo servera na klienta volá server službu vystavenú klientom. Pre zabezpečenie kompatibility posielaných správ obsahuje parameter typu String nepovinný element k prenosu 64 bitovej komunikačnej hlavičky podľa zásad protokolu TcpKS.

Informačný systém infraštruktúry ŽSR (ISI) poskytuje informácie o infraštruktúre železničnej siete na Slovensku a pokrýva nasledovné úlohy:

- zabezpečuje centrálnu evidenciu objektov infraštruktúry Železnice Slovenskej republiky (ŽSR);
- poskytuje informácie o každom evidovanom objekte infraštruktúry ŽSR;
- umožňuje jednoduchú a operatívnu aktualizáciu dát (grafická podpora);
- vytvára centrálnu databázu infraštruktúry ŽSR;
- poskytuje schémy, nákresy, konštrukčné výkresy;
- poskytuje fotodokumentáciu objektov;
- zabezpečuje publikovanie vybraných údajov na intranetových a internetových stránkach pre potreby zamestnancov ŽSR, dopravcov a ostatných klientov ŽSR.

Centrálny vozňový informačný systém (CEVIS) je celosieťovým systémom zameraným na sledovanie vlakov, vozňov a zásielok. Poskytuje informácie o vlaku ako súpis vozňov východiskového vlaku, odjazd vlaku z východiskovej stanice, prejazd vlaku kontrolným bodom, odstavenie vlaku, odjazd vlaku po odstavení, prečíslovanie vlaku a zmena cieľovej stanice, privesenie / odvesenie vozňa od vlaku, ukončenie jazdy vlaku. [21]

V súvislosti so zberom dát potrebným pre fungovanie informačných systémov je v železničnej doprave využívané prenosné zariadenie PSION (minipočítač), ktoré je určené pre urýchlenie zberu údajov o vozňoch a vlakoch priamo z koľajiska. Do systému PSION je možné preniesť údaje o stanici aj o konkrétnom vlaku a pomocou zariadenia PSION je možné vytvoriť súpis vlaku. Do informačných systémov sú informácie zo PSION prenášané pomocou siete Global System for Mobile Communication (GSM). [21]

2.2.2 Prevádzkový informačný systém

Prevádzkový informačný systém (PIS) [22] je dôležitý informačný systém, ktorý pomáha riadiť prevádzku v celej železničnej doprave, čiže rovnako v rámci železničnej siete, ako aj v jednotlivých dopravných uzloch. Tento systém poskytuje informačnú podporu zamestnancom dopravcu a zamestnancom ŽSR pre procesy súvisiace s nasledujúcimi oblasťami:

- plánovanie vlakovej dopravy,
- prevádzka a riadenie vlakovej dopravy,
- komunikácia s IS dopravcov, medzinárodnými a susednými IS,
- hodnotenia vlakovej dopravy (výstupy, prehľady, analýzy),

- odúčtovanie úhrady za prístup k železničnej infraštruktúre a servisným zariadeniam,
- spravovanie kmeňových dát IS PIS.

PIS je tvorený viacerými modulmi, ktoré sú vzájomne previazané a dokážu medzi sebou komunikovať. Tento systém disponuje mnohými informáciami o stave na trati a tieto informácie sú uchovávané v databázovom systéme. Číselníky a databázy PIS delíme na:

- Číselníky a databázy popisu a stavu železničnej siete v správe ŽSR:
 - Vnútroštátnych bodov a hraničných priechodov; staničných koľají; dopravných úsekov - kilometrických vzdialeností; plánovaných výluk.
- Databázy koľajových vozidiel:
 - Nákladných vozňov; osobných vozňov; Hnacie koľajové vozidlo (HKV) a ich radov; traťových strojov a ich typov.
- Číselníky a databázy používané na identifikáciu objektov, dopravných dejov, vlastností KV alebo nimi prepravovaného tovaru:
 - Určenia vlaku; pohybu a stavu vlaku; dopravcov; doplňujúcich údajov o vozni; dôvodov meškania vlaku; smerovania vozňových zásielok; UN kódu, IČN; dôvodu odstavenia vlaku; cestovných poriadkov vlakov.

Vstupné informácie PIS sa delia na statické informácie a dynamické informácie. Statickými informáciami sú objednávka vlakovej trasy či presun koľajového vozidla. Dynamické informácie hovoria o pohybe a dôvodoch meškania vlaku a môžu byť zadávané ručne pomocou aplikácií PIS alebo prijímané z nadstavieb elektronických zabezpečovacích systémov.

Okrem toho môžeme vstupné informácie rozdeliť na informácie zadávané dopravcom a informácie zadávané zamestnancami ŽSR. Dopravca zadáva objednávku vlakovej trasy; vstupné vety - „Súpis vlaku“, alebo „Súpis vlaku osobnej dopravy“; presun koľajového vozidla. Zamestnanec ŽSR (zvyčajne dispečer) spracováva jednotlivé objednávky vlakových trás. Okrem toho eviduje príchod, prechod a odchod vlaku, dôvod meškania alebo predpokladané meškание vlaku. V rámci spravovania údajov v PIS je prevádzkový dispečer povinný sledovať úplnosť dôvodov meškania u všetkých vlakov, ako aj ich správnosť, ak sa hodnota meškania navýšila a zaznamenávať v aplikácii *vlakového dispečerského systému* všetky udelené dispečerské príkazy pri príslušnom vlaku. [72]

Z vyššie uvedeného teda vyplýva, že aktuálne údaje pre tento systém sú získavané pri prejazde vlaku cez definovaný dopravný bod a sú teda priebežne aktualizované. Pomocou nich je možné vyhodnotiť skutočný priebeh dopravného procesu, ako aj model vývoja dopravnej situácie, čo významne uľahčuje prácu dispečerovi pri rozhodovaní.

Informačný systém zabezpečovacieho zariadenia Informačný systém Miestna práca s vozňom (IS MIP), ktorý je súčasťou PIS umožňuje evidenciu úkonov, ktoré sa týkajú procesu vlakotvorby s miestnou prácou v ŽST, ako sú kontrola po príchode vlaku, vytvorenie číselnej triedenky a rozradenie vozňov, druhotná

manipulácia a príprava finálneho súpisu vlaku, ako aj komplexnú podporu pri plánovaní miestnej práce, kontrole aktuálneho, či budúceho stavu obsadenosti železničnej stanice a sledovanie histórie vlakov a vozidiel. Vďaka týmto funkcionalitám je MIP dôležitou súčasťou informačných systémov najmä v zriaďovacích staniciach.

Podobným zástupcom informačných systémov bol v minulosti Miestny informačný systém 2 (MIS) [50]. MIS zabezpečuje sledovanie vlakov približujúcich sa k uzlu, údaje potrebné pre spracovanie vlakov po príchode do uzla a pred odjazdom z uzla, informácie pre staničného dispečera a vozňového disponenta, ďalej obsahuje archívne informácie o vlakoch, ako aj operatívnu evidenciu uzla[53].

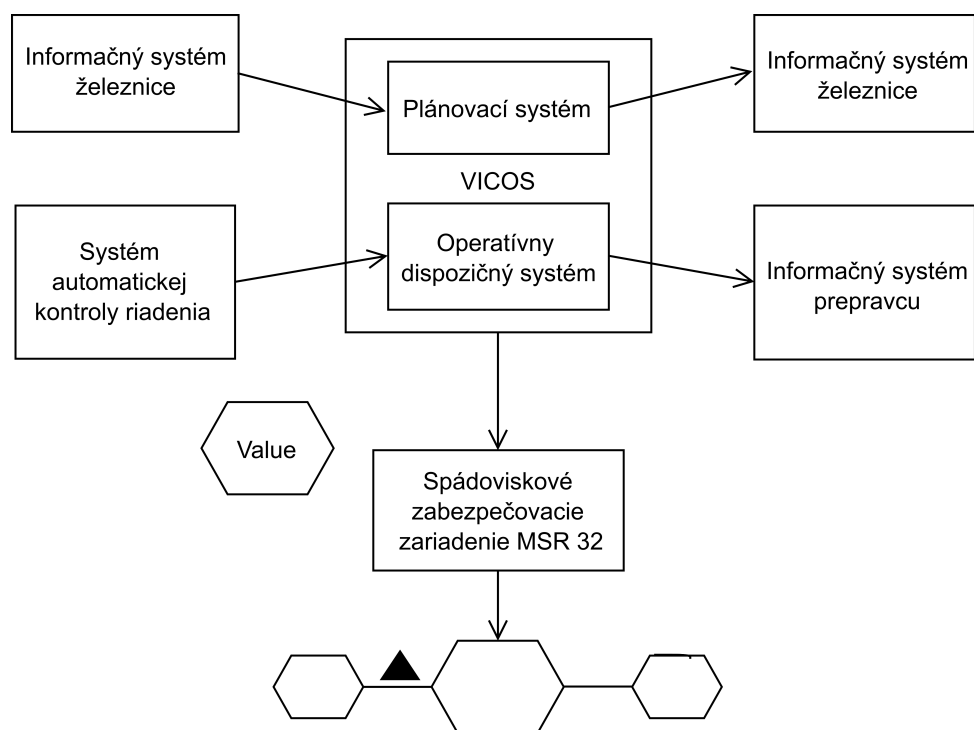
2.2.3 Systémy využívané v železničnom dopravnom uzle

Dispozičný systém Berghof[47] je centralizovaný systém zameraný na monitorovanie (prezentuje stav systému a umožňuje zistiť polohu vozidiel vo vybranej oblasti), plánovanie a riadenie a tým aj optimalizáciu zložitých dopravných procesov najmä v železničných depách.

Ďalším zo systémov je zákaznícky portál ISP, ktorý je súčasťou webového portálu Železničná spoločnosť Slovensko (ZSSK) Cargo a umožňuje dispečerom (najmä nákladnej vlakovej dopravy) efektívnu prácu s dátami z jednotlivých informačných systémov ZSSK Cargo, hlavne z interného informačného systému prevádzky ZSSK CARGO – ISP. Umožňuje sledovať polohu a stav zásielok a sledovať polohu a stav jednotlivých vozňov. [71]

Spoločnosť Siemens Mobility [51] je dlhoročným dodávateľom rôznych systémov pre podporu riadenia železničnej dopravy a v súvislosti s dispečerským riadením železničných dopravných uzlov je využívaný najmä systém VICOS, ale aj niektoré ďalšie systémy.

Dispozičný systém Vehicle and Infrastructure Control and Operating System (VICOS) [52] umožňuje monitorovať a riadiť železničné dopravné systémy. V rámci neho je možné overiť prevádzkyschopnosť jednotlivých cestovných poriadkov z hľadiska rôznych technických obmedzení, napríklad dynamické správanie sa vlakov alebo výhybiek. Systém môže tiež simulovať rôzne poruchy na trati alebo dočasné obmedzenia rýchlosti a s tým spojené prevádzkové situácie. Pri podpore rozhodovania sa však nezameriava na problémy priamo v dopravnom uzle, neuvažuje priamo s personálom ani ďalšími zdrojmi obsluhy. Pre podporu rozhodovania ponúka dispečerovi predprogramované sekvencie riešenia problémov na základe špecifik jednotlivých železničných systémov. Systém umožňuje dispečerovi železničného dopravného uzla zmenu smerovej koľaje, odrieknutie vlaku, zavádzanie mimoriadnych vlakov, zmenu poradia obsluhy súprav, kontrolu vozňov s prednostným tovarom a iné. Systém VICOS okrem iného umožňuje zostavu dlhodobého i krátkodobého plánu práce zamestnancov. Pre zabezpečenie funkcionalít systému VICOS je dôležité spomenúť jeho komunikáciu s ďalšími systémami, akými sú systémy SCADA, informačné systémy pre cestujúcich, kamerové systémy (CCTV), komunikačné systémy, veľkoplošné obrazovky, monitory, European Train Control System (ETCS), ktoré slúžia ako zdroje dát a tým umožňujú podporu v rozhodovacom procese dispečera [46].



Obr. 2.4: VICOS [46]

Operation Management System (OMS) je ďalším produktom spoločnosti Siemens, ktorý poskytuje podporu riadenia. Jeho dôležitým podsystémom je Yard Management System (YMS), ktorý je využívaným najmä v údržbových staniciach. Tento systém umožňuje sledovať celý proces údržby vlaku, od jeho vstupu do stanice až po jeho opätovný odchod. Systém poskytuje vizualizáciu a ovládanie jednotlivých zariadení v stanici, mapuje obsadenosť jednotlivých hál a stav obsadenosti koľajiska, sleduje vlak počas celého jeho pobytu v stanici, ako aj jeho aktuálny stav a polohu, dokumentuje komunikáciu medzi všetkými osobami zapojenými do procesu, ukladá informácie o údržbových procesoch vykonaných na vlaku, poskytuje údaje o cestovných poriadkoch vlakov, umožňuje výmenu dát s inými systémami.

Pre potreby riadenia v zriaďovacích staniciach bol vyvinutý Trackguard Cargo MSR32 systém. Jedná sa o modulárny, otvorený mikropočítačový systém, pričom jednotlivé mikropočítače sú pripojené k riadiacemu počítaču pomocou dátovej zbernice LAN. Všetky riadiace jednotky systému pracujú v reálnom čase. Komplexné funkcie zaznamenávania tohto systému umožňujú dispečerom kontrolovať všetky dôležité udalosti v zriaďovacej stanici (obsadenosť jednotlivých koľají, prevádzkový stav jednotlivých koľajových bŕzd, atď.). Okrem toho umožňuje MSR32 sledovať nezrovnalosti s naplánovanými udalosťami a včas na ne reagovať. Týmto systémom je zabezpečená kontrola nad rozraďovacími a zhromažďovacími operáciami v zriaďovacej stanici. Tento systém umožňuje monitorovať závary a manipulačné zariadenia na trati. V spojitosti s týmto systémom je dôležité spomenúť komunikáciu s koľajovými brzdami, ktorá umožňuje monitoring spomínaných udalostí.

Pre potreby získavania údajov v reálnom čase vyvinula spoločnosť Siemens systém Controlguide CT-

mobile. Tento systém umožňuje sledovať údaje o jednotlivých vozňoch v reálnom čase a to tak, že na tieto vozne sú umiestnené senzory, monitorujúce napríklad teplotu, tlak, vlhkosť, stav naplnenia vozňa, stav otvorenia dverí, polohu páky či polohu výklopných krytov. Tieto senzory majú integrované rádiové spojenie, ktoré umožňuje prenos dát na krátke vzdialenosti a pomocou tohto spojenia sú údaje z týchto senzorov prenášané do CTmobile jednotky, ktorá zachytáva aj údaje z Global Positioning System (GPS) / Global Navigation Satellite System (GNSS) o polohe vozňa. Ďalej sú tieto údaje pomocou GSM, prípadne GPRS siete prenesené do Controlguide CTcentral. Controlguide CTcentral je dátové centrum, do ktorého sú prijímané všetky dáta z vozidiel. Tieto dáta sa ukladajú do databázy, spracúvajú pre prístup cez webovú prezentačnú vrstvu a sú ďalej distribuované pomocou internetu k používateľom. Títo môžu údaje sledovať pomocou SMS notifikácií, emailu alebo cez webový prehliadač.

Existujúce systémy využívajú sice rôzne algoritmy, avšak nie sú schopné rozsiahlej analýzy situácie a ich algoritmy nie sú dostatočne sofistikované, aby dokázali nájsť optimálne riešenia. Ďalším problémom je, že mnohé informácie sú zadávané ručne personálom železnice, čo môže spôsobiť nepresnosť údajov. Informácie o polohe personálu (napr. posunovači a tranzitéri v zriaďovacích staniciach) vo väčšine používaných systémov nie sú dostatočné, alebo absentujú úplne. Naopak informácie o vlakoch sú mnohokrát duplicitné a nekonzistentné. Na základe analýz preto považujeme za dôležité, zaoberať sa návrhom komplexného systému určeného na podporu rozhodovania dispečerov. Vzhľadom na spomenuté nedostatky používaných systémov považujeme za prínosné navrhnutie takého systému na podporu rozhodovania, ktorý by dispečerom umožnil sledovanie aktuálnej situácie v železničnom dopravnom uzle, umožňoval experimentovanie so systémom železničného dopravného uzla a dokázal prinášať sofistikované riešenia. Na základe týchto požiadaviek je vhodné použitie simulačných modelov.

3 Návrh systému na podporu rozhodovania

Na základe analýzy súčasného stavu existujúcich informačných dispozičných systémov v oblasti dispečerského riadenia železničných dopravných uzlov je možné povedať, že podpora riadenia v súčasnosti nie je dostačujúca. Vzhľadom k tomu bola definovaná požiadavka vytvorenia systému na podporu rozhodovania dispečerov v reálnom čase. Jednotlivé požiadavky, ako aj komplexný návrh takéhoto systému je popísaný v nasledujúcej kapitole.

3.1 Definícia požiadaviek na navrhovaný systém

Požiadavky na navrhovaný systém vyplývajú z potrieb samotných dispečerov. Aby mohol navrhovaný systém poskytovať dispečerom podporu v ich rozhodovacom procese je potrebné, aby takýto systém dokázal nielen prezentovať stav systému v reálnom čase (podobne ako štandardný informačný systém), ale aby zároveň ponúkol aj pohľad do budúcnosti, či minulosti, v používateľsky vhodnej forme. Pod používateľsky vhodnou formou možno rozumieť prívetivé a intuitívne grafické používateľské rozhranie, zahŕňajúce animáciu stavu v železničnom dopravnom uzle, ako aj výstupy v podobe rôznych grafov a tabuliek.

Pohľad do budúcnosti by mal umožniť používateľovi realizáciu viacerých konfigurácií budúceho vývoja v podobe rôznych experimentov. Počas týchto experimentov môže vstupovať do vývoja stavu modelovaného systému a priamo vykonávať niektoré rozhodnutia (human-in-the-loop systém). Výsledky týchto experimentov (zahŕňajúce rôzne štatistické ukazovatele) by malo byť možné prezentovať v podobe dobre čitateľnej pre používateľa.

Keďže v súčasnosti je rozhodovací proces závislý na ľudskom faktore, navrhovaný systém by mal čiastočne znížiť túto závislosť poskytnutím návrhov rozhodnutí (zvyčajne o pridelení obmedzených zdrojov). Možnosť poznať stav systému v budúcnosti je vhodným základom pre aplikáciu sofistikovaných algoritmov poskytujúcich dispečerovi návrh riešení v prípade vzniku nepravidielností v prevádzke železničného dopravného uzla.

Na základe vyššie popísaného boli stanovené nasledujúce funkčné požiadavky na navrhovaný systém:

- zadávanie vstupov vyvolané používateľom,
- zadávanie vstupov vyvolané systémom,
- spúšťanie simulačných behov,
- výber sofistikovaných algoritmov,
- ovplyvnenie simulačného behu,
- vizualizácia stavu systému,
- získavanie čiastkových výstupov,

- získavanie celkových výstupov.

Vychádzajúc z funkčných požiadaviek boli popísané jednotlivé procesy, ktoré by mal zabezpečovať navrhovaný systém na podporu rozhodovania v reálnom čase.

Používateľ - dispečer má možnosť zadávať údaje o jednotlivých častiach systému, ako je poloha vlakov, personálu, zmeny grafikonu a stav jednotlivých technológií, v prípade, že on sám toto považuje za potrebné. Aby bola zabezpečená aktuálnosť informácií v rámci systému, bol proces *Zadávanie vstupov vyvolané používateľom* definovaný tak, ako zobrazuje tabuľka 3.1.

Tabuľka 3.1: Proces - zadávanie vstupov vyvolané používateľom

Názov atribútu	Popis atribútu
Názov	Zadávanie vstupov vyvolané používateľom
Popis	Tento proces zabezpečuje možnosť používateľovi vstupovať do fungovania navrhovaného systému, tým že mu umožňuje zadávať jednotlivé údaje potrebné pre zabezpečenie aktuálnosti simulačných modelov.
Vstup	Zadávané údaje
Výstup	Údaje uložené v systéme
Účel	Zabezpečenie aktuálnosti údajov podľa potrieb používateľa
Zdroje	Externé prostredie

V rámci navrhovaného systému môžu nastať situácie, kedy systém sám vyhodnotí potrebu interakcie s používateľom. V takomto prípade je potrebné zabezpečiť zadanie údajov používateľom, avšak táto požiadavka nevznikne na podnet používateľa, ale samotného systému. Toto môže nastať pri problémoch získania údajov z externých zdrojov (informačné systémy v železničnej doprave). Vzhľadom na tieto požiadavky bol definovaný proces popísaný v tabuľke 3.2

Tabuľka 3.2: Proces - zadávanie vstupov vyvolané systémom

Názov atribútu	Popis atribútu
Názov	Zadávanie vstupov vyvolané systémom
Popis	V prípade, že systém sám vyhodnotí, že nemá v externých zdrojov dostatok informácií pre zabezpečenie aktuálnosti, vyvolá požiadavku pre zadanie údajov používateľom.
Vstup	Zadávané údaje
Výstup	Údaje uložené v systéme
Účel	Zabezpečenie aktuálnosti údajov a tým funkčnosti systému
Zdroje	Externé prostredie

Dôležitou výhodou navrhovaného systému je možnosť realizovať mnoho rôznych simulačných behov a tým zistiť, ako by sa pri rôznych vstupných parametroch mohla vyvíjať situácia v budúcnosti. Daný proces je popísaný v tabuľke 3.3.

Tabuľka 3.3: Proces - spúšťanie simulačných behov

Názov atribútu	Popis atribútu
Názov	Spúšťanie simulačných behov
Popis	V rámci tohto procesu je zabezpečená funkcionálnosť, ktorá zabezpečuje realizáciu rôznych simulačných behov. Používateľ nastaví jednotlivé parametre, ako sú dĺžka simulačného behu, počet replikácií, východiskové nastavenie jednotlivých parametrov (napríklad počet personálu, pridelenie koľají a pod.)
Vstup	Parametre simulačného behu
Výstup	Realizovaný simulačný beh
Účel	Navrhovaný systém musí umožniť experimentovanie pomocou viacerých simulačných behov, ktoré majú rôzne vstupné parametre a ukazujú používateľovi, ako by sa situácia v reálnom železničnom systéme mohla vyvíjať.
Zdroje	Používateľ

Aby nebolo rozhodovanie závislé len na dispečerovi, je vhodné využívať rôzne sofistikované algoritmy, napríklad heuristiky, ktoré ponúkajú návrhy riešení. V rámci navrhovaného systému na podporu rozhodovania budú môcť byť využívané rôzne algoritmy a používateľ si bude môcť vybrať medzi týmito algoritmami. Proces umožňujúci využitie sofistikovaných algoritmov je popísaný v tabuľke 3.4.

Tabuľka 3.4: Proces - využívanie sofistikovaných algoritmov

Názov atribútu	Popis atribútu
Názov	Výber sofistikovaných algoritmov
Popis	V navrhovanom systéme používateľ môže využiť sofistikovaný algoritmus, aby získal návrh rozhodnutia, kvalitu tohto návrhu následne overí realizovaním simulačného behu.
Vstup	Údaje zo simulačného modelu
Výstup	Výsledky daného algoritmu
Účel	Aby neboli rozhodnutia plne závislé na dispečerovi, je dobré využívať podporu rôznych sofistikovaných algoritmov (napr. heuristika pre výber vhodnej koľaje)
Zdroje	Používateľ

Počas behu jednotlivých simulačných behov môže nastať situácia, kedy používateľ potrebuje zasiahnuť do simulačného behu a zmeniť určité parametre - napríklad prideliť konkrétnu koľaj konkrétnemu vlaku. Tento proces zobrazuje tabuľka 3.5.

Tabuľka 3.5: Proces - ovplyvnenie simulačného behu

Názov atribútu	Popis atribútu
Názov	Ovplyvnenie simulačného behu
Popis	Hoci používateľ nastavuje parametre simulačného behu na jeho začiatku, v niektorých prípadoch môže vyhodnotiť ako potrebné zasiahnuť do prebiehajúceho simulačného behu a zmeniť niektoré parametre, napríklad priradiť danému vlaku inú koľaj a tým zmeniť konečný výsledok simulačného behu.
Výstup	Zmenený simulačný beh
Účel	Umožnenie experimentovania aj počas experimentálnych simulačných behov, v prípade, že používateľ vyhodnotí zásah do realizovaného behu za potrebný.
Zdroje	Používateľ

Dôležitou súčasťou navrhovaného systému musí byť možnosť vizualizácie sledovaného železničného uzla vo vhodnej forme s jednotlivými detailami, tak aby bolo používateľovi umožnené analyzovať stav systému, pochopiť jeho fungovanie v aktuálnom čase a následne sa vhodne rozhodnúť. Proces vizualizácie stavu systému popisuje tabuľka 3.6.

Tabuľka 3.6: Proces - vizualizácia stavu systému

Názov atribútu	Popis atribútu
Názov	Vizualizácia stavu systému
Popis	Aby mohol používateľ analyzovať sledovaný systém a následne sa správne rozhodnúť, je potrebné, aby mu bola poskytnutá vizualizácia sledovaného systému v aktuálne sledovanom čase.
Vstup	Požiadavka zadaná používateľom
Výstup	Vizualizácia sledovaného systému
Účel	Vizualizácia systému je používateľovi poskytnutá s cieľom umožniť mu lepšie pochopenie fungovania systému.
Zdroje	Simulačný model

Aby mohol používateľ vhodne vyhodnotiť dôsledky svojich rozhodnutí, je potrebné, aby mu bolo umožnené sledovať hodnoty jednotlivých ukazovateľov (napr. celkový čas meškania vlakov) počas simulačného behu. Tento proces je popísaný v tabuľke 3.7.

Tabuľka 3.7: Proces - Získavanie čiastkových výstupov

Názov atribútu	Popis atribútu
Názov	Získavanie čiastkových výstupov
Popis	Aby systém vedel poskytnúť podporu rozhodovania v reálnom čase, je potrebné, aby používateľ mohol sledovať vývoj jednotlivých ukazovateľov už počas simulačného behu a tak lepšie vyhodnotiť dôsledky svojich rozhodnutí.
Vstup	Parametre požadovaného výstupu (napr. typ výstupu - tabuľka/graf, rozsah výstupu, veličina - počet odpracovaných hodín, využitie koľají a mnohé ďalšie).
Výstup	Grafické zobrazenie aktuálneho stavu systému (informácie o aktuálnych štatistikách) v podobe grafov či tabuliek.
Účel	Poskytnutie rôznych výkonnostných parametrov vo forme tabuľky, grafu počas simulačného behu, tak aby používateľ vedel vyhodnotiť stav sledovaného systému v reálnom čase.
Zdroje	Simulačný model

Aby mohol systém poskytovať podporu rozhodovania, je potrebné, aby používateľ mohol získavať údaje o jednotlivých parametroch systému aj na konci simulačného behu vo forme súhrnných štatistík tak, ako to popisuje tabuľka 3.8.

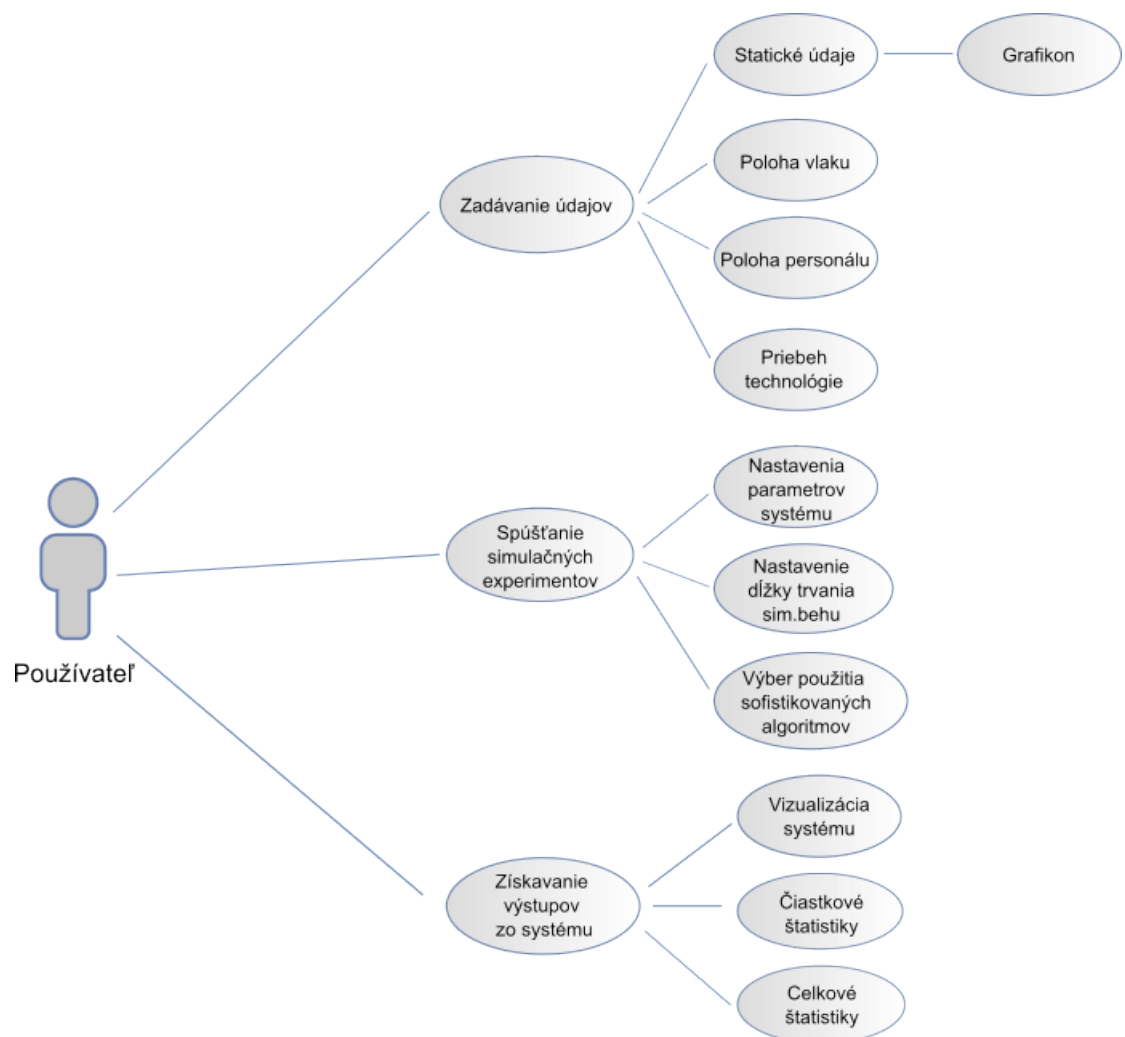
Tabuľka 3.8: Proces - Získavanie celkových výstupov

Názov atribútu	Popis atribútu
Názov	Získavanie celkových výstupov
Popis	Jednou z funkcionalít navrhovaného systému je umožniť používateľovi posúdiť dôsledky svojich budúcich rozhodnutí pomocou simulačných experimentov. Aby bolo možné toto realizovať, je potrebné získať obraz o budúcom možnom vývoji železničného systému pomocou rôznych ukazovateľov, akými sú napríklad pracovné vyťaženie personálu, využitie jednotlivých koľají, počet obslužených vlakov, celkové meškanie a podobne. Tieto ukazovatele sú používateľovi prezentované v prívetivej podobe vo forme tabuľky alebo grafu.
Vstup	Parametre požadovaného výstupu (napr. typ výstupu - tabuľka/graf, rozsah výstupu, veličina - počet odpracovaných hodín, využitie koľají a ďalšie)
Výstup	Tabuľka / graf zobrazujúci požadovaný výstup (napr. štatistika množstva odpracovaných hodín)

Názov atribútu	Popis atribútu
Účel	Poskytnutie rôznych výkonnostných parametrov vo forme tabuľky, grafu, ktoré slúžia k tomu, aby používateľ dokázal posúdiť kvalitu svojich budúcich rozhodnutí.
Zdroje	Simulačný model

Zainteresovanou stranou, hlavným a zároveň jediným aktérom navrhovaného systému je používateľ - dispečer. Jeho kompetencie v rámci navrhovaného systému na podporu rozhodovania v reálnom čase zobrazuje nasledujúci use case diagram zobrazený na obrázku 3.1.

Vychádzajúc z definovaných požiadaviek bola navrhnutá architektúra real-time systému pre podporu rozhodovania dispečerov v železničných dopravných uzloch.

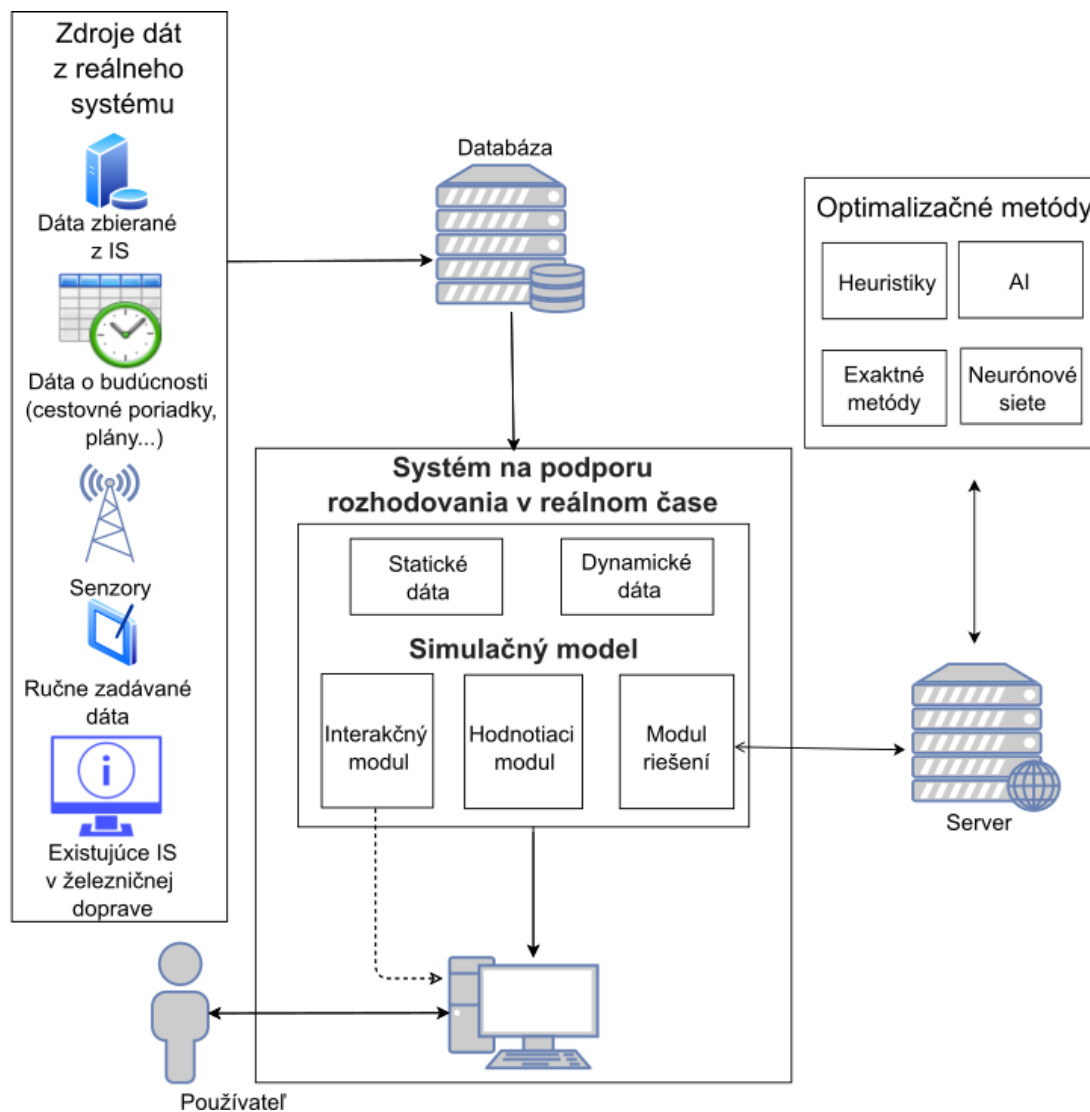


Obr. 3.1: Use case diagram - používateľ

3.2 Popis navrhovaného systému na podporu rozhodovania v reálnom čase

Architektúru navrhovaného systému na podporu rozhodovania zobrazenú na obrázku 3.2 možno rozdeliť do troch základných častí a týmito časťami sú:

- zdroje dát z reálneho systému,
- systém na podporu rozhodovania v reálnom čase,
- optimalizačné metódy.



Obr. 3.2: Architektúra systému na podporu rozhodovania založený na simulačnom modeli

Vzhľadom na definované požiadavky (sledovanie aktuálneho stavu, ako aj budúceho vývoja) je vhodným základom pre navrhovaný systém simulačný model. Bez existencie kvalitných simulačných modelov by nebolo možné navrhovaný systém vôbec realizovať.

Kvalitný simulačný model musí nevyhnutne obsahovať statické a dynamické dáta. Aby boli tieto dáta v požadovanej kvalite a v reálnom čase zahrnuté do simulačného modelu, potrebuje jadro systému tvorené simulačným modelom čerpať dáta z existujúcich zdrojov z reálneho systému železnice, akými sú napríklad dispozičné systémy. Toto spojenie prebieha pomocou databázového servera, kde sú ukladané dáta z existujúcich systémov a pre potreby sledovania aktuálnej situácie pomocou simulačných modelov sú tieto informácie posielané do jadra navrhovaného systému.

Už samotný systém zahŕňajúci všetky výhody simulačných modelov, ktorý čerpá aktuálne informácie z reálneho systému za pomoci existujúcich dispozičných systémov, poskytuje možnosť sledovať budúci vývoj pomocou rôznych vstupov od používateľa poskytuje dispečerom efektívne možnosti podpory rozhodovania. Napriek tomu sa ukazuje ako vhodné zahrnutie ďalšej, nemenej dôležitej funkcionality. Trefou zložkou navrhovanej architektúry sú preto rôzne sofistikované metódy, keďže tieto sú potrebné pre kvalitnejšie návrhy rozhodnutí pre dispečera. Takéto sofistikované algoritmy potrebujú dodávať informácie do jadra navrhovaného systému na základe požiadaviek, vznikajúcich v priebehu realizácie simulačných experimentov. Tieto algoritmy môžu byť súčasťou simulačného modelu, avšak častejšie vstupujú do systému ako externý modul/aplikácia a pripojenie k tomuto modulu je zabezpečované pomocou servera.

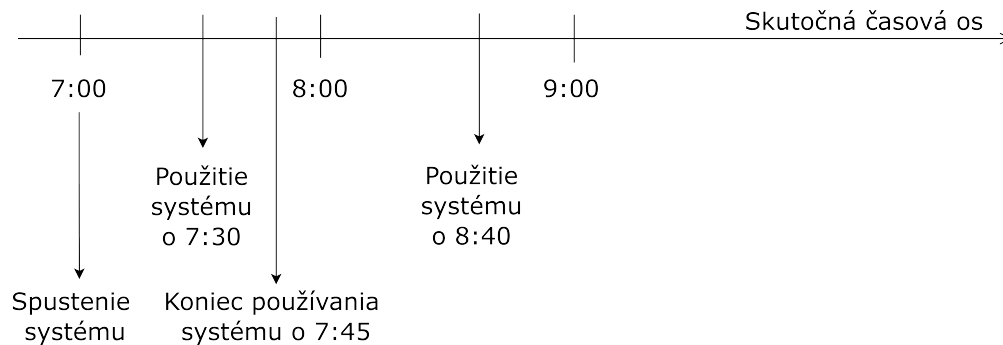
Funkcionality, ktoré boli definované ako potrebné vzhľadom na používateľa možno rozdeliť do troch modulov a to *modul riešení*, *interakčný modul* a *hodnotiaci modul*.

3.2.1 Interakčný modul

V rámci interakčného modulu je zabezpečená interakcia systému s používateľom. Toto zahŕňa nastavenie počiatočnej konfigurácie experimentov, ako aj ich spúšťanie. Pre tým ako používateľ nastaví konfiguráciu experimentov je nutné zabezpečiť aktuálnosť simulačného modelu vzhľadom na realitu. Keďže používateľ môže vykonať viacero experimentov vychádzajúcich z rovnakého počiatočného stavu, je potrebné zabezpečiť uloženie aktuálneho stavu vzhľadom na realitu.

Obrázok 3.3 ukazuje príklad použitia systému na časovej osi. V čase 7:00 bol systém nasadený, to znamená, že dáta z reálneho systému železnice sú odosielané na databázový server a ukladané pre ich opätovné použitie, tento proces prebieha počas celého trvania existencie systému. V čase 7:30 sa dispečer rozhodol použiť systém na podporu rozhodovania. V tomto čase, teda prebehlo zabezpečenie aktuálneho stavu a to tak, že do simulačného jadra boli poslané všetky dáta týkajúce sa času medzi 7:00 a 7:30 a prebehla synchronizácia dát. Po synchronizácii je stav simulačného modelu uložený. V čase 7:30 - 7:45 sú realizované experimenty. To znamená, že pri každom ďalšom experimente sa načíta stav uložený o 7:30. 7:45 dispečer prestáva používať daný systém.

V čase 8:40 sa dispečer znovu rozhodne použiť systém. Zabezpečovať aktuálnosť systému aj v čase, keď s ním používateľ nepracuje by bolo výpočtovo náročné, okrem toho by tento prístup mohol viesť ku mnohým chybám, vzhľadom k náročným opravným mechanizmom v simulačnom modeli pri synchronizácii s reálnym systémom, a preto bol systém navrhnutý tak, že jednotlivé údaje sú načítané z databázy.



Obr. 3.3: Časová os použitia systému

To znamená, že z databázy sú poslané všetky údaje od 7:30 do 8:40, prebehne synchronizácia a používateľ môže realizovať ďalšie experimenty. V priebehu týchto experimentov je dôležité, aby rozhodnutia, ktoré sa v rámci simulačného modelu vykonávajú, boli čo najlepšie. To je problematické, pretože možných variantov riešenia je veľké množstvo a je potrebné klásť dôraz na viaceré ukazovatele, ktoré hovoria o kvalite rozhodnutia. V simulačných modeloch je možné tieto rozhodnutia realizovať interakciou s používateľom. V prípade, že používateľ vidí možnosť zlepšiť spôsob pridelovania zdrojov, simulačný model mu môže umožniť priamo zasiahnuť do vykonávania simulačného experimentu. Tento zásah bude realizovaný takisto pomocou interakčného modulu.

Keďže simulačný model bude súčasťou systému, musí byť vykonávanie simulačných experimentov dostatočne rýchle, preto je základným predpokladom existencie navrhovaného systému existencia efektívne (dostatočne rýchlo) pracujúcich simulačných modelov.

Hlavnou výhodou navrhovaného systému, ktorý využíva simulačný model obsahujúci sofistikované algoritmy (najmä optimalizačné algoritmy), oproti systémom bez napojenia na simulačný model sú práve možnosti simulačného modelu a to prezentácia stavu riešenia v budúcnosti, vizuálne zobrazenie simulovaného systému, grafické či textové zobrazenie výsledkov.

3.2.2 Hodnotiaci modul

Systém založený na využití simulačného modelu musí byť schopný prezentovať všetky relevantné informácie pre rozhodovanie dispečera. Nejedná sa len o jeden izolovaný návrh rozhodnutia, ale simulačný model by mal ponúknuť súbor relevantných štatistických ukazovateľov a vyhodnotení, ktoré predstavujú ukazovatele výkonnosti železničného dopravného uzla - KPI. Tieto ukazovatele môžu byť zobrazené v grafickej podobe či alfanumerickej podobe vo forme grafov, plánov či tabuliek. Výsledky simulačného modelu môžu používateľovi ukázať nakoľko sa jednotlivými preverenými variantami zlepšili sledované ukazovatele (napr. súhrnné meškanie vlakov). Vzhľadom na vlastnosti simulačných modelov je veľkou výhodou, že používateľovi nie sú poskytnuté len súhrnné hodnoty sledovaných ukazovateľov, ale aj priebežné hodnoty počas simulačného behu. Okrem týchto ukazovateľov je možné sledovať zobrazenie simulovaného systému

v reálnom čase s ohľadom na mnohé podrobnosti a používateľ tak môže lepšie analyzovať vzniknutú situáciu. Pre podporu rozhodovania je potrebné, aby systém na podporu rozhodovania poskytoval údaje o systéme pred vznikom konfliktnej situácie, aby sa tejto dalo v budúcnosti predchádzať.

3.2.3 Modul riešení

V snahe zefektívniť (najmä časovo) proces riešenia problémov v železničnom dopravnom uzle je potrebné, aby systém bol sám schopný prinášať rozhodnutia o prideľovaní zdrojov. V súčasnosti sa simulačné modely pri prideľovaní jednotlivých zdrojov zákazníkom často rozhodujú za pomoci jednoduchých metód, napríklad metóda priority list. Takéto metódy však nie sú optimalizačné a neprinášajú riešenia blížiac sa optimálnym. Na hľadanie optimálnych alebo takmer optimálnych riešení je možné použiť mnohé optimalizačné metódy. Toto je možné aj vďaka tomu, že simulačný model stojaci v jadre navrhovaného systému disponuje všetkými potrebnými informáciami o stave reálneho systému. Týmito metódami môžu byť exaktné matematické výpočty, heuristiky, neurónové siete a iné. Napriek tomu, že systém využívajúci optimalizačné metódy dokáže prinášať riešenia sám, bez intervencie používateľa (dispečera), model nemusí správne modelovať všetky podrobné vlastnosti reálneho systému a ani sofistikované algoritmy nedokážu v plnej miere nahradiť ľudský faktor riadenia. Preto je žiadúca možnosť vstupov od používateľa s cieľom vyskúšať rôzne spôsoby riešenia.

3.2.4 Spôsob nasadenia navrhovaného systému

Nasadenie navrhovaného systému môže byť realizované na dvoch úrovniach. V prvej úrovni je systém schopný poskytovať používateľovi možnosť sledovať aktuálny stav reálneho systému a vhodne s ním experimentovať, avšak jednotlivé rozhodnutia sú plne závislé na vstupoch od používateľa. V prípade, že dokážeme zabezpečiť existenciu tejto úrovne, je možné sa zaoberať možnosťami ako čiastočne oslobodiť celý systém od ľudského faktora zapojením sofistikovaných algoritmov. Druhá úroveň navrhovaného systému teda zahŕňa získavanie výstupov rôznych sofistikovaných algoritmov pri realizácii rozhodnutí v rámci jednotlivých simulačných behov.

Pre realizáciu navrhovaného systému je potrebné dôkladne definovať informácie, ktoré sú čerpané z reality a sú potrebné pre udržiavanie aktuálneho stavu simulácie vzhľadom na reálny systém. Je zrejmé, že pri súčasnom fungovaní železničnej dopravy nie je možné poskytnúť všetky informácie, ktoré by simulačný systém potreboval. Vzhľadom k tomu sa v ďalších kapitolách tejto práce zaoberáme analýzou už existujúcich systémov, ako aj možnosťami, ako by mohli byť zabezpečené kvalitnejšie zdroje informácií.

V prípade, že v možných zdrojoch informácií v reálnom systéme železničnej dopravy (informačné systémy, dlhodobé plány, senzory a iné) budú existovať všetky potrebné informácie, je ďalej potrebné sa zaoberať možnosťami získavania takýchto informácií v požadovanej forme. Z tohto vyplýva, že je potrebné zaoberať sa možnosťami implementácie komunikačných kanálov smerom k simulačnému modelu.

Zabezpečenie kvalitného simulačného modelu, ktorý spĺňa všetky naň kladené požiadavky je nemenej dôležité. Existujúce simulačné modely nedisponujú všetkými funkcionalitami potrebnými pre aktualizáciu

stavu vzhľadom na realitu. V tejto oblasti sa vynára mnoho problémov. Je potrebné, aby simulačný model použitý v navrhovanom systéme na podporu rozhodovania v reálnom čase umožňoval realizáciu mnohých experimentov, kde východiskovými parametrami sú parametre odrážajúce aktuálny stav systému. To predpokladá možnosť uchovávať aktuálny stav simulačného modelu. V dátových štruktúrach musí byť možné meniť dáta na základe správ prichádzajúcich z vonkajšieho prostredia. Okrem zmeny dynamických dát je potrebné pamätať aj na možnosť zmeny dát v rámci plánovaných činností. Možnosti riešenia týchto problémov sú podrobnejšie popísané v kapitole 4, ako aj v kapitole 7.4.

4 Metodika tvorby systému na podporu rozhodovania s využitím simulačných modelov

Nasledujúca kapitola popisuje metodický postup, ktorý je potrebné dodržať pri tvorbe systémov na podporu rozhodovania v železničnej doprave v reálnom čase s využitím simulačných modelov. Cieľom tvorby takýchto systémov je poskytnutie podpory dispečerom v ich rozhodovacom procese, pričom tieto systémy využívajú dobré vlastnosti simulačných modelov železničných dopravných uzlov. Takýto systém zabezpečuje dispečerovi zobrazenie aktuálnej situácie v reálnom systéme železnice, sledovanie jednotlivých ukazovateľov a možnosti experimentovania s viacerými variantami budúceho vývoja.

Predpoklady pre realizáciu metodiky

- dostupnosť informácií o železničnom uzle,
- existencia nástrojov vhodných na vytvorenie simulačného modelu železničného uzla,
- možnosť vytvoriť simulačný model železničného uzla,
- zabezpečenie sieťovej komunikácie so simulačným modelom,
- zmluvné podmienky umožňujúce čerpať informácie o vybranom uzle v reálnom čase.

Metodický postup zahŕňa všetky činnosti potrebné na implementáciu systému na podporu rozhodovania v reálnom čase a je možné ho rozdeliť do nasledovných fáz:

- definovanie požiadaviek na systém,
- definovanie dát, ktoré je potrebné čerpať z reálneho systému,
- definovanie komunikačnej architektúry,
- výber softvéru na tvorbu simulačného modelu,
- integrácia spracovania dát z reálneho systému do softvéru na tvorbu simulačného modelu,
- vytvorenie simulačného modelu,
- zabezpečenie ukladania aktuálneho stavu simulačného modelu,
- zabezpečenie prenosu dát,
- overenie funkčnosti navrhovaného systému.

Jednotlivé fázy sú v rámci tejto metodiky podrobnejšie popísané a pre každú z týchto fáz boli definované jednotlivé náležitosti, ktorými sú cieľ, vstup, postup a výstup.

4.1 Fáza 1 - Definovanie požiadaviek na systém

- Cieľ:** V rámci tejto fázy je potrebné detailne definovať jednotlivé požiadavky, ktoré musí vytváraný systém spĺňať, aby mohol slúžiť ako podpora rozhodovania v operatívnom riadení. Tieto požiadavky zahŕňajú aj jednotlivé ukazovatele, ktoré potrebuje dispečer sledovať v simulačnom modeli. Hlavnou funkcionalitou systému na podporu rozhodovania pri operatívnom riadení je zabezpečenie možnosti sledovať aktuálnu situáciu v železničnom dopravnom uzle. Pre realizáciu tohto cieľa je potrebné definovať, aké jednotlivé ukazovatele bude používateľ sledovať. Tieto ukazovatele sa môžu líšiť v závislosti od charakteru železničného dopravného uzla. Zvyčajne sa však jedná o informácie o jednotlivých vlakoch, personáli a technológiách prebiehajúcich na vlakoch.
- Vstup:** Interná dokumentácia práce dispečera, smernice prevádzky železničného uzla.
- Postup:** Tvorca predmetného systému musí na základe analýz konkrétneho dispečerského pracoviska a železničného uzla, ale aj na základe znalostí situácie v železničnej doprave definovať požiadavky na systém. Dôležitým je najmä rozhovor s budúcim používateľom vytváraného systému - dispečerom. Požiadavky sa môžu meniť v závislosti od charakteristík konkrétneho dispečerského pracoviska. Vo všeobecnosti je ale potrebné zabezpečiť sledovanie významných ukazovateľov o vlakoch, personáli a prebiehajúcich činnostiach na vlakoch.
- Výstup:** Dokumentácia požiadaviek na systém, najmä jednotlivých ukazovateľov, ktoré je potrebné sledovať v rámci simulačného modelu.

4.2 Fáza 2 - Definovanie dát, ktoré je potrebné čerpať z reálneho systému

- Cieľ:** Aby mohol byť zabezpečený aktuálny stav simulačného modelu, ktorý je súčasťou systému na podporu rozhodovania v reálnom čase, je potrebné získavať dáta z reálneho systému. V rámci tejto fázy je potrebné realizovať analýzu aktuálnej situácie v železničnom uzle z hľadiska potenciálnych zdrojov relevantných dát, ako aj analýzu dát samotných.
- Vstup:** Dokumentácia informačných systémov, dokumentácia zabezpečovacích zariadení, senzorov a iných možných zdrojov dát.
- Postup:** V rámci tejto fázy je potrebné analyzovať možné zdroje relevantných dát. Toto môžu byť informačné systémy alebo fyzické zariadenia na trati. Je potrebné definovať, ako často sú jednotlivé dáta evidované a nakoľko sú relevantné pre prácu dispečera. Ďalej je dôležité vedieť, ktoré dáta sú potrebné pre správne modelovanie skutočného systému, ale nie je možné ich z poskytnutých zdrojov dát získať.
- Výstup:** Špecifikácia typov dát, ktoré je možné čerpať z reálneho systému.

4.3 Fáza 3 - Definovanie komunikačnej architektúry

- Cieľ:** Aby mohli byť získavané dáta z reálneho prostredia, je potrebné definovať komunikačné rozhranie medzi zdrojmi týchto dát a systémom na podporu rozhodovania v reálnom čase.
- Vstup:** Špecifikácia vyplývajúca z fázy 2, dokumentácia existujúcich riešení sieťovej komunikácie.
- Postup:** V rámci tejto fázy je potrebné definovať komunikačnú architektúru medzi zdrojmi dát a simulačným jadrom. Návrh tejto architektúry zahŕňa návrh komunikačného rozhrania. V rámci definovania tohto rozhrania je potrebné zvoliť spôsob, akým bude komunikácia prebiehať. Takýchto spôsobov je veľké množstvo, napríklad REST, SOAP, web sockety a ďalšie. Vybraná technológia by mala byť schopná zabezpečiť nepretržitú obojsmernú komunikáciu, ktorá je potrebná v systémoch reálneho času. Pri tvorbe návrhu komunikačného rozhrania je dôležitý výber formátu vymieňaných dát. Opäť má tvorca systému na výber viacero formátov, ako napríklad XML alebo JSON. Vybraný formát by mal byť dostatočne komplexný pre popis rôznych dátových typov, mal by byť dobre prenositeľný v rámci sieťovej komunikácie a nemal by byť príliš náročný na pamäťové požiadavky. V rámci vybraného formátu je potrebné definovať presnú štruktúru jednotlivých správ.
- Výstup:** Dokumentácia definujúca komunikačné rozhranie.

4.4 Fáza 4 - Výber softvéru na tvorbu simulačného modelu

- Cieľ:** Simulačný model, ktorý bude tvoriť jadro systému na podporu rozhodovania, musí spĺňať viaceré charakteristiky vyplývajúce z požiadaviek na tento systém (najmä sledovania mnohých mikroskopických ukazovateľov). Vzhľadom na existenciu kvalitných softvérov slúžiacich na tvorbu simulačných modelov je vhodné vytvoriť simulačný model v takomto softvéri. Cieľom tejto fázy je výber vhodného softvéru na základe požiadaviek, ako aj analýza jeho slabých stránok a potrebných rozšírení tak, aby simulačný model mohol slúžiť na podporu rozhodovania.
- Vstup:** Dokumentácia požiadaviek na softvér vyplývajúca z fázy 1.
- Postup:** Na základe analýzy existujúcich softvérov je potrebné vybrať vhodný softvér na tvorbu simulačných modelov. Je dôležité zvážiť skúsenosti s prácou s daným softvérom, nakoľko skúsenosti s daným softvérom môžu značne znížiť čas potrebný na tvorbu simulačného modelu. Predpokladá sa, že softvér bude potrebné modifikovať pre potreby systému na podporu rozhodovania, a preto je potrebné definovať, ktoré požiadavky nie sú splnené.
- Výstup:** Vhodný softvér na tvorbu simulačných modelov, dokumentácia silných a slabých stránok vybraného softvéru.

4.5 Fáza 5 - Integrácia spracovania dát z reálneho systému do softvéru na tvorbu simulačného modelu

Cieľ: Aby mohol simulačný model zobrazovať aktuálny stav, je potrebné definovať a implementovať spôsob, akým budú dáta v reálnom čase integrované do dátových štruktúr simulačného modelu.

Vstup: Definícia typu dát z reálneho systému vyplývajúca z fázy 2.

Postup: Stav simulačných modelov je modifikovaný mnohými vnútornými komponentami systému, ktoré môžu používať rôzne nezávislé algoritmy, či používať rôzny prístup k dátam. Každá časť systému má vlastné kompetencie a oblasť pôsobenia. Aby bola zabezpečená synchronizácia reálneho systému a simulačného modelu, musí byť stav simulačného modelu modifikovaný na základe prichádzajúcich dát. Táto modifikácia prebieha v simulačnom čase 0. Hoci aktuálny beh simulačného modelu prebieha podľa naplánovaného rozvrhu, jednotlivé zmeny musia byť realizované, až keď sú potvrdené pomocou správy z reálneho systému. Keďže je pravdepodobné, že správy z reálneho systému nezahŕňujú všetky informácie potrebné pre konzistentnosť dátovej štruktúry simulačného modelu, je potrebné definovať možnosti riešenia vzniknutých nekonzistentností.

Príchod vlaku - vlak môže v simulačnom modeli vzniknúť, len ak bola prijatá správa o jeho príchode z reálneho systému. Je dôležité povedať, že príchod vlaku neznamena len jeho skutočný príchod z okolia železničného uzla. V rámci simulačného modelu je možné, aby sa vlak objavil na akomkoľvek mieste, pokiaľ sú splnené podmienky, ako dostatočná dĺžka koľaje, na ktorej má daný vlak stáť.

Realizácie technologického postupu na vlaku - na každom vlaku prebiehajú jednotlivé technologické kroky, tieto sú naplánované v rámci tvorby simulačného modelu, avšak v realite často trvajú inú časovú dĺžku a mnohé z nich nie sú v reálnom systéme evidované. Pre každý technologický postup je potrebné definovať, ktoré technologické kroky sú evidované v reálnom systéme. V rámci udržiavania aktuálneho stavu simulácie je potrebné posúvať technológiu na technologický krok, ktorý bol evidovaný v reálnom systéme (správa bola poslaná simulačnému modelu).

Výstup: Zabezpečenie schopnosti meniť stav simulácie na základe správ z reálneho systému.

4.6 Fáza 6 - Vytvorenie simulačného modelu

Cieľ: Vytvorenie konkrétneho simulačného modelu vybraného železničného uzla vo vybranom softvéri.

Vstup: Definované požiadavky, dokumentácia o danom železničnom uzle (grafikon, pracovné rozvrhy, nákresy, podklady o infraštruktúre, atď.).

- Postup:** Vytvorenie simulačného modelu začína vymedzením systému, ktorý sa bude modelovať. Je potrebné, aby simulant podrobne analyzoval a spoznal skutočný systém. Následne je vytvorený konceptuálny model. Ďalej tvorba simulačného modelu zahŕňa zber vstupných dát, ich analýzu a rozhodnutie o spôsobe modelovania jednotlivých vstupov (deterministické, stochastické, na základe rozdelenia pravdepodobnosti). Samotný simulačný model je vytvorený vo vybranom softvéri a je realizovaná verifikácia a validácia simulačného modelu. Verifikáciou sa určuje či model zodpovedá špecifikácii a konceptuálnemu modelu a táto verifikácia môže byť realizovaná viacerými technikami, ako ladenie, kontrola kódu druhou osobou, trasovanie, testy vstupných generátorov, pozorovanie animačného výstupu a iné. Validáciou modelu sa potvrdí, či je model správnou reprezentáciou simulačného modelu a je možné ju realizovať porovnaním s realitou, s iným modelom alebo empirickou validáciou.
- Výstup:** Valídny simulačný model železničného dopravného uzla.

4.7 Fáza 7 - Zabezpečenie ukladania a opätovného načítania aktuálneho stavu simulačného modelu

- Cieľ:** Je potrebné zabezpečiť uloženie a opätovné načítanie aktuálneho stavu simulačného modelu, nakoľko s týmto simulačným modelom používateľ realizuje mnoho experimentov.
- Vstup:** Simulačný model.
- Postup:** Aby bolo možné realizovať sledovanie a experimentovať so simulačným modelom, ktorý zobrazuje aktuálny stav reálneho prostredia, je potrebné zabezpečiť dátové úložisko. Toto úložisko existuje na rovnakom zariadení ako simulačný model. V rámci tohto úložiska je uložený stav simulačného modelu a je možné ju načítať podľa potrieb používateľa.
- Výstup:** Uložený aktuálny stav vo formáte, s ktorým dokáže pracovať simulačný model.

4.8 Fáza 8 - Zabezpečenie prenosu dát

- Cieľ:** Zabezpečenie prenosu dát medzi zdrojmi dát z reálneho prostredia a simulačným softvérom. Táto fáza zahŕňa fyzickú realizáciu sieťovej komunikácie.
- Vstup:** Komunikačné rozhranie definované vo fáze 3.

- Postup:** Pri tvorbe navrhovaného systému je potrebné zabezpečiť komunikáciu medzi simulačným modelom a externými zdrojmi. Aby bola táto komunikácia možná, je potrebné zabezpečiť komunikáciu podľa návrhu, vytvoreného vo fáze 3. To znamená, že je potrebné rozšíriť simulačný model o komponenty zabezpečujúce sieťovú komunikáciu. Následne je potrebné, aby tieto komponenty dokázali komunikovať na základe definovaného protokolu (posielať správy v požadovanom formáte). Táto komunikácia musí byť obojsmerná, takže komunikačné komponenty musia byť schopné prijímať správy zo siete. Následne je potrebné zabezpečiť komunikačnú bránu a to realizáciou infraštruktúry v podobe serverov, databáz, webových služieb a iných. Tvorca systému musí byť schopný zabezpečiť bezpečnosť, spoľahlivosť a výkonnosť systému. V navrhovanom systéme prebieha komunikácia z už existujúcimi externými aplikáciami. Pri týchto aplikáciách je potrebné zabezpečiť možnosť pripojenia na základe protokolu, ktorý dokážu implementovať. Ak hovoríme o externých aplikáciách, ktoré sú vytvárané až v rámci tvorby systému na podporu rozhodovania, je dôležité, aby tieto tak isto dokázali nadviazať komunikáciu podľa definovaného protokolu. Po nasadení celej infraštruktúry je dôležitým krokom testovanie a ladenie systému.
- Výstup:** Realizácia sieťovej komunikácie.

4.9 Fáza 9 - Overenie funkčnosti navrhovaného systému

- Cieľ:** Funkčnosť nasadeného systému na podporu rozhodovania je potrebné overiť, pred tým, ako bude poskytnutý používateľovi. Toto overenie je realizované sledovaním výstupných hodnôt, ktoré pomáhajú určiť či systém pracuje podľa požiadaviek.
- Vstup:** Systém na podporu rozhodovania.
- Postup:** Pre overenie funkčnosti navrhovaného systému na podporu rozhodovania, je potrebné sledovanie viacerých vybraných ukazovateľov, akými sú napríklad poloha vlakov, celkové meškania, časy príchodu vlakov do uzla a mnohé iné. Je potrebné sledovať, či v aktuálnom čase simulačný model reflektuje reálny stav, to znamená, či sledované ukazovatele skutočne nadobúdajú hodnoty ako v reálnom systéme.
- Výstup:** Dokumentácia sledovaných parametrov, potvrdzujúca správne fungovanie systému.

5 Spôsohy získavania aktuálnych informácií o stave železničného dopravného uzla

Aby bolo možné vytvoriť navrhovaný systém na podporu rozhodovania, je potrebné zabezpečiť aktuálnosť stavu simulačného modelu, ktorý tvorí jadro tohto systému. Zabezpečenie aktuálneho stavu v simulačnom modeli predpokladá získavanie informácií z reálneho systému. Aby to bolo možné zabezpečiť, musia byť dôkladne špecifikované informácie, ktoré simulačný model k svojej funkčnosti potrebuje. Okrem toho musia byť definované zdroje takýchto informácií a spôsob, akým je možné takéto informácie získavať a vhodne doručiť do simulačného modelu.

5.1 Dáta využívané v systéme na podporu rozhodovania

Simulačný model by mal v každom okamihu odrážať aktuálny stav modelovaného systému. Je teda nevyhnutné zabezpečiť jeho aktualizáciu na základe spojenia s reálnym systémom využitím existujúcich informačných systémov. Informácie potrebné pre tvorbu modelu je možné rozdeliť na statické a dynamické.

5.1.1 Statické informácie

Statické informácie predstavujú infraštruktúru modelu a nie sú závislé od aktuálneho stavu systému. Takéto informácie je možné rozdeliť do mnohých kategórií, pričom základnými sú infraštruktúra - koľajisko, personál, katalóg vozňov a hnacích koľajových vozidiel, technologické postupy, zoznam cieľových a tranzitných vlakov, zoznam vozňov a cestovných poriadkov, zoznam východiskových vlakov a ďalšie.

Infraštruktúra je tvorená údajmi o presne definovanej polohe jednotlivých koľají určenej súradnicami a dĺžkou týchto koľají, v ideálnom prípade aj s vlastnosťami georeliéfu. Ďalej sú to definície jednotlivých traťových úsekov, oddielov či skupín a priradenie koľají týmto úsekom. Je potrebné definovať, ktoré koľaje sa používajú pre vchody vlakov. Pri definovaní infraštruktúry je potrebné vedieť, kde sa nachádzajú odstavné koľaje, parkovacie miesta lokomotív, rôzne jednoduché a križovatkové výhybky, návestidlá, koľajové križovatky a podobne.

Ďalšími statickými informáciami sú informácie o počte zdrojov a ich funkcií, pričom dôležité sú najmä informácie o personáli. Je potrebné poznať jednotlivé profesie, ktoré pracujú v konkrétnom dopravnom uzle. Dôležitou profesiou v súvislosti so železničnou dopravou sú samozrejme rušňovodiči, avšak môžu to byť aj vozmajstri, posunovači, hradlári, výhybkári, tranzitéri, skladníci, personál zabezpečujúci údržbu vlakov či rôzny pomocný personál. Je dobré poznať aj konkrétne kompetencie jednotlivých profesií, na-

príklad vozmajstra vo vchodovej skupine. Medzi dôležité informácie patria počty tohto personálu, ale aj jeho pracovné zmeny.

V rámci informácií, ktoré nie sú závislé od aktuálneho času sú definované aj jednotlivé vlaky, rušne a vozne. Vlak je definovaný vozňami a rušňom, ďalej plánovaným časom príchodu a odchodu zo železničného uzla. V rámci rôznych železničných staníc je ďalej potrebné poznať činnosti vykonávané v týchto staniach. Presne definovaná postupnosť týchto činností tvorí technologický postup, v rámci ktorého sú určené časy trvania jednotlivých činností, ako aj personál potrebný na ich vykonanie.

Statickými informáciami, ktoré poznáme vopred sú aj jednotlivé technologické postupy práce s vlakom. Tieto sú tvorené jednotlivými technologickými úkonmi, pričom technologický postup môže byť tvorený aj desiatkami technologických úkonov. Technologickým úkonom je napríklad "úplná skúška brzdy". Jednotlivé technologické úkony majú definovaný aj personál, ktorý je potrebný na ich realizáciu.

5.1.2 Dynamické informácie

Druhým typom informácií, ktoré sú potrebné pre fungovanie systému na podporu rozhodovania sú dynamické informácie. Tieto informácie sú na rozdiel od statických informácií vytvárané v reálnom čase. Takýmito informáciami je najmä aktuálna poloha vlaku, hnacieho motorového vozidla, či konkrétnych vozňov. Pokiaľ hovoríme o železničných dopravných uzloch, je dôležitým typom dynamickej informácie stav prebiehajúcej technológie vlaku. Ďalej je to poloha personálu, ako aj stav, v ktorom sa aktuálne nachádza či práca, ktorú vykonávajú. Okrem toho to môže byť aktuálny časový rozvrh príchodov, obsluhy a odchodov vlakov, ktorý sa môže výrazne líšiť od plánovaného.

5.2 Možnosti získavania informácií

Pre zabezpečenie fungovania navrhovaného systému je teda potrebné určiť možnosti získavania statických a dynamických informácií.

5.2.1 Statické informácie

Všetky statické informácie (informácie o cestovných poriadkoch, pracovných rozvrhoch a infraštruktúre) je možné pripraviť vopred.

Informácie o cestovných poriadkoch

Významným zdrojom informácií o plánovaných cestovných poriadkoch je najmä Prevádzkový informačný systém (PIS) popísaný v kapitole 2.2.2. PIS poskytuje možnosti získavania dát pomocou webových služieb (zvyčajne pomocou SOAP protokolu), pričom komunikačné rozhranie je definované presne pre potreby konkrétneho systému, na základe charakteru dát, ktoré tento systém potrebuje. Tieto webové služby umožňujú prístup k rôznym typom informácií o cestovných poriadkoch, vrátane plánovaných spojení, odchodov a príchodov vlakov, meškaní a ďalších relevantných údajov.

Informácie o pracovných rozvrhoch

Dobrym zdrojom informácií o pracovných rozvrhoch je informačný systém VICOS. Na Slovensku však pracovné zmeny často nie sú plánované pomocou systému VICOS, ale skôr v interných systémoch jednotlivých uzlov, ako je napríklad systém SAP. Systém SAP má definované komunikačné rozhranie využívajúce webové služby používajúce formát JavaScript Object Notation (JSON) (ale aj iné formáty).

Informácie o infraštruktúre

Informácie o infraštruktúre sa získavajú transformáciou mapových podkladov v papierovej alebo elektronickej podobe vytvorenej napr. programom AutoCAD. Väčšina železničných dopravných uzlov má v rámci svojich interných systémov prístup k takejto elektronickej (digitálnej) podobe infraštruktúry, pričom tieto interné systémy zahŕňajú rôzne platformy geografických informačných systémoch. Keďže sa jedná o statické dáta, ktoré je možné pripraviť vopred, tvorca systému by sa mal interne oboznámiť s týmito podkladmi, prípadne detailné informácie získať priamo od vedenia/personálu príslušného železničného uzla.

Napriek tomu, že statické informácie sa nezvyknú meniť, v prípade cestovných poriadkov, ale aj pracovných zmien môže v určitých prípadoch ísť k zmene. Túto zmenu je takisto možné získať z vyššie spomenutých systémov, avšak pri návrhu komunikácie je potrebné pamätať aj na takéto situácie.

5.2.2 Dynamické informácie

Dynamické informácie je potrebné získavať v reálnom čase a niektoré z nich sa nachádzajú v PIS, prípadne v GTN (viď kapitola 2.2.1).

Informácie o polohe vlaku

Z týchto systémov je možné získať najmä informácie o polohe vlaku na trati, v prípade PIS aj o konkrétnej polohe v rámci uzla. Do týchto systémov prichádzajú informácie aj priamo z trate, z rôznych senzorov, GPS lokátorov, RFID lokátorov, prípadne rádiovkej komunikácie a je teda možné informácie zbierať aj priamo z týchto zdrojov. Takéto zariadenia zvyčajne odosielať informácie na server (v prípade systému PIS ide o Tomcat server), kde sa tieto dáta o polohe vlakov agregujú a tento server je schopný ich následne rozposielať klientom v požadovanom formáte. Pokiaľ hovoríme o dátach o vlakoch, systém PIS ich poskytuje štandardne v XML formáte a obsahujú údaje ako typ vozňa, číslo vozňa, hmotnosť a dĺžka vozňa, režim brzdenia, špeciálne vlastnosti (nebezpečný tovar, nesmie ísť cez zväzky pahorok a podobne). Kontrola integrity dát poskytovaných PIS je definovaná na strane systémov požadujúcich informácie, čo znamená, že takáto kontrola musí byť zahrnutá aj v navrhovanom systéme na podporu rozhodovania.

Informácie o polohe a stave personálu

V porovnaní so zberom informácií o vlaku sa viac problematickým sa ukazuje zber údajov o polohe a stave personálu, nakoľko tieto nie sú takmer vôbec zaznamenávané do informačných systémov, ktoré sú používané v železničných uzloch. Hoci sú v rámci železničných dopravných uzlov používané systémy pre riadenie personálu, tieto systémy sa používajú na evidencie jednotlivých pracovných zmien, prítomnosti zamestnanca v práci a podobne (napríklad systém EDOS). Poloha zamestnanca, prípadne konkrétna práca vykonávaná v aktuálnom čase sa v týchto systémoch nevyskytuje. Poloha personálu je však dôležitým ukazovateľom v rozhodovacom procese dispečera. Ak dispečer vo svojom rozhodovacom procese potrebuje informácie o polohe personálu, zvyčajne ju zisťuje pomocou vysielacky. Toto však môže byť veľmi neefektívne. Pre efektívnu podporu rozhodovania s využitím simulačných modelov sa preto odporúča zaviesť automatické zisťovanie polohy personálu, prípadne môžu byť tieto informácie zadávané ručne dispečerom (čo však nie je veľmi efektívne). Informácie o polohe personálu je však možné zbierať aj pomocou rôznych technológií ako sú GPS lokátory, bluetooth zariadenia, vysielacky či s využitím moderných zariadení, akými sú smart hodinky.

GPS lokátory sú prenosné zariadenia, ktoré umožňujú monitorovanie osôb alebo vozidiel v železničnej doprave. Tieto zariadenia prijímajú GPS signál a na základe prijatých informácií vypočítajú súradnice určujúce ich polohu. Je preto potrebné, aby bolo používané v oblastiach pokrytých GPS signálom. Výhodou GPS lokátorov, ktoré môžu byť zabudované do smart hodínok alebo do vysielacky je, že pre svoje správne fungovanie nepotrebujú žiadne ďalšie hardvérové zariadenia a pomocou softvérovej aplikácie zabudovanej do smart hodínok sú informácie zasielané na server, odkiaľ ich môže získavať akýkoľvek ďalší systém. Nevýhodou je ich nižšia presnosť a to najmä v interiéri, keďže pre správne fungovanie GPS lokátora je potrebné zabezpečiť kontakt so satelitom. Táto nepresnosť sa v interiéri môže pohybovať až okolo 100 metrov. Znížená presnosť je možné zaznamenať aj v exteriéri a to najmä za nepriaznivého počasia. Okrem toho sa na železničnej trati často nachádzajú oblasti (najmä tunely), kde použitie GPS lokátora vôbec nie je možné. V neposlednom rade je potrebné spomenúť možnosť kybernetických útokov, ktoré môžu rušiť alebo manipulovať signály GPS, čo vedie k nepresným alebo falošným údajom o polohe.

Niektoré smart hodinky využívajú namiesto GPS lokátora mobilnú sieť, avšak určovanie polohy pomocou mobilnej siete je rovnako (prípadne menej) presné ako GPS.

Na rozdiel od GPS, bluetooth zariadenia môžu byť najmä v interiéri lepším riešením. Funkčnosť takéhoto riešenia však požaduje zabudovanie bluetooth majákov do priestoru železničného dopravného uzla. Tieto majáky totiž vysielajú signál, ktorý bluetooth zariadenie prijíma a na základe neho určuje svoju polohu. Pre správne fungovanie zisťovania polohy pomocou bluetooth zariadenia musí byť sieť týchto majákov dostatočne hustá (každých 15-50 metrov).

Radio Frequency IDentification (RFID) sa v súčasnosti používa na identifikáciu a sledovanie v mnohých odvetviach. Zisťovanie polohy pomocou RFID tagov je vhodným riešením v oblasti vlakovej dopravy a využíva sa pri určovaní polohy vozňov. Pre určovanie polohy jednotlivých vozňov sú RFID tagy pri-

pevnené k trati v určitých intervaloch. Keď vlak prechádza cez značky, palubná čítačka zhromažďuje informácie o polohe vozňov a prenáša ich do centrálného riadiaceho systému. Tento systém potom použije informácie na určenie polohy a rýchlosti vlaku a môže podľa toho upraviť návestidlá a výhybky. Technológia RFID je vhodná aj pre určenie polohy personálu, aj keď predpokladá existenciu RFID brán. V prípade, že zamestnanec prejde takouto bránou, systém to zaznamená a teda je možné získať údaj, že zamestnanec sa v danom čase nachádzal pri tejto bráne. Výhodou RFID technológie je jej vysoká presnosť aj za nepriaznivých podmienok okolia. Jej nevýhodou je však nevyhnutnosť inštalácie jednotlivých RFID tagov a RFID brán. [61].

Ďalšie možnosti pri určovaní polohy jednotlivých zamestnancov môžu vychádzať z biometrických systémov na rozpoznávanie hlasu. Takéto systémy sú zabudované do vysieláčiek, ktoré personál na trati používa. Keďže moderné vysielacky zväčša disponujú GPS lokátorom, je možné spojiť konkrétnu osobu (pomocou rozpoznania hlasu) s polohou vysielacky. V prípade biometrických systémov však treba upozorniť, že ich presnosť je značne závislá na rôznych fyzikálnych javoch (v prípade rozpoznávania hlasu to môže byť prítomnosť šumu). Hoci sú všetky tieto technológie známe a ich použitie sa javí ako vhodné, v súčasnosti väčšina železničných dopravných uzlov takými zariadeniami nedisponuje.

Informácie o činnosti prebiehajúcej na vlaku

Podobné problémy, aké nastávajú pri zisťovaní polohy personálu súvisia aj s informáciami o činnosti prebiehajúcej na vlaku. V niektorých prípadoch sú takéto informácie hlásené do PIS, avšak nie všetky a najmä nie dostatočne detailne. Získavanie týchto informácií je možné najmä prostredníctvom komunikácie s personálom na trati, ako sú strojvodcovia alebo pracovníci údržby. Takáto forma komunikácie však nie je vždy efektívna, najmä ak ide o rýchle rozhodovanie v reálnom čase. Riešením by mohlo byť monitorovanie železničného dopravného uzla pomocou senzorov, ktoré by zachytili pohyb vlaku v rámci uzla (napríklad ak sa vlak nachádza na koľaji, kde prebieha tankovanie, môžeme predpokladať, že na vlaku práve prebieha aktivita tankovanie). Iným prístupom by mohlo byť získavanie údajov z kamerových záznamov, toto však predpokladá zabezpečenie takéhoto záznamu a samozrejme implementáciu algoritmov, ktoré by dokázali vhodne spracovať takéto obrazové dáta.

Hoci tento postup nie je ideálny a získavanie informácií automatizovaným spôsobom by bolo efektívnejšie, v rámci navrhovaného systému na podporu rozhodovania navrhujeme možnosť upravovať dynamické dáta používateľom-dispečerom, v prípade, že on takýmito informáciami disponuje.

5.3 Komunikačné štandardy

Pri zabezpečení prenosu dát medzi existujúcimi informačnými systémami a navrhovaným systémom na podporu rozhodovania, je potrebné klást dôraz na dodržiavanie komunikačných štandardov. V rámci projektu OPTIMA bola vytvorená špecifikácia inovatívnej komunikačnej platformy, ktorá pomocou štandardizovaných dátových štruktúr umožní efektívny manažment výmeny dát medzi rôznymi službami-

/klientmi. Na definovanie dátových štruktúr a na zabezpečenie integrácie starších aj budúcich aplikácií do jednej komunikačnej štruktúry sa použije koncepčný dátový model. Základným komponentom komunikačnej platformy je integračná vrstva, ktorá umožňuje integrovaný a automatizovaný proces výmeny údajov a ich trvalú dostupnosť v reálnom čase, podporuje viacero klientov a umožňuje bezproblémovú a dynamickú výmenu dát medzi rôznymi službami. [9]

Aby bol možný zber, presun a uchovávanie týchto dát sú v železničnej doprave používané viaceré dátové štandardy:

- railML - voľne dostupný formát založený na XML štandardoch, ktorý bol vytvorený, aby bola zabezpečená interoperabilita medzi jednotlivými aplikáciami v železničnej doprave.
- railML2 - rozšírenie railML. Tento formát je tvorený tromi subschémami a to subschémy pre infraštruktúru, cestovný poriadok, železničné koľajové vozidlá. V rámci rozšírenia railML3 bola pridaná subschéma pre rušne.
- RailTopoModel/railML3 - objektový logický model na štandardizáciu reprezentácie údajov týkajúcich sa železničnej infraštruktúry.
- Register of Infrastructure (RINF) model - nástroj pre statický popis charakteristík a možností železničnej siete.
- Infraštruktúra pre priestorové informácie v Európskom spoločenstve - Infrastructure for Spatial Information in Europe (INSPIRE).
- sensorML - súčasť súboru otvorených štandardov OGC (Open Geospatial Consortium). Definuje XML modely pre opis senzorov, aktuátorov a procesov. [6]
- Sensor Web Enablement (SWE) framework - ďalší zo štandardov OGC, ktorý umožňuje, aby všetky typy snímačov, prevodníkov a dátových úložísk zo snímačov mohli byť nájdené, prístupné a použiteľné prostredníctvom webu. [62]
- Semantic Web for Earth and Environmental Terminology (SWEET) - je databázou, ktorá poskytuje veľké množstvo údajov o ortogonálnych pojmoch (priestor, čas, fyzikálne veličiny, atď.) aj integračných pojmoch vedeckých poznatkov (javy, udalosti, atď.) a tieto údaje sú popísané pomocou webového ontologického jazyka (OWL). OWL je jazyk určený na reprezentáciu bohatých a komplexných znalostí o veciach, skupinách vecí a vzťahoch medzi vecami.
- Semantic Sensor Network (SSN) - slúži na modelovanie rôznych senzorových zariadení, systémov, procesov a pozorovaní získaných z týchto zariadení.

Na základe analýz bol v rámci projektu OPTIMA navrhnutý hybridný prístup využívajúci railML (na popis statických prvkov) a OGC/sensorML (na popis dynamických prvkov).

Ďalšími dátovými štandardmi, ktoré sú používané v železničnej doprave sú:

- Service interface for real-time information (SIRI) - poskytuje framework na špecifikáciu komunikačných protokolov a protokolov na výmenu údajov v reálnom čase týkajúcich sa prevádzky verejnej dopravy. SIRI je navrhnutý ako modulárny a rozširiteľný štandard, ktorý využíva schému XSD.

- Public transport - Network and Timetable Exchange (NeTEx) - vznikol s cieľom uľahčiť interoperabilitu medzi IT systémami zavedením jednotnej architektúry vymieňaných správ. Poskytuje štandard pre výmenu správ o cestovných poriadkoch verejnej dopravy vrátane zastávok, trás, časov odchodu/frekvencií, prevádzkových poznámok a súradníc na mape, trás so zložitou topológiou, dňoch, v ktorých sa služby prevádzkujú, vrátane dostupnosti počas štátnych sviatkov a iných výnimiek, štruktúry cestovného zahŕňajúce paušálne cestovné, cestovné z bodu do bodu, zónové cestovné, tarifné produkty zahŕňajúce jednorazové lístky, spiatočné lístky, denné a predplatné lístky, tarify, ktoré sa uplatňujú len v určitých termínoch a mnohé ďalšie. NeTEx bol vyvinutý ako doplnkový formát k SIRI. [56]
- General Transit Feed Specification (GTFS) - definuje otvorený formát údajov pre cestovné poriadky verejnej dopravy a súvisiace geografické informácie, ktoré sa vzťahujú na rôzne druhy dopravy. Údaje GTFS sa vymieňajú vo forme súborov CSV. GTFS-RT je rozšírením GTFS a umožňuje prenos dát v reálnom čase vo forme binárnych súborov. Kombinácia GTFS a GTFS-RT umožňuje poskytovať údaje o všetkých trasách a vozidlách naraz s použitím minima bitov.
- OpenRailwayMap - je podrobná online mapa svetovej železničnej infraštruktúry, vytvorená na základe údajov OpenStreetMap
- viaceré ontológie z oblasti železničnej dopravy, ako napr. NEPTUNE, Railway Core Ontology, Semantic Railway Infrastructure Ontology a ďalšie.[8]

5.4 Definovanie komunikačného rozhrania

Hoci väčšina štandardov používaných v železničnej doprave vychádza z formátu XML, pre systém navrhovaný v tejto práci sa nejaví tento štandard ako vhodný, nakoľko sa jedná o podporu operatívneho riadenia, čo predpokladá rýchly prenos väčšieho množstva dát v krátkom čase. Syntax jazyka XML však spôsobuje, že tieto súbory sú zbytočne veľké a majú teda vysoké nároky na pamäť. Okrem toho je spracovanie týchto štruktúr (parsovanie) aj výpočtovo náročné. Vzhľadom k tomu sme sa rozhodli použiť najmä na prenos dynamických dát formát JSON.

JSON (JavaScript Object Notation) je dátový formát nezávislý na počítačovej platforme a je vhodný na prenos dát, ktoré sú organizované v podobe objektov. Výhodou tohto formátu je dobrá čitateľnosť pre človeka, ale najmä malá veľkosť prenášaných dát, čo je v prípade zabezpečovania podpory v reálnom čase dôležitým faktorom. Bežne sa používa na prenos údajov vo webových aplikáciách (napr. odoslanie niektorých údajov zo servera klientovi alebo naopak).

Pre realizovanie efektívnej komunikácie bolo potrebné definovať formálnu štruktúru - schému JSON správ, ktoré budú posielané medzi jednotlivými časťami navrhovaného systému na podporu rozhodovania. Všetky správy majú definované ID v celočíselnej podobe a ďalej konkrétnym obsahom v časti *"text"*.

```
{
  "id": "integer",
  "text": "String",
  "timeStamp": "long"
}
```

"Text" je tvorený informáciami, ktoré sa týkajú jednotlivých vlakov, personálu, technologických postupov, grafikonu, pracovných zmien personálu.

Správa o vlaku:

```
{
  "id": "Number",
  "name": "String",
  "coordinateX": "Number",
  "coordinateY": "Number",
  "trainLength": "Number",
  "plannedArrivalTime": "Number",
  "realArrivalTime": "Number",
  "plannedDepartureTime": "Number",
  "realDepartureTime": "Number"
}
```

Správa o personáli:

```
{
  "id": "Number",
  "name": "String",
  "coordinateX": "Number",
  "coordinateY": "Number",
  "currentJob": "String"
}
```

Správa o technológii:

```
{
  "id": "Number",
  "name": "String",
  "trainName": "String",
  "personnelName": "String",
  "startTime": "String",
  "duration": "Number"
}
```

Správa o pracovnej zmene jedného zamestnanca:

```
{  
  "id": "Number",  
  "name": "String",  
  "shiftStart": "String",  
  "shiftEnd": "String"  
}
```

Správa o zmene grafikonu:

```
{  
  "trainName": "String",  
  "originalArrivalTime": "String",  
  "newArrivalTime": "String",  
}
```

Aby mohla byť nadviazaná takáto komunikácia, je potrebné zabezpečiť, že existujúce zdroje informácií (dispozičné systémy, GPS-lokátory, zabezpečovacie zariadenia a pod.) budú schopné posielat informácie v dohodnutom formáte. Rovnako tak simulačné modely, musia byť schopné takéto informácie prijímať a spracovať. Je preto potrebné analyzovať možnosti existujúcich softvérov slúžiacich na tvorbu simulačných modelov. Tieto možnosti sú popísané v nasledujúcej kapitole.

6 Simulačné modely železničných dopravných systémov

Vzhľadom na požiadavky na navrhovaný systém na podporu rozhodovania môže byť počítačová simulácia so všetkými svojimi vlastnosťami dobrým základom takéhoto systému.

Simulácia je výskumná technika, ktorej podstatou je náhrada skúmaného dynamického systému (originálu) jeho simulujúcim systémom (simulačným modelom), s ktorým sa experimentuje s cieľom získať informácie o pôvodnom dynamickom systéme.[36] Počítačová simulácia je širokou oblasťou, zahŕňajúcou realizáciu mnohých typov simulačných modelov, avšak v prípadoch, ktorými sa zaoberá táto dizertačná práca sa jedná o simulačné modely obslužných dynamických systémov, akými sú železničné dopravné uzly. Obslužným systémom rozumieme systém, v ktorom s pomocou určitých aktivít sú požiadavky na vstupe (vlak má prísť na určenú koľaj) zmenené na požadovaný výstup (vlak sa nachádza na určenej koľaji). V rámci prevádzky obslužných systémov je dôležitá najmä správa obmedzených zdrojov, ktorými musia byť uspokojené jednotlivé požiadavky. Tento systém je dynamický, nakoľko nemôžeme zanedbať faktor času.

Simulácia sa využíva na skúmanie systémov, ktorých vlastnosti nie je možné zistiť inak a to z viacerých dôvodov. V prípade simulácií železničných dopravných uzlov by experimenty s reálnym systémom narušili prevádzku celého systému a mohli by ohroziť bezpečnosť. Okrem toho sú takéto experimenty časovo aj finančne náročné.

Keďže železničný dopravný uzol je komplexným systémom a je náročné ho modifikovať, často dochádza k situáciám, v ktorých sú simulačné modely dobre využiteľné. Takýmito situáciami môže byť plánovaná rozsiahla rekonštrukcia, ako napríklad pridanie nových koľají, ale aj menšie zmeny, napríklad plánovaná zmena počtu personálu alebo zmena množstva obsluhovaných vlakov.

6.1 Softvér pre tvorbu simulačných modelov

Softvér pre tvorbu simulačných modelov železničných dopravných uzlov zabezpečuje vytvorenie simulačného modelu v používateľsky prívetivom prostredí. Požiadavky na simulačné modely sa môžu značne líšiť, napríklad v miere detailnosti, a preto existuje široká škála softvérov, z ktorých je potrebné zvoliť taký, ktorý spĺňa konkrétne požiadavky na simulačný model. Dôležitou požiadavkou sú najmä dostatočne detailné modely. Okrem toho je potrebné príjemné grafické používateľské rozhranie, možnosť nastavovania vstupných parametrov simulačných behov a výstupné štatistiky. Vzhľadom na požiadavky, ktorú sú definované pre potrebu použitia simulačného modelu v jadre navrhovaného systému na podporu rozhodovania, sa ako vhodné javia viaceré simulačné softvéry, a to OpenTrack, RailSys, Rail Traffic Controller

alebo Villon.

6.1.1 Simulácia železničnej prevádzky - OpenTrack

Nástroj pre simuláciu železničnej dopravy OpenTrack podporuje modelovanie a simuláciu mnohých zložitých dopravných systémov, akými sú diaľkové aj prímestské železnice, nákladné železnice, metro, trolejbusová doprava a iné, avšak jeho hlavným zameraním je práve tvorba modelov železničnej dopravy a to tak železničných dopravných uzlov, ale najmä medzistaničných úsekov. OpenTrack umožňuje používateľovi analyzovať kapacitu tratí, vytvoriť grafikon, analyzovať vplyv porúch na celkovú prevádzku železničnej trate a ďalšie. Vďaka svojim možnostiam je používaný viacerými železničnými podnikmi, správcami infraštruktúry, ale aj univerzitami, kde poskytuje podporu pri výskumnej činnosti.

V rámci simulačných modelov vytvorených v simulačnom softvéri OpenTrack sú používateľovi poskytované informácie o cestovných poriadkoch, plánovanom aj skutočnom čase príchodu a odchodu jednotlivých vlakov, ako aj o rôznych technických charakteristikách vlaku a jednotlivých vozňov. Simulačné modely vytvorené v tomto softvéri disponujú informáciami o trati, ktoré sú zaznamenané pomocou špeciálneho dvojvrcholového grafu.

Tento softvér poskytuje výstupné informácie vo viacerých bežných formátoch ako napríklad ASCII alebo XML. Zároveň podporuje formát RailML.

OpenTrack ponúka rozhranie OpenTrack API, ktoré umožňuje pripojenie externých aplikácií. Technicky je toto API realizované prostredníctvom štandardizovaných SOAP správ posielaných protokolom HTTP. Externé aplikácie môžu byť použité na implementáciu rôznych používateľských postupov pri riadení dopravy alebo na vývoj nových riadiacich postupov pomocou optimalizácie využitia zdrojov. [57]

Simulačné modely vytvorené v nástroji Opentrack sa nejavia ako vhodné na použitie v rámci systému na podporu rozhodovania v železničnom dopravnom uzle a to najmä preto, že sú primárne zamerané na železničnú trať. S tým súvisí, že neposkytujú žiadne možnosti monitorovania personálu a značne obmedzené možnosti monitorovania prebiehajúcich činností na vlaku, čo je nevyhnutné pre použitie simulačných modelov v navrhovanom systéme na podporu rozhodovania.

6.1.2 RailSys

RailSys je komerčný softvérový systém vyvinutý spoločnosťou RMCon, ktorý sa používa na analýzu, plánovanie a optimalizáciu prevádzky v železničných dopravných sieťach. Tento softvér je tvorený komplexom viacerých modulov a je postavený na mikroskopicknej simulácii, ktorá je zobrazovaná používateľovi a umožňuje mu optimalizovať prevádzkové postupy v železničnej doprave.

Tento softvér umožňuje detailný popis infraštruktúry modelu, zahŕňajúci údaje o koľajách, výhybkách, umiestnení návestidiel, rýchlosti, pomere sklonu trate a iné, a to aj v spojení so systémom ETCS. Okrem údajov o infraštruktúre je pre potreby realizácie simulácie možné definovať rôzne prevádzkové údaje, napríklad typ vlaku a jeho vlastnosti. Pomocou interaktívnej funkcionality umožňuje jednoduché zostavovanie cestovných poriadkov, pričom má zabudované algoritmy na riešenie konfliktných situácií a pomocou počí-

tačovej simulácie je testovaná aj robustnosť navrhovaného poriadku. Hodnotenie fungovania simulovaného systému je možné realizovať pomocou rôznych výstupných grafov a tabuliek.

Softvérový systém RailSys bol úspešne použitý pri tvorbe simulačných modelov viacerých železničných systémov a to napríklad vysokorýchlostnej trati Koln-Rhein/Main, vysokorýchlostnej trati v Sydney Canberra, mestskej železnici v Mníchove a v Kolíne nad Rýnom.[60]

Tento softvér sa zaoberá modelovaním najmä vysokorýchlostných tratí a nedisponuje takými možnosťami modelovania priebehu prevádzky v železničnom uzle, aké sú potrebné v rámci navrhovaného systému na podporu rozhodovania. Rovnako tak jeho možnosti zobrazovania stavu v simulačnom modeli v graficky prívetivej podobe sú značne obmedzené.

6.1.3 Rail Traffic Controller (RTC)

Rail Traffic Controller (RTC) od spoločnosti Berkeley Simulation Software je softvérový balík pre simuláciu železničnej dopravy. V tomto simulačnom softvéri je možné modelovať rôzne typy železničných systémov, od malého terminálu, po komplexnú medzinárodnú sieť. Tento softvér simuluje pohyb vlakov v železničných sieťach na detailnej a realistickej úrovni. Používa sa na rôzne účely, od taktického zlepšenia dopravného toku až po určenie, na čo by sa mali vynaložiť kapitálové prostriedky na infraštruktúru.

Integrovaná kalkulačka výkonnosti vlaku (Train Performance Calculator-TPC) zohľadňuje rôzne typy zariadení, zostáv vlakov, rôzny terén a podmienky trate. Tento simulačný softvér do určitej miery umožňuje aj riešenie konfliktných situácií, a to tak, že čas príchodu vlakov je možné zmeniť, avšak iba v rámci časového obmedzenia, ktoré nastavil používateľ. Podobne ako iné simulačné softvéry, tento softvér umožňuje prezentovať výsledky v používateľsky prívetivej podobe a poskytuje rôzne typy výstupov, napríklad vo forme grafov. [66]

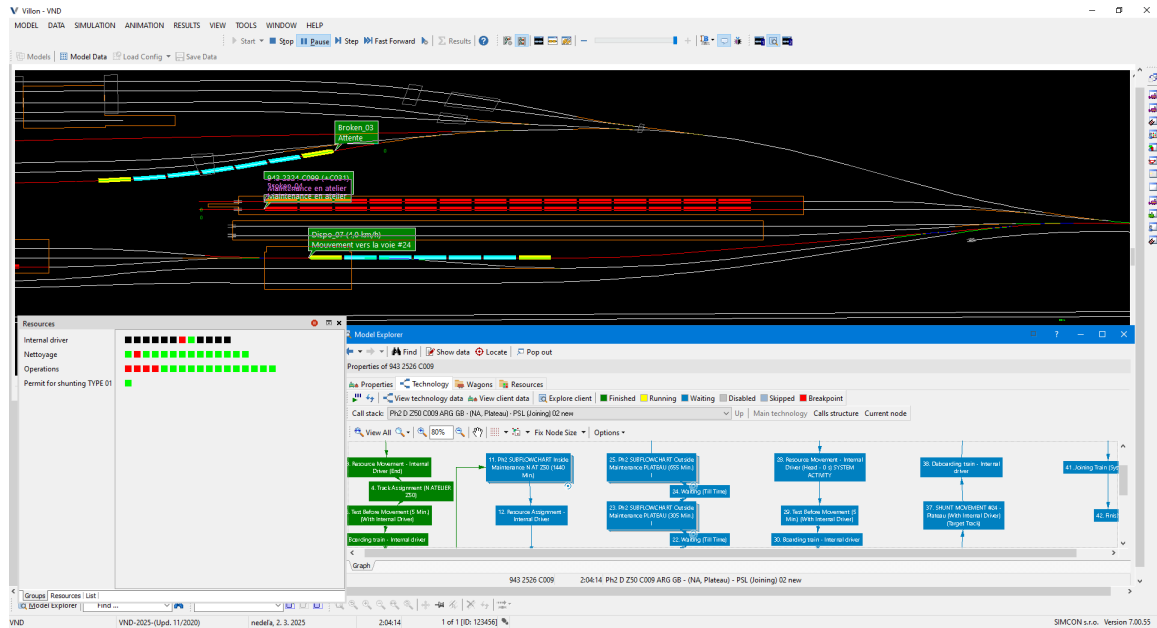
Hoci sa tento simulačný softvér detailne zaoberá modelovaním železničných systémov, zameriava sa viac na problémy v rámci celej siete a nedostatočne modeluje jednotlivé procesy prebiehajúce v uzle, ako sú najmä obslužné procesy na vlakoch.

6.1.4 Simulačný softvér Villon

Simulačný softvér Villon [2] sa používa na vytváranie univerzálnych simulačných modelov rôznych typov dopravných uzlov železničnej alebo cestnej infraštruktúry, akými sú napríklad zriaďovacie stanice, osobné a nákladné stanice, kontajnerové terminály, depá atď. Okrem toho podporuje aj modelovanie menších dopravných sietí, napr. regionálnej železničnej dopravy. Tento simulačný softvér sa zameriava na tvorbu mikroskopických simulačných modelov, umožňuje definovať rôzne simulačné scenáre, vykonávať experimenty a následne hodnotiť výsledky týchto experimentov, pričom tieto výsledky sú dodávané v podobe prívetivej pre používateľa (grafy, tabuľky). Jeho hlavným účelom je v súčasnosti poskytovanie podpory pri dlhodobom a strednodobom plánovaní.

Z používateľského hľadiska poskytuje Villon editory na úpravu potrebných vstupných údajov, poskytuje 2D a 3D animáciu, hodnotenie simulačných experimentov vo forme postsimulačných štatistík

v textovej aj grafickej podobe (vo forme rôznych tabuliek a grafov).



Obr. 6.1: Model v simulačnom softvéri Villon

Tento simulačný softvér je založený na agentovo-orientovanej architektúre ABAsim. V simulačnom softvéri Villon je modelovaný systém považovaný za obslužný systém a je rozdelený do troch subsystémov: systém zdrojov, systém zákazníkov a systém riadenia. Súčasťou systému zdrojov je model infraštruktúry, ktorý je založený na presnom geometrickom pláne alebo schematickom modeli a pozostáva z koľají, výhybiek, križovatiek a ďalších komponentov. Okrem infraštruktúry sú v systéme zdrojov definované aj mobilné zdroje ako sú lokomotívy alebo personál. Systém zákazníkov obsahuje definíciu príchodu zákazníkov (vlakov) do systému a aj ich požadovaný postup obsluhy. Obsluha je vykonávaná v rámci systému riadenia a je definovaná technologickým postupom určeným používateľom. Celý životný cyklus jednotlivých vlakov v simulačnom modeli je definovaný týmto technologickým postupom (ďalej len technológiou) a táto technológia pozostáva z technologických krokov, označených ako aktivita. Alokácia jednotlivých zdrojov, potrebných na realizovanie obsluhy sa väčšinou realizuje pomocou algoritmu, ktorý vyberá zo zoznamu zdrojov, buď pomocou metódy „prvý voľný“ alebo podľa nastavenej priority.

Simulačné modely vytvorené v simulačnom softvéri Villon boli použité vo viacerých simulačných štúdiách, a to napríklad pri zlepšovaní prevádzky zriaďovacích staníc v Rakúsku (Viedeň, Linz), Nemecku (Hamburg), Švajčiarsku (Basel) a v Číne (Mudanjiang), ako aj pri rekonštrukcii údržbových centier v Ulme alebo v Paríži a mnohých ďalších železničných uzloch.

Základnými požiadavkami, ktoré by mal spĺňať simulačný model stojaci v jadre systému na podporu rozhodovania v železničných dopravných uzloch v reálnom čase sú:

- detailné zobrazenie aktuálnej situácie v simulačnom modeli vzhľadom na skutočnosť,
- možnosti rôznej konfigurácie experimentov,

- zasahovanie do simulačného behu používateľom (prípadne výsledkami sofistikovaných algoritmov),
- získavanie výstupov vo forme tabuliek alebo grafov,
- zameriavanie na simulácie v železničných dopravných uzloch.

Simulačný softvér Villon spĺňa viaceré tieto požiadavky a má implementované mnohé funkcionality, ktoré sú pre železničné dopravné uzly špecifické, preto sa simulačné modely vytvorené v tomto softvéri javia ako vhodné pre použitie v navrhovanom systéme na podporu rozhodovania.

Napriek tomu, že simulačné modely vytvorené v tomto softvéri spĺňajú časť definovaných požiadaviek, v aktuálnej podobe nie sú schopné aktualizovať stav simulácie na základe informácií z existujúcich dispozičných systémov, pretože nie sú schopné takéto informácie prijímať a spracovať. Rovnako nie je implementovaná možnosť, aby používateľ zasahoval do simulačného behu počas jeho realizácie a menil tak stav simulačného modelu v danom okamihu. V rozšírenej podobe navrhovaného systému (podoba využívajúca sofistikované algoritmy) je potrebné zabezpečiť aj integráciu výsledkov týchto algoritmov do prebiehajúceho simulačného behu, čo v súčasnosti takisto nie je možné.

7 Rozšírenie architektúry ABAsim

Navrhovaný systém na podporu rozhodovania popísaný v tejto práci využíva simulačné modely, ktoré sú postavené na agentovo-orientovanej architektúre ABAsim[1].

7.1 Agentovo orientovaná simulácia - ABAsim

Agentovo orientovaná simulácia je jednou z viacerých paradigiem (systémová dynamika, udalostná simulácia a iné) tvorby simulačných modelov. V rámci agentovo orientovanej simulácie sa modelovaný systém dekomponuje na jednotlivých autonómnych agentov, pričom sa definuje ich správanie a vzťahy medzi nimi. Agent je zapuzdrený počítačový systém zasadený do nejakého prostredia, ktorý v ňom pružne a autonómne pôsobí za účelom plnenia daného cieľa [74]. Dôležitou vlastnosťou agenta je autonómnosť. To znamená, že agent je schopný pracovať samostatne bez vonkajších zásahov, riadiť svoju činnosť a kontrolovať vnútorný stav. Okrem autonómnosti je agent schopný spoločenského správania, vďaka ktorému prebieha interakcia s ostatnými agentami prostredníctvom komunikačného mechanizmu. Agent sa dokáže správať reaktívne (reaguje na podnety prichádzajúce z jeho okolia), ale aj proaktívne (cielené, z vlastnej iniciatívy). Tvorbu agentovo orientovaných modelov umožňuje architektúra ABAsim.

Architektúra ABAsim je agentovo orientovaná architektúra vyvinutá kolektívom autorov na Fakulte riadenia a informatiky Žilinskej univerzity v Žiline, ktorú je možné použiť na modelovanie zložitých obslužných systémov a obsahuje prvky, ktoré uľahčujú tvorbu flexibilných simulačných modelov. Táto architektúra sa skladá z jednotlivých komponentov, ktoré disponujú rôznymi schopnosťami. Týmito komponentami sú agenti (riadiaci a dynamickí) a entity, pričom základným komponentom sú práve agenti. Pomocou agenta je možné modelovať autonómny prvok systému (človek, vlak, inštitúcia). Ďalším komponentom je entita, teda prvok, ktorý nemá vlastnú iniciatívu (koľaj).

Každý agent má definované funkcie a cieľ, ktorý by mal plniť, resp. splniť. Tento cieľ môže mať agent vopred definovaný, alebo si ho sám definuje v priebehu simulácie. V architektúre ABAsim sú definované dva typy agentov a to riadiaci a dynamickí agenti.

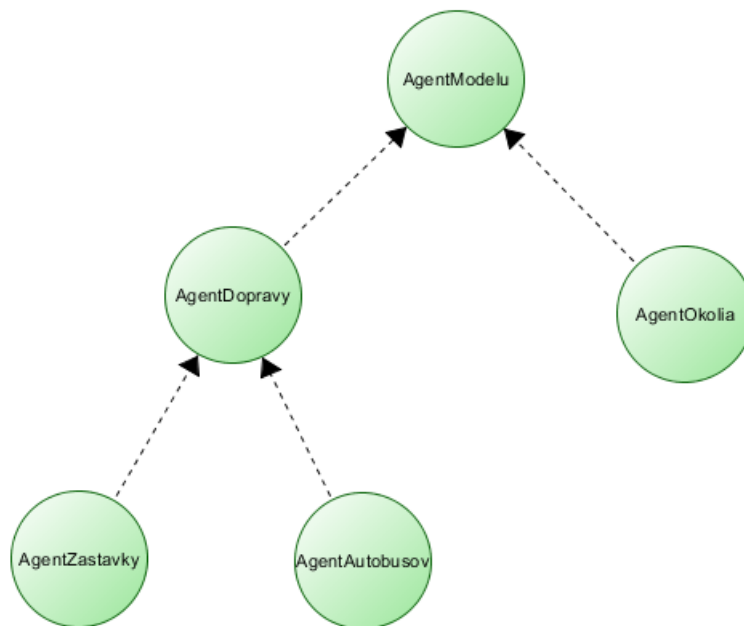
Riadiaci agenti sú organizovaní v hierarchickej štruktúre (obr. 7.1), ktorá modeluje vzájomné väzby medzi nimi a títo agenti majú jeden spoločný, systémový cieľ (obslúženie zákazníka), ktorý je dosahovaný ich vzájomnou spoluprácou.

V rámci hierarchickej povahy architektúry je nadradený agent zodpovedný za činnosť všetkých svojich podradených agentov a zároveň vystupuje ako ich zástupca voči ostatným riadiacim agentom. Okrem podradených riadiacich agentov riadi riadiaci agent aj dočasné skupiny dynamických agentov. Dynamickí agenti môžu v rámci simulácie vznikať a zanikať. Zvyčajne ich používame na modelovanie inteligentných

prvkov systému.

V architektúre ABAsim nie je agent atomickým prvkom a skladá sa z viacerých komponentov. Riadiacim komponentom agenta je manažér. Manažér koordinuje komunikáciu s ostatnými komponentami a aj s okolím a tým riadi činnosť agenta. Dôležitými internými komponentami sú asistenti. Títo sa delia na senzory (sprístupňujú informácie o stave prostredia), riešiteľov (poskytujú manažérovi návrhy na spôsob riešenia), efektory (vykonávajú zmenu stavu systému). Na základe doby trvania činnosti jednotlivých asistentov, ich môžeme rozdeliť na:

- kontinuálnych asistentov - ich činnosť trvá určitý nenulový čas,
- promptných asistentov - ich činnosť nemá časové trvanie, vykonáva sa v jednom okamihu.



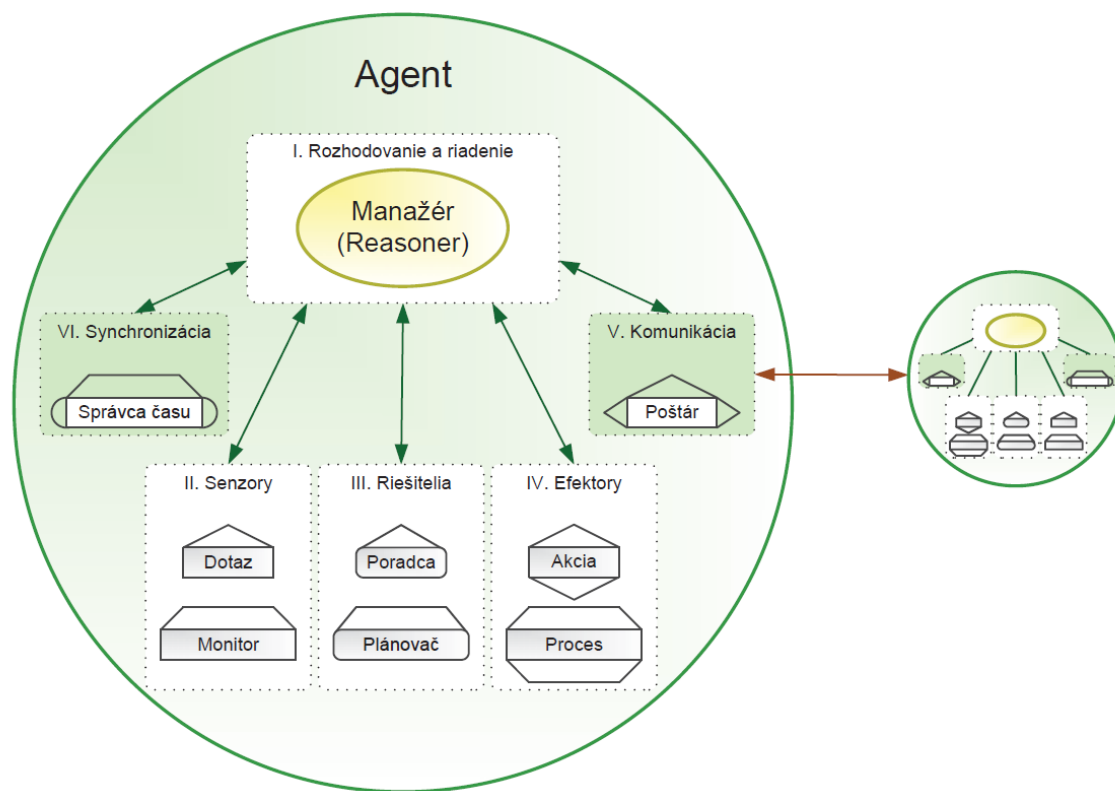
Obr. 7.1: Hierarchická štruktúra agentov [1]

Okrem týchto komponentov má každý agent implementované dva komponenty a to správcu času (zabezpečuje kauzálnu synchronizáciu agentov) a poštára (zabezpečuje komunikáciu). Jednotlivé komponenty agenta sú zobrazené na obrázku 7.2.

Komunikačný mechanizmus je zabezpečovaný posielaním správ medzi jednotlivými prvkami architektúry. Komunikácia môže prebiehať tromi spôsobmi:

- agenti medzi sebou,
- manažér a asistent,
- kontinuálni asistenti medzi sebou počas svojej činnosti.

Dôležitá je najmä komunikácia medzi agentmi. Táto komunikácia musí spĺňať rôzne funkcie, a preto je zabezpečovaná viacerými typmi správ:



Obr. 7.2: Dekompozícia agenta [1]

- *Notice* (oznámenie) - správa, na ktorú sa neočakáva žiadna odpoveď,
- *Request* (žiadost') - správa, obsahujúca určitú požiadavku a jej odosielateľ očakáva od adresáta odpoveď,
- *Response* (odozva) - odpoveď na správu typu *request* a môže byť doručená len jej odosielateľovi,
- *Handover* (odovzdanie) - odovzdanie dynamického agenta do pôsobnosti iného agenta,
- *Entrust* (zverenie) - dočasné zverenie dynamického agenta do pôsobnosti iného agenta,
- *Return* (vrátenie) - vrátenie dynamického agenta do pôsobnosti pôvodného riadiaceho agenta,
- *Done* (hotovo) - správa, ktorou dynamický agent oznamuje, že splnil svoj cieľ,
- *Goal* (cieľ) - správa, ktorou nadriadený riadiaci agent určuje cieľ dynamickému agentovi,
- *Quit* (koniec) - správa, ktorou nadriadený agent ruší ciele podriadeného dynamického agenta,
- *Transfer* (presun) - správa, ktorou dynamický agent požaduje presun pod iného agenta.

Nemenej dôležité je zabezpečiť komunikáciu medzi manažérom a jeho jednotlivými asistentami a toto je realizované pomocou nasledovných typov správ:

- *Start* - po prijatí tejto správy začne kontinuálny asistent svoju činnosť.
- *Break* - po prijatí tejto správy je činnosť kontinuálneho asistenta ukončená.
- *Execute* - táto správa je určená promptnému asistentovi, ktorý po jej prijatí okamžite vykoná príslušnú činnosť a obratom pošle správu svojmu manažérovi.

Kontinuálni asistenti komunikujú viacerými typmi správ:

- *Finish* - je správa, ktorú pošle kontinuálny asistent svojmu manažérovi, keď skončí svoju činnosť.
- *Notice* - je správa, ktorou asistent upozorňuje svojho manažéra o určitej dôležitej situácii.
- *Hold* - táto správa obsahuje časovú pečiatku, ktorá určuje, kedy má byť správa doručená a tento typ správy posielal asistent sám sebe.

Na tomto mieste je dôležité spomenúť aj spôsob adresovania správ. Správy môžu byť:

- adresné, čiže priamo adresované agentovi, ktorý vie správu spracovať;
- čiastočne adresné správy, kedy sú správy adresované nadriadenému agentovi;
- neadresné správy, ktoré nemajú určeného adresáta.

Každý agent si udržiava vlastný register správ, ktoré dokáže spracovať on alebo jeho podriadení agenti. [1]

Aby bolo možné zabezpečiť komunikáciu tak, ako je popísané na obrázku 3.2, bolo potrebné realizovať určité zmeny v tejto architektúre, nakoľko táto architektúra nedisponovala dostatočne rozvinutou podporou komunikácie s tretími stranami.

7.2 Návrh komunikačnej architektúry

Vzhľadom na systém navrhnutý v tejto práci je dôležité si uvedomiť, že proces, ktorý zabezpečuje dátový tok súvisiaci s podporou rozhodovania sa skladá z dvoch základných podprocesov a to:

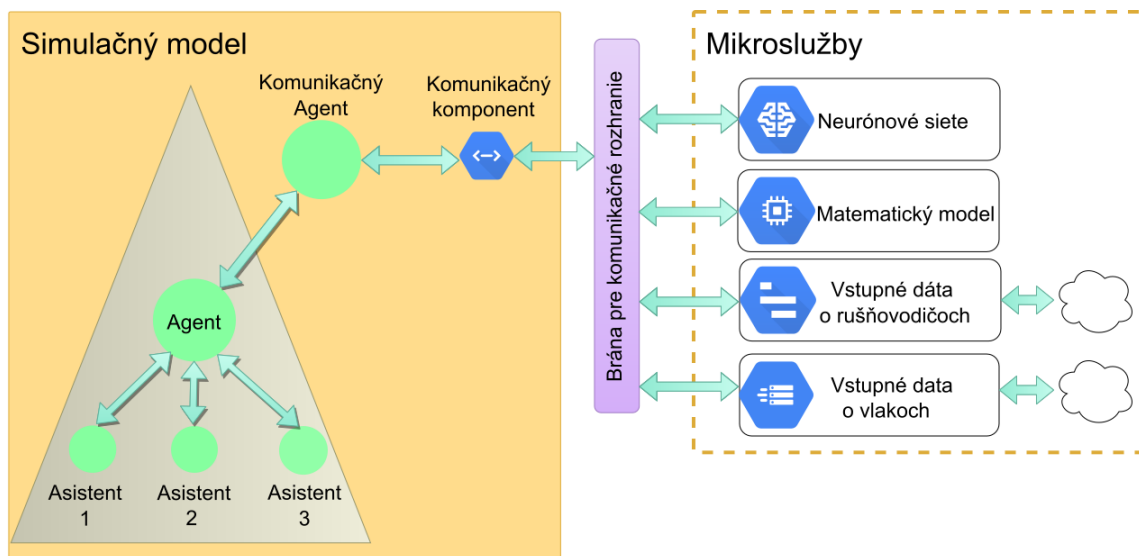
- procesu, ktorý zabezpečuje aktuálnosť stavu simulácie vo vzťahu ku realite a v rámci neho je teda zabezpečovaný prenos vstupných informácií vo vopred definovanom rozsahu a forme do simulačného modelu;
- procesu, ktorý zabezpečuje použitie sofistikovaných algoritmov a v rámci neho je zabezpečovaný prenos výstupných požiadaviek zo simulačného modelu a následne vstupných informácií obsahujúcimi odpoveď, ktorú poskytla sofistikovaná metóda.

V súvislosti s dátovým tokom je teda potrebné zabezpečiť, aby bolo možné zmeniť stav systému, aby bol aktuálny vo vzťahu ku reálnemu systému. V súčasnosti zmeny stavu zabezpečujú asistenti - efektori. Týmto asistentom je preto potrebné zabezpečiť spojenie s existujúcimi informačnými systémami. Druhou skupinou vstupných informácií sú informácie súvisiace s realizáciou rozhodnutí (napr. efektívne priradenie personálu). Takéto rozhodnutia sú realizované v asistentoch - riešiteľoch, títo asistenti majú implementované rozhodovacie algoritmy. V nami navrhovanom systéme však môže byť rozhodovacím algoritmom aj externý algoritmus (napríklad neurónová sieť) alebo používateľ. Je preto potrebné zabezpečiť komunikačný kanál medzi asistentom - riešiteľom a týmto externým komponentom (algoritmus/ používateľ).

Výstupnými informáciami vzhľadom na návrh architektúry systému na podporu rozhodovania sú informácie, ktoré potrebuje podporný algoritmus, aby mohol poskytnúť návrh riešenia, to znamená, že asistent - riešiteľ potrebuje byť schopný odosielať informácie smerom k podpornému algoritmu. V rámci asistenta

- riešiteľa je teda potrebné zabezpečiť obojstrannú komunikáciu. Podporné algoritmy môžu byť z rôznych dôvodov implementované ako externé moduly. Týmito dôvodmi môže byť lepšia podpora v inom programovacom jazyku, optimalizačná metóda implementovaná v minulosti bez toho, že by sa predpokladala možnosť ju integrovať do simulačného modelu a podobne. V mnohých prípadoch je však možné takúto sofistikovanú metódu implementovať priamo v simulačnom modeli. Tento postup je odporúčaný najmä pre možnosť zdieľať celý stavový priestor v rámci jedného dátového úložiska a teda nie je potrebné zabezpečiť prenos veľkého množstva dát po sieti k externým riešiteľom.

Vzhľadom na agentovo-orientovanú povahu architektúry ABAsim budú jednotlivé sofistikované metódy implementované v jednotlivých asistentoch agenta a použitie konkrétneho asistenta bude realizované na základe konfiguračných nastavení pred začiatkom simulačného behu. Ak bude sofistikovaná metóda realizovaná mimo simulačný model (napríklad v riešiteľ FICO xpress), asistent zabezpečujúci túto metódu vhodne spracuje dáta, ktoré táto metóda potrebuje. Tieto dáta bude následne potrebné poslať externe implementovanej metóde. Toto bude realizované, poslaním špecifickej správy agentovi, ktorého súčasťou je daný asistent. Tento agent odošle správu komunikačnému agentovi, ktorý je zahrnutý do komunikačnej architektúry, tak ako je zobrazené na obrázku 7.3.



Obr. 7.3: Návrh komunikačnej architektúry

Predtým ako bude popísaná presná implementácia v rámci ABAsim architektúry, je potrebné popísať celý návrh komunikačnej architektúry. Koncept architektúry postavenej na mikroslužbách [31][15][70] vychádza zo *Service Oriented Architecture* (SOA) - architektúra orientovaná na služby. SOA je tvorená princípmi a metodológiami, ktoré odporúčajú skladať zložitejšie aplikácie zo skupín na sebe nezávislých komponentov. Taktýmto komponentami sú v prípade takejto architektúry mikroslužby. Mohlo by sa zdať, že sa jedná o jednu z vrstiev monolitckej aplikácie (napr. web controllera), ale nie je to tak. Mikroslužba je vytvorená tak, aby dokázala zabezpečiť jednu funkčnosť, má jasne definované rozhranie a nemusí nevy-

hnutne implementovať viac vrstiev. Takýto prístup javí ako dobre použiteľný v rámci nami navrhovaného systému na podporu rozhodovania.

Výhodami architektúry postavenej na mikroslužbách sú najmä:

- modulárnosť: aplikácia je oveľa zrozumiteľnejšia, vývoj a testovanie sú jednoduchšie;
- škálovateľnosť: mikroslužby sú implementované a nasadzované nezávisle na sebe (bežia na nezávislých procesoch) a preto sa dajú škálovať nezávisle;
- integrácia heterogénnych a zastaralých systémov: je možné nahradiť časti existujúcich systémov mikroslužbami;
- distribuovaný rozvoj: umožňuje paralelný vývoj jednotlivých služieb nezávisle na sebe.

Architektúra postavená na mikroslužbách má samozrejme aj niekoľko nevýhod:

- služby vytvárajú informačné bariéry, keď jedna služba čaká na spracovanie inej služby;
- vzájomné volanie medzi mikroslužbami kladie vyššie nároky na sieť;
- vzhľadom na distribuovaný rozvoj môže byť nasadenie komplikovanejšie;
- vývoj môže byť náročnejší vzhľadom na izolovanosť jednotlivých mikroslužieb (môžu byť vyvíjané v rôznych jazykoch a podobne).

Jednotlivé mikroslužby sú teda izolovanými službami, ktoré na základne požiadavky (vstup) poskytujú odpoveď (výstup). Príkladom mikroslužby je napríklad výpočet na základe matematického modelu, pomocou ktorého sa určí, ktorého rušňovodiča je vhodné prideliť k danej úlohe.

V rámci nami navrhovanej architektúry systému na podporu rozhodovania (vzhľadom na jeho distribuovanosť po sieti) bolo potrebné navrhnuť komunikačný protokol pre komunikáciu po sieti. Takýchto protokolov existuje veľké množstvo, známymi sú najmä SOAP, REST, JSON-RPC, XML-RPC, WS-RPC, gRPC a ďalšie.

SOAP (Simple Object Access Protocol) je protokolom pre výmenu správ založených na formáte XML. Štandardné požiadavky a odpovede rozhrania SOAP API sú realizované cez správy vo forme obálky a tieto správy pozostávajú zo štyroch prvkov so špecifickými funkciami pre každý z nich. Týmito prvkami sú obálka, hlavička, telo a chyba. Obálka je základným a podstatným prvkom každej správy. Správa začína a končí značkami obálky, ktoré ju obklopujú. Hlavička (nepovinná) určuje špecifiká, dodatočné požiadavky na správu, napr. autentifikáciu. Telo obsahuje žiadosť alebo odpoveď. Chyba (nepovinná) obsahuje všetky údaje o prípadných chybách, ktoré by sa mohli objaviť v priebehu žiadosti a odpovede API. Jednou z hlavných vlastností rozhraní SOAP API je, že takmer vždy používajú WSDL dokument. Zjednodušene povedané, dokument WSDL je opis webovej služby vo formáte XML, ktorý funguje ako návod na komunikáciu s webovou službou, definuje koncové body a opisuje všetky procesy, ktoré by mohli vykonávať pripojené aplikácie. Môžu zahŕňať dátové typy používané v správach SOAP a všetky akcie dostupné prostredníctvom webovej služby. Súbor WSDL teda slúži ako podpísaná zmluva medzi klientom a serverom. SOAP podporuje viacero prenosových protokolov ako HTTP, HTTPS, Transmission Control Protocol (TCP), Simple Mail Transfer Protocol (SMTP), File Transfer Protocol (FTP) a iné.

Komunikácia po sieti môže byť realizovaná pomocou stavových alebo bezstavových operácií. Pri použití stavových operácií server uchováva informácie, ktoré dostane od klienta v rámci viacerých požiadaviek, čo môže byť kľúčové pri realizácii komplikovaných služieb. Bezstavová komunikácia znamená, že každá správa obsahuje dostatok informácií o stave klienta, takže server sa tým nemusí zaoberať. Keď server vráti požadované údaje, na klienta zabudne. Každá požiadavka je izolovaná od predchádzajúcej. Bezstavové operácie znižujú zaťaženie servera a zvyšujú rýchlosť komunikácie. SOAP podporuje oba typy operácií. Hlavnými nevýhodami tohto protokolu je nízka rýchlosť a nedostatočná flexibilita. [35]

REST je architektonický štýl, ktorý definuje súbor obmedzení zameraných na zlepšenie výkonnosti, dostupnosti a škálovateľnosti webových distribuovaných systémov [18]. RESTful služby musia mať bezstavové správanie, t. j. server neuchováva informácie o požiadavkách odoslaných klientmi. Toto obmedzenie je nevyhnutné na zlepšenie škálovateľnosti systému. Bezstavové správanie si vyžaduje samopopisné správy, t. j. požiadavka musí obsahovať všetky požadované informácie na spracovanie serverom [63]. V rámci REST je najčastejšie používaný prenosový protokol HTTP, ale je možné využiť aj iné protokoly.

JSON-RPC je odlahčený protokol vzdialeného volania procedúr (Remote Procedure Call), ktorý používa správy vo formáte JSON. Správa s požiadavkou JSON-RPC môže obsahovať tri možné prvky:

- metóda - refazec, ktorý pomenúva metódu, ktorá sa má vyvolať;
- params - objekty alebo polia hodnôt, ktoré sa odovzdávajú ako parametre cieľovej aplikácii;
- id - refazec alebo číslo, ktoré zodpovedá odpovedi s požiadavkou, na ktorú sa odpovedá.

Tento protokol je využívaný najmä pre svoju jednoduchosť, čo je však zároveň aj nevýhodou, keďže nemusí pokryť všetky špecifikácie konkrétneho typu komunikácie. [54]

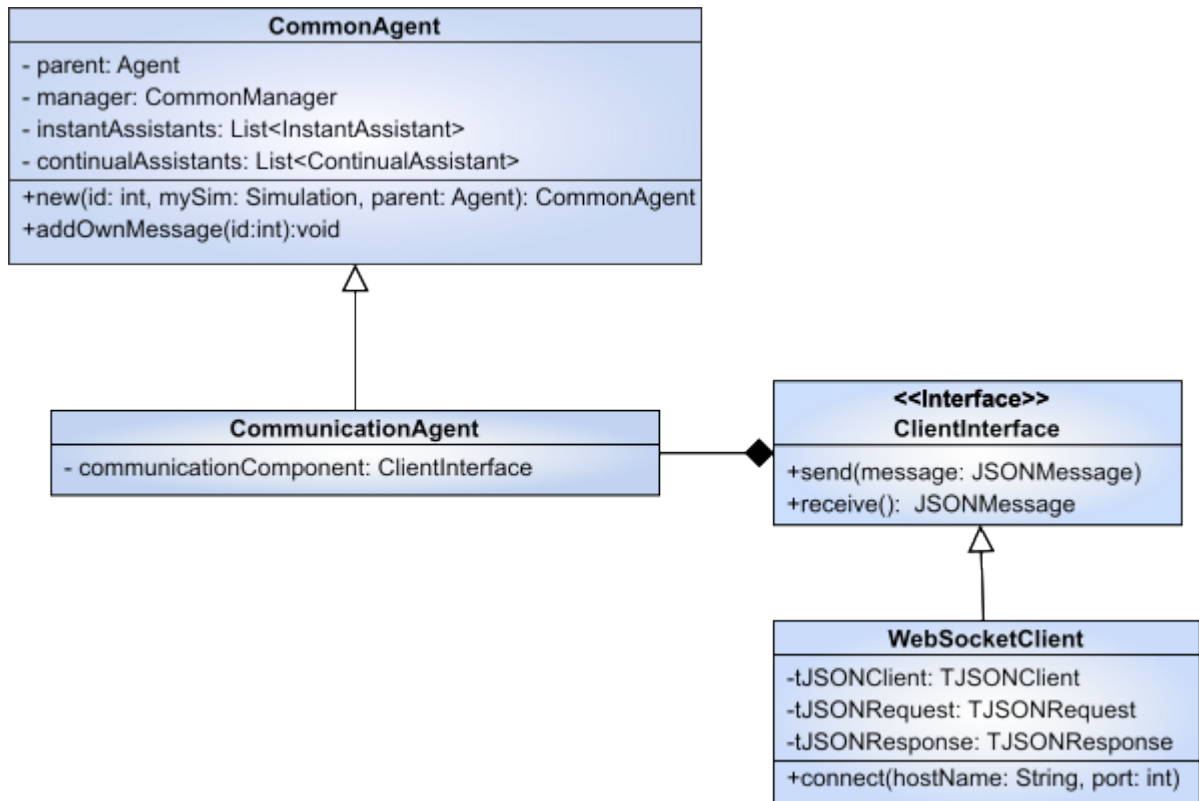
XML-RPC je protokolom vzdialeného volania procedúr, ktorý využíva XML formát. Tento protokol umožňuje používať mnoho dátových typov, avšak jeho nevýhodou je veľká veľkosť posielaných správ (vzhľadom na povahu formátu XML). [20]

Hoci spomenuté protokoly podporujú použitie štandardného HTTP protokolu, pre realizovanie komunikácie v reálnom čase sa javí ako vhodnejšie použitie WebSocket protokolu. WebSocket je štandardizovaný komunikačný protokol, ktorý poskytuje trvalý obojsmerný komunikačný kanál, ktorý pomocou jediného TCP pripojenia dokáže posilať správy z klienta na server aj zo servera ku klientovi. Spojenie je teda udržiavané počas celej doby trvania komunikácie. Tento protokol potrebuje podporu HTTP na iniciovanie spojenia. V prípade websocket komunikácie sa jedná o rýchly spôsob komunikácie, ktorý je žiaduci najmä pri tvorbe systémov reálneho času. [45][49]

V rámci navrhovanej architektúry bolo navrhnuté použitie JSON-RPC a websocket komunikácie a to vzhľadom na možnosť nepretržitej obojsmernej komunikácie, ktorú je potrebné zabezpečiť, keďže je potrebné udržiavať aktuálnosť simulačného modelu v reálnom čase. Nami definované správy spĺňajú požiadavky definované v tomto protokole a komunikačné rozhranie (Gateway API) je realizované ako http server a jednotlivé mikroslužby môžu byť realizované v rámci tohto servera, rôzne virtualizované, prípadne distribuované ako virtuálne stroje v cloude.

7.2.1 Implementácia komunikačného agenta

V rámci architektúry ABASim bolo potrebné rozšíriť funkcionality jednotlivých agentov ich rozšírením o využitie služieb určitého typu webového klienta. Pre tieto potreby bol definovaný nový špeciálny typ komunikačného agenta - `CommunicationAgent`. Dôležitým parametrom komunikačného agenta, ktorý je schopný komunikovať s externými modulmi je komunikačný komponent - `CommunicationComponent`, ktorý je implementáciou webového klienta. V rámci implementácie navrhovaného systému sme sa rozhodli použiť komunikačný protokol JSON-RPC, preto musia byť správy posielané externým aplikáciám vo formáte JSON. `CommunicationComponent` je modul, ktorý transformuje požiadavku agenta (odoslanú správu v rámci agentovo-orientovanej komunikácie) na správu vo formáte JSON s vopred definovanou štruktúrou. Komunikácia s externou aplikáciou je možná, nakoľko aj táto aplikácia dokáže nadviazať spojenie cez komunikačný kanál (Gateway API). Klient posieľa túto transformovanú požiadavku smerom von k externej službe, ktorá je realizovaná pomocou Gateway API. Gateway API následne určuje na základe charakteru požiadavky prichádzajúcej od agenta prostredníctvom klienta, ktorá mikroslužba požiadavku spracuje.



Obr. 7.4: UML diagram tried, potrebných pre implementáciu komunikačného agenta

Novonavrhnutý modul `CommunicationComponent` je potrebné zahrnúť do existujúcich štruktúr ABASim architektúry. Z popisu fungovania tejto architektúry vyplýva, že prístup k jednotlivým externým

modulom potrebujú využívať jednotliví agenti, aby mohli prijať rozhodnutia na základe odporúčaní doručených z externých modulov a pomocou svojich asistentov realizovať zmeny stavu systému.

Komunikačný komponent (parameter komunikačného agenta) implementuje rozhranie pomenované `ClientInterface`. Správa s požiadavkou na informácie z externého zdroja môže byť komunikačnému agentovi zaslaná od viacerých odosielateľov, najčastejšie iných agentov. V rámci komunikačného agenta je pomocou jeho asistenta ABAsim správa transformovaná na požadovaný formát a pomocou komunikačného komponentu odoslaná do siete. Obdobným spôsobom funguje aj komunikácia od externých komponentov smerom k jednotlivým agentom architektúry ABAsim. Ak komunikačný agent prijme správu zo siete, pomocou komunikačného rozhrania ju prevedie na požadovaný formát (správu definovanú v architektúre ABAsim) a túto správu odošle ostatným agentom. UML diagram popísaných komponentov zobrazuje obrázok 7.4.

7.3 Ukladanie aktuálneho stavu simulačného modelu

Aby bolo možné zabezpečiť aktuálnosť simulačného modelu vzhľadom na skutočnosť a najmä umožniť užívateľovi realizovať viaceré experimenty z jedného východiskového stavu, je potrebné zabezpečiť uloženie a znovu načítanie simulačného modelu.

Simulačné modely postavené na architektúre ABAsim (ale aj ostatné simulačné modely) sú veľmi komplexné. Ich dátové štruktúry sú rôzne previazané a pri ukladaní aktuálneho stavu nemôžeme opomenúť ani štruktúry týkajúce sa simulačného jadra. V prípade agentovo orientovanej simulácie je to najmä uloženie ešte nespracovaných prijatých správ v rámci jednotlivých agentov.

7.4 Zabezpečenie aktuálneho stavu vzhľadom na realitu

Aby bol zabezpečený aktuálny stav simulačného modelu vzhľadom na realitu, je potrebné sa zaoberať možnosťami, ako takúto synchronizáciu realizovať.

Jednou z možností je neustále aktualizovanie simulačného modelu. To znamená, že každá informácia, ktorá sa dostane na databázový server bude okamžite poslaná simulačnému modelu. Tento prístup sa môže javiť ako vhodný, vzhľadom na neustálu aktuálnosť, avšak je náročný najmä vzhľadom na zabezpečenie komunikácie medzi jednotlivými komponentami. Hoci sa môže zdať, že týmto spôsobom je možné odbúrať použitie databázy, toto nemožno tvorcom systému odporúčať, najmä s ohľadom na možnú stratu dát. Zasahovať do štruktúr bežiaceho simulačného modelu vždy, keď v reálnom systéme nastane zmena tiež nemusí byť vhodné, vzhľadom na rozsiahle väzby v dátových štruktúrach modelu.

Druhou možnosťou je aktualizácia po určitom čase, napr. každých 5 minút, prípadne v čase, kde sa používateľ rozhodne systém používať. V tomto prípade sú teda všetky relevantné dáta uložené v databáze, odkiaľ sú posielané do simulačného modelu. Synchronizácia prebieha spôsobom, že simulácia sa pustí z posledného uloženého bodu, následne sa prebiehajúci simulačný beh koriguje pomocou dát posielaných z databázy, napríklad sa menia časy aktivít, nastavujú sa cieľové kolaje podľa skutočnosti, pridelujú sa skutočné zdroje, atď. Hoci tento spôsob nezabezpečuje aktualizáciu v každom čase, má nižšie požiadavky

na výpočtové prostriedky, nižšie požiadavky na zabezpečenie bezpečnosti na sieti a nižšie požiadavky na správu celej komunikácie.

Aby mohli existujúce simulačné modely tvoriť súčasť systému na podporu rozhodovania, je potrebné zabezpečiť, aby boli schopné spracovať správy prichádzajúce z externého prostredia. Toto môže byť problematické, keďže jednotlivé simulačné modely a softvér slúžiaci na ich tvorbu nevznikli s ohľadom na možnosti komunikácie s externými zdrojmi.

Spôsob, akým bude správa poslaná vo vnútra simulačného modelu sme popísali v kapitole 7.2.1. Týmto krokom distribúcia informácie nekončí, keďže zmenu stavu simulácie zabezpečujú jednotliví agenti. Týmto agentom je potrebné definovať možnosť prijať správy prichádzajúce z externého prostredia. Túto skutočnosť je potrebné mať na zreteli aj v prípade, že by jadro systému na podporu rozhodovania netvorili simulačné modely postavené na architektúre ABASim, ale akékoľvek iné simulačné modely.

V rámci architektúry ABASim je potrebné zabezpečiť posielanie jednotlivých správ od komunikačného agenta k jednotlivým agentom, ktorí zabezpečujú funkcionality systému, ktorých sa tieto správy týkajú. Správy prichádzajúce z externého prostredia môžu informovať o zmene polohy vlaku, stavu vlaku, stavu personálu a podobne. Jednotliví agenti, ktorí tieto oblasti spravujú, musia teda očakávať správu (s príslušnými parametrami), ktorú im pošle komunikačný agent a túto správu musia vedieť vhodne spracovať, napríklad zmenou polohy konkrétneho vlaku.

Takýto zásah z vonkajšieho prostredia do fungovania simulačného modelu nie je jednoduché realizovať, nakoľko jedna informácia o zmene môže mať (a vo väčšine prípadov skutočne má) vplyv na mnohé iné atribúty. Simulačný model, ktorý stojí v jadre systému má totiž určitú počiatočnú konfiguráciu. Táto konfigurácia bola vytvorená na základe analýzy skutočného systému v štandardných podmienkach, ktoré nie vždy presne zachytávajú skutočné fungovanie systému. Počas behu simulačného modelu v reálnom čase totiž samozrejme dochádza k neočakávaným zmenám v reálnom systéme. Problematické je najmä dodržiavanie jednotlivých krokov technológie vlaku, nakoľko tieto nemusia byť dostatočne podrobne monitorované v rámci skutočného železničného uzla.

Situácie nastávajúce v rámci zabezpečenia udržania aktuálnosti simulačného modelu možno rozdeliť do dvoch skupín. V prvom prípade sa jedná o nesúlad medzi realitou a simulačným modelom, ktorý však nemení charakter definovanej technológie vlaku. K takejto situácii dochádza napríklad v takýchto prípadoch:

- vlak mal byť poslaný na koľaj 1, avšak bol poslaný na koľaj 2. Všetky nasledujúce úkony sú realizované podľa plánu,
- na realizáciu aktivity mal byť použitý personál 1, ale bol použitý personál 2,
- dĺžka aktivity (prípadne viacerých aktivít) bola v realite iná ako bola naplánovaná,
- aktivita (prípadne viacero aktivít) nebola vykonaná, doba jej trvania bola nulová.

V rámci takýchto situácií je potrebné pri spracovaní danej aktivity v simulačnom modeli upraviť vstupujúce dáta, akými sú plánovaná koľaj, personál alebo čas. Plánovaná koľaj a personál sú parametrami

správy, ktorá je posielaná príslušnému agentovi. Je preto potrebné zabezpečiť, aby komunikačný agent (jeho vnútorné komponenty) zmenil parametre správy ešte pred jej poslaním, nakoľko komunikačný agent disponuje informáciami z vonkajšieho prostredia. V prípade, že sa líši čas trvania aktivity, a teda nastane situácia, že z externého prostredia príde informácia, že aktivita (napríklad presun vlaku z miesta A do miesta B) začala v čase 8:00, avšak podľa plánu mala začať o 7:50. Znamená to, že aktivita v rámci simulačného behu je v inom bode, ako v realite (prípadne nemusí vôbec prebiehať). Je teda potrebné zabezpečiť zmenu dĺžky trvania jednotlivých aktivít.

Druhá skupina situácií je o čosi komplikovanejšia. Dochádza k nej vtedy, keď v realite dôjde k významnej zmene v technológii vlaku. Príkladom môže byť, že na vlaku má byť vykonané čistenie. V pôvodnej (plánovanej) technológii mala obsluha vlaku v simulačnom modeli prebiehať nasledovne: vlak sa objaví na vstupnej koľaji -> vlak sa presúva na koľaj 1 -> do vlaku prichádza personál vykonávajúci čistenie -> prebieha čistenie -> personál odchádza z vlaku -> vlak odchádza po výstupnej koľaji. Vzhľadom na nepredpokladané okolnosti v realite prebiehal obsluha takto: vlak sa objaví na vstupnej koľaji -> vlak sa presúva na koľaj 2 -> vlak stojí na koľaji 2 tri hodiny -> vlak sa presúva na koľaj 3 -> prichádza personál vykonávajúci čistenie -> prebieha čistenie -> personál odchádza z vlaku -> vlak odchádza po výstupnej koľaji. V tomto prípade predpokladáme, že celá obsluha sa už odohrala, avšak v skutočnosti sa môžeme na časovej osi nachádzať v akomkoľvek momente. To znamená, že je potrebné zaoberať sa synchronizačným mechanizmom, ktorý zabezpečí zmenu technológie vlaku. Toto predpokladá zásah do pôvodne statických dát počas simulačného behu.

7.4.1 Zabezpečenie zmeny priradeného zdroja

V pôvodnom simulačnom behu bolo definované, ktorý zdroj má byť priradený. Ak príde z reálneho systému informácia, že bol priradený iný zdroj, je potrebné zachytiť správu, ktorá sa posielala agentovi, zabezpečujúcemu priradenie zdroja a zmeniť parameter tejto správy na základe informácie z reálneho systému. Zvyšok práce zabezpečí agent, ktorý zabezpečuje priradenie zdroja. Vo fungovaní simulačného modelu sa nič významné nemení, nakoľko agenti vopred nevedia, aký zdroj majú priradiť.

Problém by mohol nastať jedine vtedy, ak by zdroj, ktorý bol v realite pridelený bol už pridelený inému vlaku v danom čase. V prípade, že budú zabezpečené kvalitné zdroje dát z reality by však takáto situácia nemala nastať.

7.4.2 Zabezpečenie zmeny polohy vlaku

Každý vlak, ktorý sa nachádza v simulačnom modeli má v danom čase definovanú svoju polohu. Táto poloha úzko súvisí s aktuálne vykonávaným technologickým krokom, ako aj všetkými technologickými krokmi, ktoré už boli vykonané. Pri synchronizácii stavu v realite a v simulačnom modeli však môže nastať situácia, v ktorej sa vlak nachádza v realite na inom mieste, ako by sa mal podľa preddefinovanej situácie, ktorú zobrazuje simulačný model. Môžu nastať dve situácie. V prvej situácii bola obsluha vlaku v realite realizovaná podľa určených technologických krokov, avšak cieľová koľaj v technologickom kroku

zabezpečujúcom presun bola zmenená. V tomto prípade, stačí vlaku v simulačnom modeli zmeniť polohu. Komplikovanejšia situácia nastáva, ak sa vlak na danú polohu dostal tak, že sa zmenili jeho technologické kroky. Vtedy je potrebné zabezpečiť synchronizáciu tak, že sa jednotlivé technologické kroky odoberú alebo pridajú.

7.4.3 Odobratie technologického kroku

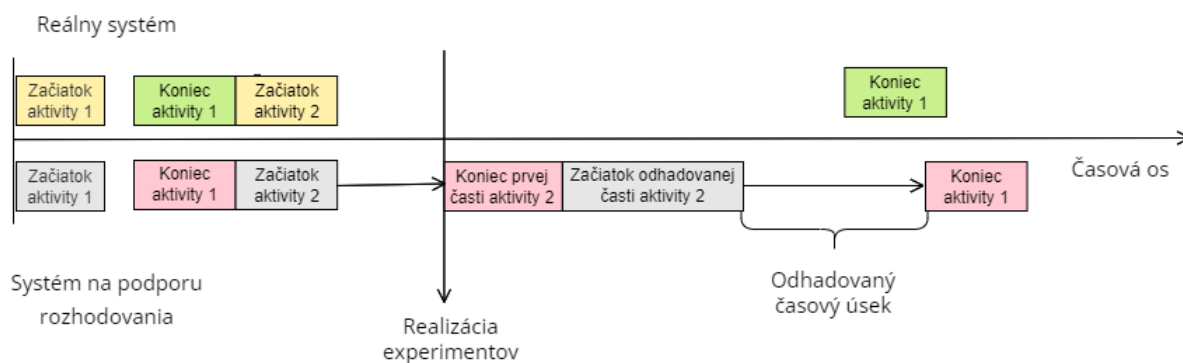
Ak je v rámci synchronizácie simulačného modelu s realitou potrebné odobrať nejaký technologický krok, o ktorom vieme, že sa v realite neuskutoční, je možné tento krok pre aktuálny vlak vynechať.

7.4.4 Pridanie technologického kroku

V prípade, že v realite nastane situácia, že na vlaku je vykonaný technologický krok, ktorý nebol naplánovaný, je potrebné zabezpečiť synchronizačné mechanizmy, ktoré umožnia prídanie takéhoto kroku. Toto bude realizované tak, že sa daný krok pridá priamo do štruktúry, ktorá definuje technológiu vlaku.

7.4.5 Zmena času trvania technologického kroku

Dĺžka trvania aktivity je zabezpečovaná pomocou posielania správy hold. Túto správu posielajú kontinúálny asistent, ktorý je súčasťou agenta zodpovedného za zabezpečenie priebehu technológie sám sebe. Aby bolo možné zmeniť dĺžku trvania na základe informácií z reality, je potrebné upraviť metódu, ktorá zabezpečuje posielanie správy hold. Keď sa rozhodneme so systémom realizovať experimenty, je potrebné, aby v určitom čase bola simulácia v konzistentnom stave, kedy je možné realizovať tieto experimenty, avšak s použitím plánovaných činností (v realite tieto činnosti ešte neboli vykonané). Vtedy je potrebné, aby všetky bežiacie technologické kroky boli v stave, kedy ich koniec vieme odhadnúť pomocou plánu. Ako je vidieť na obrázku 7.5, odhadovaný časový úsek nemusí zodpovedať realite. Toto je však skutočnosť, s ktorou musí tvorca systému pri odhade budúceho stavu počítať.



Obr. 7.5: Zmena času trvania technologického kroku

8 Implementácia navrhovaného systému bez sofistikovaných metód

Navrhovaný systém na podporu rozhodovania v reálnom čase má poskytovať dispečerovi, pomocou simulačného modelu, možnosť sledovať aktuálnu situáciu v železničnom dopravnom uzle a využívať výhody simulačných modelov, akými je najmä experimentovanie s rôznymi variantami budúceho vývoja. Pri implementácii systému na podporu rozhodovania sme postupovali podľa navrhutej metodiky.

V prvom kroku boli teda definované požiadavky na daný systém. V nami implementovanom systéme bude realizovaná podpora rozhodovania v rámci údržbového depa. Dispečerovi bude umožnené sledovať aktuálnu situáciu tak, ako to umožňuje vybraný simulačný model. Ako výstupné štatistiky budú sledované časy odchodu jednotlivých vlakov zo systému, ako aj časy začiatkov trvania a koncov trvania jednotlivých obslužných aktivít.

Aby bolo možné implementovať navrhovaný systém, bolo potrebné definovať dáta, ktoré je potrebné čerpať z reálneho systému. Pripojenie k skutočným systémom nebolo možné, najmä z byrokratických dôvodov, ale aj z dôvodu absencie niektorých zdrojov informácií, ktorých použitie navrhujeme v tejto práci. Vzhľadom k tomu bol ako náhrada reality použitý simulačný model. Predpokladáme, že podobné dáta by bolo možné získavať aj z reálnych systémov, avšak uvedomujeme si, že získavanie týchto dát by bolo oveľa náročnejšie, nakoľko nemusia byť v dostatočnej kvalite, na rozdiel od dát získaných zo simulačného modelu. Aby bolo možné overiť navrhovanú metodiku, bolo určené, že z reality je potrebné získavať dáta o čase príchodu vlakov do systému, okrem toho časy začiatkov a koncov jednotlivých technologických krokov, ako aj informácie o zdrojoch (personálnych), ktoré tieto kroky realizujú.

Následne bolo potrebné definovať komunikačné rozhranie. Vychádzajúc z návrhu popísaného v kapitole 3.2 bola vytvorená serverová aplikácia, ktorej súčasťou je databáza. Táto serverová aplikácia je schopná zbierať údaje z reálneho systému a ukladať ich do danej databázy. Databáza je tvorená tromi tabuľkami a to tabuľkou s informáciami o vlakoch, tabuľkou o jednotlivých zdrojoch (najmä personálnych) a tabuľkou o aktivitách. Informácie z reality sú serverovej aplikácii posielané vo formáte JSON pomocou websocket komunikácie, ktorá zabezpečuje sieťovú komunikáciu. Zabezpečenie tejto komunikácie následne umožňuje rozposielať dáta ďalším pripojeným stranám, v našom prípade systému na podporu rozhodovania. Správa, ktorá je poslaná a obsahuje informácie o technologickom kroku môže mať takýto tvar:

```
{  "trainId": "Number",
    "staffId": "Number",
    "startTime": "DateTime",
    "endTime": "DateTime" }
```

Navrhovaný systém na podporu rozhodovania má vo svojom jadre simulačný model vytvorený v simulačnom nástroji Villon. Pri skutočnej implementácii by musel byť vytvorený simulačný model predmetného železničného uzla so všetkými údajmi, ktoré je možné získať z plánov prevádzky daného železničného uzla. Pre demonštráciu riešenia bol použitý simulačný model, ktorý bol použitý aj na nahradenie reality, však upravený tak, aby bolo možné overiť funkčnosť navrhovaného riešenia (boli zmenené plánované technológie, časy príchodov vlakov, personálu a podobne).

V rámci implementácie navrhovaného systému bolo upravené simulačné jadro ABASim tak ako je popísané v kapitole 7.1 a bol upravený simulačný softvér Villon tak, aby bolo možné zabezpečiť komunikáciu s vonkajším prostredím.

Pre overenie funkčnosti celého systému boli realizované tri testovacie scenáre. Všetky testovacie scenáre vychádzajú zo základného datasetu, ktorý obsahuje 15 vlakov, tieto vlaky sú buď dieselové, alebo elektrické a na každom treba vykonať určité obslužné aktivity. Týmito aktivitami môže byť veľká údržba, malá údržba, čistenie interiéru, čistenie exteriéru, čistenie grafitov a tankovanie, pričom veľká údržba a tankovanie sa týkajú iba dieslových vlakov.

Testovací scenár 1 zahŕňa meškanie jednotlivých vlakov a čas tohto meškania je modelovaný pomocou spojitého rovnomerného rozdelenia pravdepodobnosti na intervale (0,60) minút. Časy trvania jednotlivých obslužných aktivít sú pevne dané. Bolo realizovaných 30 simulačných behov pre generovanie dát, ktoré nahrádzajú realitu. Rovnako tak bolo realizovaných 30 použití systému na podporu rozhodovania, ktorý načítaval získané údaje. Pre jednotlivé ukazovatele boli zostrojené 90% intervaly spoľahlivosti a výsledky porovnávajúce stav v systéme, ktorý nahrádza realitu a v navrhovanom systéme na podporu rozhodovania zobrazuje tabuľka 8.1.

Testovací scenár 2 zahŕňa meškanie jednotlivých vlakov a čas tohto meškania je modelovaný pomocou spojitého rovnomerného rozdelenia pravdepodobnosti na intervale (0,60) minút a časy trvania niektorých aktivít sú taktiež náhodne generované a je ich možné modelovať pomocou takýchto rozdelení pravdepodobnosti:

- čistenie grafitov dieselového vlaku - exponenciálne rozdelenie so strednou hodnotou 60 minút,
- čistenie exteriéru dieselového vlaku - exponenciálne rozdelenie so strednou hodnotou 10 minút,
- veľká údržba dieselového vlaku - spojité rovnomerné rozdelenie na intervale (30,180) minút,
- tankovanie - spojité rovnomerné rozdelenie na intervale (30,40) minút,
- malá údržba elektrického vlaku - spojité rovnomerné rozdelenie na intervale (30,180) minút,
- čistenie exteriéru elektrického vlaku - exponenciálne rozdelenie so strednou hodnotou 60 minút,
- čistenie interiéru elektrického vlaku - exponenciálne rozdelenie so strednou hodnotou 60 minút.

Výsledky porovnávajúce stav v systéme, ktorý nahrádza realitu a v navrhovanom systéme na podporu rozhodovania pre testovací scenár 2 zobrazuje tabuľka 8.2.

Testovací scenár 3 má rovnaké charakteristiky ako scenár 1 a okrem toho zahŕňa zrušenie príchodu dvoch vlakov. Sledované parametre zobrazuje tabuľka 8.3.

Tabuľka 8.1: Porovnanie ukazovateľov medzi simulačným modelom nahrádzajúcim realitu a systémom na podporu rozhodovania-testovací scenár 1

Ukazovateľ	Sim. model nahrádzajúci realitu (hod.)	Systém na podporu rozhodovania (hod.)
Celkové meškanie vypravených vlakov	<42.82;48.69>	<42.73;48.55>
Trvanie aktivity Veľká údržba	<38.98;40.52>	<37.99;40.48>
Trvanie aktivity Malá údržba	<16.17;16.17>	<16.17;16.17>
Trvanie aktivity Tankovanie	<5.34;5.70>	<5.25;5.65>
Trvanie aktivity Čistenie grafitov	<14.81; 15.15>	<14.92;15.12>
Trvanie aktivity Čistenie interiéru	<17.5;17.5>	<17.5;17.5>
Trvanie aktivity Čistenie exteriéru	<16.49;16.98>	<16.42;16.85>

Tabuľka 8.2: Porovnanie ukazovateľov medzi simulačným modelom nahrádzajúcim realitu a systémom na podporu rozhodovania-testovací scenár 2

Ukazovateľ	Simulačný model nahrádzajúci realitu (hod.)	Systém na podporu rozhodovania (hod.)
Celkové meškanie vypravených vlakov	<63,54;70,54>	<63.35;70.78>
Trvanie aktivity Veľká údržba	<15,36;15,38>	<15.25;15.48>
Trvanie aktivity Malá údržba	<46,23;46,24>	<46.12;46.85>
Trvanie aktivity Tankovanie	<14,52;14,58>	<14.54;14.60>
Trvanie aktivity Čistenie grafitov	<24,96;25,22>	<24.42;24.90>
Trvanie aktivity Čistenie interiéru	<19,41;19,44>	<19.35;19.42>
Trvanie aktivity Čistenie exteriéru	<5,86;5,90>	<5.44;5.95>

Pre správne vyhodnotenie výsledkov jednotlivých experimentov boli, pre všetky ukazovatele, vykonané párové T-testy. Tieto testy potvrdili, že jednotlivé hodnoty získané z reálneho systému sa na štatisticky

Tabuľka 8.3: Porovnanie ukazovateľov medzi simulačným modelom nahrádzajúcim realitu a systémom na podporu rozhodovania-testovací scenár 3

Ukazovateľ	Simulačný model nahrádzajúci realitu (hod.)	Systém na podporu rozhodovania (hod.)
Celkové meškanie vypravených vlakov	<34.78;37.76>	<34.62;36.99>
Trvanie aktivity Veľká údržba	<34.79;36.16>	<34.68;36.14>
Trvanie aktivity Malá údržba	<13.60;14.06>	<13.55;14.12>
Trvanie aktivity Tankovanie	<4.72;5.02>	<4.72;5.02>
Trvanie aktivity Čistenie grafitov	<13.01;13.46>	<13.10;13.44>
Trvanie aktivity Čistenie interiéru	<14.36;15.23>	<14.31;15.21>
Trvanie aktivity Čistenie exteriéru	<13.73;14.51>	<13.73;14.45>

významnej hladine nelíšia od hodnôt, ktoré sú výstupom zo systému na podporu rozhodovania.

V rámci overovania funkčnosti navrhovaného postupu bol teda vytvorený systém, v ktorého jadre stojí simulačný model. Tento simulačný model bol vytvorený v simulačnom softvéri Villon. Keďže simulačný model tvorí jadro systému na podporu rozhodovania, je schopný prijímať dáta z externého prostredia. Toto je realizované pomocou komunikácie s databázovým serverom, kde sú ukladané dáta popisujúce stav reálneho systému. Načítané údaje sú zobrazované v simulačnom modeli v užívateľsky prívetivej forme. Bola realizovaná séria experimentov, ktorých cieľom bolo overiť, či simulačný model zobrazuje aktuálnu skutočnú situáciu. Toto bolo potvrdené. Už takýto systém vie byť nápomocný v rozhodovacom procese, nakoľko umožňuje dispečerovi experimentovať s rôznymi variantami budúceho vývoja. Takýto systém však stále očakáva vytváranie návrhov rozhodnutí dispečerom - používateľom. Nami navrhovaný systém však umožňuje čiastočne znížiť závislosť na ľudskom faktore tým, že využíva rôzne sofistikované - optimalizačné algoritmy, ktoré sú popísané v nasledujúcej kapitole.

9 Metodika použitia sofistikovaných metód v systémoch využívajúcich simulačné modely

Aby bolo možné do systému na podporu rozhodovania v reálnom čase integrovať rôzne sofistikované metódy, ktoré by umožnili lepšiu podporu rozhodovania bol vypracovaný metodický postup. Tento metodický postup zahŕňa všetky činnosti, ktoré sú potrebné, ak chceme v navrhovanom systéme na podporu rozhodovania používať sofistikované metódy, pričom hlavným zameraním tohto metodického postupu je používanie algoritmov učenia posilňovaním. Fázy takéhoto metodického postupu možno definovať nasledovne:

- definovanie úlohy,
- výber sofistikovanej metódy,
- definovanie dát, potrebných pre použitie vybranej metódy,
- implementácia sofistikovanej metódy,
- úprava simulačného modelu,
- definovanie komunikačného rozhrania,
- prípravná fáza (trénovanie),
- nasadenie sofistikovanej metódy,
- overenie funkčnosti systému.

9.1 Fáza 1 - Definovanie úlohy

- Cieľ:** Presné vymedzenie úlohy, v rámci, ktorej má byť použitá sofistikovaná metóda. Sofistikované metódy sú zväčša používané na návrh riešenia jedného problému. Takýmto problémom môže byť potreba rozhodnúť o vhodnom personáli; vhodnej koľaji; uprednostňovaní obslužných úloh, atď.
- Vstup:** Informácie o fungovaní konkrétneho železničného uzla, ktoré umožňujú dôkladnú analýzu jeho fungovania.

Postup: V prvom rade je potrebné analyzovať fungovanie daného vybraného železničného uzla. Toto je možné na základe jednotlivých zdrojov informácií, rozhovormi s pracovníkmi uzla, sledovaním jeho prevádzky a inými spôsobmi. Výstupom z tejto analýzy je definovanie úlohy, na ktorej je možné použiť sofistikovanú metódu. Väčšina metód, ktoré sú v súčasnosti používané predpokladá dobre izolované zadanie, keďže je potrebné kvantifikovať všetky okolnosti danej situácie. Toto je dobre realizovateľné na úlohách, do ktorých nevstupuje priveľa vedľajších vplyvov. Samozrejme je možné definovať aj úlohy, ktoré zahŕňajú komplexnejšie problémy, do ktorých vstupuje mnoho ťažko identifikovateľných faktorov.

Výstup: Popis vybranej úlohy v slovnej podobe, zahŕňajúci jednotlivé vstupné a výstupné veličiny.

9.2 Fáza 2 - Výber sofistikovanej metódy

Cieľ: Výber vhodnej sofistikovanej metódy vzhľadom na vybranú úlohu.

Vstup: Popis vybranej úlohy.

Postup: Vzhľadom na popísanú vybranú úlohu je potrebné vybrať vhodnú metódu. Treba dôkladne zvážiť charakter danej úlohy, pretože rôzne metódy sú vhodné na rôzne typy úloh. Pri úlohách menšieho rozsahu, v ktorých sú dobre definované jednotlivé vstupné veličiny a vzťahy medzi nimi, je vhodné použitie exaktného výpočtu na základe matematického modelu. V prípade, že sa jedná o úlohu väčšieho rozsahu, môžu byť použité heuristické metódy. V prípade, že väzby medzi jednotlivými parametrami nie je možné dobre definovať, odporúča sa použitie algoritmov učenia posilňovaním. Samozrejme tento postup nie je možné jednoducho zovšeobecniť, odporúča sa vychádzať zo skúseností s jednotlivými algoritmami, či opierať sa o použitie algoritmov na podobné úlohy v minulosti.

Výstup: Vybraný typ sofistikovanej metódy.

9.3 Fáza 3 - Definovanie dát, potrebných pre použitie vybranej metódy

Cieľ: V tejto fáze je potrebné detailné definovanie dát, ktoré sú potrebné ako vstup do sofistikovanej metódy, aby táto mohla poskytnúť požadovaný výstup.

Vstup: Popis vybranej úlohy, typ vybranej sofistikovanej metódy.

Postup: Na základe popisu vybranej úlohy a vlastnostiach vybranej sofistikovanej metódy je potrebné definovať vstupné dáta. Tieto dáta sa samozrejme líšia vzhľadom na vybranú metódu. Exaktný a heuristický prístup zvyčajne potrebuje celý dataset - dáta z celého časového úseku, o ktorom sa rozhoduje. To znamená, že simulačný model musí realizovať tzv. „inicializačný“ beh, v ktorom získa všetky potrebné dáta. Je potrebné navrhnuť spôsob, akým budú tieto dáta získavané.

V prípade použitia algoritmov učenia posilňovaním sa dáta týkajú aktuálneho časového okamihu a spolupráca so simulačným modelom prebieha počas celej doby tréningu (viac popísanej vo fáze 7). Dáta získavané zo simulačného modelu musia byť normalizované, v prípade algoritmov učenia posilňovaním väčšinou tak, aby sa nachádzali na intervale $(-1,1)$ alebo $(0,1)$. Netreba zabúdať, že dáta týkajúce sa jednej veličiny je potrebné získavať v rovnakých jednotkách. Pre použitie algoritmov učenia posilňovaním je potrebné definovať stav prostredia v danom okamihu a odmenu/pokutu za vykonané akcie. V prípade definovania vektora, ktorý popisuje stav prostredia je potrebné zvážiť, ktoré vstupné informácie sú potrebné pre realizovanie správnych rozhodnutí. Rovnako je dôležité zvážiť rozmer daného vektora - množstvo informácií, ktoré budú algoritmu poskytnuté, nakoľko sa s narastajúcim počtom dát spomaľuje tréning neurónovej siete, ktorú využívajú algoritmy učenia posilňovaním.

Dáta definujúce stav prostredia musia vychádzať z hĺbkovej analýzy rozhodovacieho procesu dispečera. Pri výbere podstatných dát navrhujeme rozdeliť si tieto informácie do viacerých skupín a to:

- informácie o vlakoch v systéme - toto môže zahŕňať ich aktuálnu polohu, stav rozpracovania technológie, plán budúcich aktivít realizovaných na danom vlaku, plánovaný čas odchodu;
- informácie o plánovaných vlakoch - plán príchodov vlakov v budúcnosti a to najmä čas príchodu vzhľadom na aktuálny čas v systéme. Okrem toho plán činnosti na daných vlakoch (jednotlivé obslužné aktivity, prechod určitými koľajami a iné);
- informácie o infraštruktúre - obsadenosť jednotlivých koľají, pričom pri výbere podstatných dát je dôležité zvážiť, ktoré koľaje sú dôležité pre riešenie daného rozhodovacieho problému, napr. ak rozhodujeme o poradí jednotlivých vlakov nad servisnou jamou, nemusí byť signifikantné obsadenie príjazdových koľají do depa;
- informácie o personáli - počet jednotlivých zamestnancov, ich aktuálna pozícia, aktuálny stav (pracujúci, presúvajúci sa, voľný). Samozrejme je potrebné zvážiť relevantnosť vzhľadom na riešenú úlohu.

Pre použitie algoritmov učenia posilňovaním je potrebné definovať odmenu. Táto môže byť daná počas simulačného behu (čiastková) alebo na jeho konci (celková). Typ odmeny je potrebné zvážiť vzhľadom na typ úlohy, ako aj vzhľadom na možnosti simulátora. Odmena by mala vychádzať z parametrov, ktoré je potrebné dosiahnuť a to napríklad minimálny počet meškajúcich vlakov, minimálny súčet časov meškania, vyťaženosť personálu a iné.

Výstup: Zoznam zahŕňajúci definíciu jednotlivých dát spolu s jednotkami, prípadne spôsobom ich normalizácie a s informáciou o rozsahu získavaných dát. Príklad: pre použitie algoritmu učenia posilňovaním pre realizáciu obslužnej úlohy - údaj o polohe vlakov (napr. súradnice) v danom okamihu, čas príchodu vlakov v minútach, zoznam časov trvania všetkých úloh na predmetnom vlaku v minútach. Všetky veličiny sú normalizované, tak aby nadobúdali hodnoty na intervale (0,1). Toto je realizované spôsobom, že je zistený maximálny rozsah sledovaného ukazovateľa, tento maximálny rozsah nadobúda v normalizovanej veličine hodnotu 1, ostatné hodnoty sú potom určené príslušným vzťahom, ktorý vyjadruje pomerovú vzdialenosť od maximálnej hodnoty.

9.4 Fáza 4 - Implementácia sofistikovanej metódy

Cieľ: Implementácia vybranej sofistikovanej metódy vo vybranom programovacom jazyku.

Vstup: Popis úlohy, definícia vstupných dát, rozhodnutie o programovacom jazyku.

Postup: V tejto fáze je potrebné implementovať sofistikovanú metódu, ktorá bude schopná prinášať návrhy rozhodnutí pri riešení daného problému na základe definovaných vstupných dát. Výstup z tejto metódy bude v požadovanej forme. V tejto fáze by sa mal tvorca systému rozhodnúť, či bude sofistikovaná metóda súčasťou vybraného simulačného modelu. Táto možnosť zjednodušuje prenos dát, avšak predpokladá, že sofistikovaná metóda bude implementovaná v rovnakom programovacom jazyku, prípadne musí byť realizovaná integrácia pomocou obalovej aplikácie. Je na tvorcovi systému, aby sa rozhodol, či je tento prístup vhodnejší ako implementácia optimalizačnej metódy v rámci externej aplikácie. Na základe tohto rozhodnutia je potrebné vybrať vhodný programovací jazyk, technológie, navrhnúť údajové štruktúry a architektúru danej aplikácie/module (v rámci OOP hovoríme o rôznych triedach).

Výstup: Funkčná implementácia sofistikovanej metódy.

9.5 Fáza 5 - Definovanie komunikačného rozhrania

Cieľ: Definovanie spôsobu akými budú prenášané dáta medzi simulačným modelom a sofistikovanou metódou.

Vstup: Vybraná metóda, definícia vstupných dát.

- Postup:** Pre použitie sofistikovanej metódy je potrebné zabezpečiť prenos dát medzi simulačným modelom a danou metódou. Sofistikovaná metóda môže byť súčasťou simulačného modelu. V takom prípade je prenos dát jednoduchší, avšak aj tu treba myslieť na použitie vhodných údajových štruktúr a zabezpečenie prístupu k výsledkom sofistikovanej metódy vhodných spôsobom, v rámci ktorého sú dodržané pravidlá tvorby daného softvéru. V prípade, že byt použitá externá aplikácia, je potrebné zabezpečiť komunikáciu medzi dvomi aplikáciami. Preto je potrebné navrhnúť komunikačný protokol. Tento protokol závisí od možností simulačného modelu, ako aj externej aplikácie, ktorá umožňuje použitie sofistikovanej metódy. Tento protokol by mal obsahovať definíciu formátu a štruktúry jednotlivých správ, ako aj spôsob ich posielania.
- Výstup:** Rozhodnutie o spôsobe prenosu dát. Definícia správ, ktoré budú prenášané.

9.6 Fáza 6 - Úprava simulačného modelu

- Cieľ:** Simulačné modely nemusia umožňovať získavanie výstupov v požadovanej forme a to najmä v reálnom čase počas simulačného behu. Okrem toho zvyčajne nie je umožnená komunikácia, tak ako to definuje komunikačné rozhranie, vyplývajúce z predchádzajúcej fázy. V prípade, že sa v predchádzajúcej fáze rozhodlo, že sofistikovaná metóda bude súčasťou simulačného modelu, je potrebné rozšíriť architektúru tak, aby toto bolo možné.
- Vstup:** Simulačný model.
- Postup:** V rámci rozšírenia simulačného modelu môžeme hovoriť o dvoch čiastkových fázach. Prvou fázou je zabezpečenie poskytnutia výstupných údajov v požadovanej forme (forma bola definovaná vo fáze 2) - vzhľadom na povahu mnohých simulačných modelov (napríklad oddelenie jednotlivých funkcionalít do uzavretých častí) môže byť náročné získať prístup k jednotlivým dátam počas behu simulačného modelu. Je potrebné definovať komponent, ktorý zabezpečí zber a analýzu týchto dát a rozšíriť jednotlivých agentov o možnosť poskytovať tieto dáta tomuto agentovi.
- Druhou fázou je rozšírenie o možnosti komunikácie podľa komunikačného rozhrania/integrácia sofistikovanej metódy do simulačného modelu - v prípade, že je potrebné zabezpečiť komunikačné rozhranie, je potrebné definovať štruktúry, ktoré túto komunikáciu môžu zabezpečiť (komunikačný agent v agentovo orientovanej simulácii). Tieto štruktúry musia byť schopné pripojiť sa k externým stranám a komunikovať s nimi. V prípade, že je potrebná integrácia sofistikovanej metódy je možné definovať osobitnú štruktúru (agenta), ktorá bude mať v sebe zahrnutú implementáciu danej metódy a bude komunikovať s ostatnými agentmi.
- Výstup:** Rozšírený simulačný model.

9.7 Fáza 7 - Prípravná fáza (trénovanie)

- Cieľ:** Realizácia tréningu vybraného algoritmu (týka sa najmä algoritmov učenia posilňovaním) tak, aby dosahoval dostatočne kvalitné výsledky.
- Vstup:** Systém na podporu rozhodovania s implementovaným funkčným vybraným algoritmom.
- Postup:** Táto fáza sa týka najmä použitia algoritmov učenia posilňovaním, nakoľko tieto vyžadujú fázu tréningu. Pri tejto fáze treba myslieť najmä na dlhý čas tréningu (dni, týždne). Tvorca systému by mal preto zvážiť, aké výpočtové zariadenie bude použité, prípadne aj použitie paralelného tréningu na viacerých výpočtových zariadeniach. Toto však môže byť náročné, najmä pri použití algoritmov učenia posilňovaním, nakoľko treba použiť špecifické synchronizačné mechanizmy. Zvyčajne táto synchronizácia prebieha spôsobom, že sa trénuje viacero neurónových sietí a po určitom čase sú tieto spojené pomocou špeciálnych funkcií do jednej siete. Čas tréningu však nemusí byť pre tvorca systému vždy kľúčový (záleží na tom, za aký čas je potrebné uviesť systém do prevádzky). Pri realizácii tréningu je potrebné určiť aj dĺžku tréningu - počet replikácií. Toto je možné odhadnúť na základe skúseností, avšak skutočný počet replikácií je zrejmý až počas tréningu a to tak, že pozorujeme ustalovanie dosiahnutých výsledkov. Je potrebné určiť parametre, pomocou ktorých je možné určiť, že sú dosahované dostatočne dobré výsledky. Tieto parametre závisia od povahy vybranej úlohy. To, aké dobré sú dosiahnuté výsledky je možné zistiť porovnaním výsledkov s použitím iných metód, alebo konzultáciou s dispečerom príslušného železničného uzla. Pri prevádzke systému na podporu rozhodovania v reálnom čase sa používa už natrénovaný algoritmus.
- Výstup:** Natrénovaný algoritmus, dosahujúci dostatočne kvalitné výsledky.

9.8 Fáza 8 - Nasadenie sofistikovanej metódy

- Cieľ:** Nasadenie sofistikovanej metódy do fungujúceho systému na podporu rozhodovania.
- Vstup:** Sofistikovaná metóda, systém na podporu rozhodovania.
- Postup:** Táto fáza zahŕňa zabezpečenie získania dát zo simulačného modelu, ich spracovanie vybranou metódou a poskytnutie výstupu simulačnému modelu. V prípade, že na prenos dát bude použitý sieťový komunikačný mechanizmus, v tejto fáze je potrebné zabezpečiť jeho fungovanie.
- Výstup:** Systém na podporu rozhodovania zahŕňajúci sofistikovaný algoritmus.

9.9 Fáza 9 - Overenie funkčnosti systému

- Cieľ:** Pre overenie funkčnosti systému je potrebné sledovať výstupné ukazovatele zo simulačného modelu.

Vstup: Systém na podporu rozhodovania.

Postup: Pre overenie funkčnosti systému, je potrebné testovať správnosť riešení, ktoré navrhuje sofistikovaný algoritmus a to tak, že budú sledované výstupy zo simulačného modelu. Na základe výstupných ukazovateľov je možné sledovať, či integrácia odporúčaných hodnôt do simulačného modelu prispela k zlepšeniu fungovania systému a to napríklad porovnaním viacerých metód.

Výstup: Funkčný systém.

10 Integrácia optimalizačných metód do systému na podporu rozhodovania v reálnom čase

Po vytvorení funkčného systému na podporu rozhodovania v úrovni 1 je na základe návrhu celého systému potrebné zabezpečiť aj poskytovanie návrhov rozhodnutí pri riešení vzniknutých problémov pomocou sofistikovaných metód. Takého návrhy rozhodnutí (zvyčajne o pridelení určitého obmedzeného zdroja) je ideálne realizovať optimálne, hoci vo väčšine prípadov je možné sa ku takému optimu len priblížiť. Hľadať takého optimálne/takmer optimálne riešenia je možné pomocou mnohých prístupov, ako napríklad pomocou exaktného výpočtu pomocou matematického modelu, heuristických metód alebo pomocou algoritmov strojového učenia.

10.1 Aplikácia optimalizačných metód v železničnej doprave

Rozhodnutia, ktoré je potrebné realizovať pri riadení železničnej dopravy, sú značne komplexné a nie je jednoduché jednoznačne určiť spôsob ich realizácie. Aby bolo možné zvoliť vhodnú metódu pre riešenie určitého problému, je potrebné analyzovať spôsoby použitia jednotlivých optimalizačných metód pri riešení problémov v železničnej doprave (najmä železničnom uzle).

10.1.1 Exaktné metódy

Hoci sa vo väčšine prípadov riešenia problémov v železničnej doprave jedná o NP ťažké úlohy, viacero autorov aplikovalo pri hľadaní riešenia exaktné metódy.

Autori [33][34] riešili problém priradenia vhodného nástupišťa vlaku v prípade meškania s cieľom minimalizovať meškanie ostatných vlakov. Pre tento cieľ boli vytvorené matematické modely a pomocou nich následne riešená úloha na skutočných železničných uzloch pomocou exaktných metód. V článku [33] bol zostavený matematický model, ktorého účelová funkcia má dve časti. Prvá časť minimalizuje vážené časové odchýlky od plánovaného časového rozvrhu pre jednotlivé vlaky, pričom váha tu zohľadňuje či sa jedná o rýchlik alebo regionálny vlak. Druhá časť maximalizuje vhodnosť priradeného nástupišťa vlaku. Navrhovaný model bol aplikovaný na reálne dáta získané zo železničného dopravného uzla Praha - hlavná stanica a to z cestovných poriadkov platných v rokoch 2004/2005. Stanica mala v tom čase 7 nástupíšť, 17 nástupištných koľají a 8 príchodových/odchodových koľají. Dáta použité pre experimenty boli z ranej špičky medzi 5:00 a 10:00. Časový horizont plánovania bol nastavený na 2 hodiny a boli vytvorené dve sady experimentov. Prvý variant počítal s meškaním jedného vlaku v rôznych časových intervaloch

od 1 po 60 minút. V druhom variante sa počítalo v meškaním 2 vlakov 25 minút. Model bol implementovaný v dvoch riešiteľoch a to Xpress a Gurobi. Výpočty boli porovnané s návrhmi skutočných dispečerov a ukázalo sa, že v niektorých prípadoch matematický model ponúka lepšie riešenia a môže byť použitý na podporu riadenia na operatívnej úrovni v reálnom čase. V článku [34] bola použitá čiastočná modifikácia navrhovaného matematického modelu. Experimenty boli rovnako realizované na dátach z hlavnej stanice v Prahe. V týchto experimentoch sa však neuvažovalo nad meškaním vlakov, ale matematický model slúžil na overenie prípustnosti cestovného poriadku vzhľadom na predpisy. Výsledky ukázali, že cestovný poriadok nie je možné dodržať, ak všetky vlaky prídu bez omeškania, čo sa však v praxi nestáva.

V článku [19] autori predpokladajú časovú náročnosť výpočtov pomocou exaktných metód, a preto síce využívajú matematické modely, ale optimalizačný problém riešia sústavou jednoduchých heuristík.

Autori [58] navrhli zložitý algoritmus RECIFE-MILP (REcherche sur la Capacité des Infrastructures FErroviaires), ktorý síce využíva matematický model a hľadá exaktné riešenia, avšak je obmedzený nastaveným časovým limitom a využíva viaceré zlepšenia ako poskytnutie počiatočného riešenia, zníženie hodnoty veľkej konštanty, zavedenie implicitnej priority medzi vlakmi a iné.

Aby bola dosiahnutá lepšia výpočtová efektivita pri riešení problémov v železničnej doprave v reálnom čase je možné využiť metódu vetiev a hraníc a zahrnúť do výpočtu implikačné pravidlá [12]. Pre riešenie konfliktov vznikajúcich na trati autori navrhli použitie jednoduchých pravidiel, ktoré vychádzajú z dispečerskej praxe a to „First Come First Served (FCFS)“, čiže pri obsluhu je uprednostnený ten vlak, ktorý prišiel na križovatku ako prvý. Ďalším pravidlom je „First Leave First Served (FLFS)“, a teda ak dva vlaky vyžadujú prístup k rovnakej sekcii, najprv sa vypočíta čas, ktorý v tejto sekcii potrebujú stráviť a uprednostní sa vlak, ktorý má tento čas kratší. Okrem toho bol v tomto článku navrhnutý heuristický prístup (vychádzajúci z predchádzajúcej práce autorov), ktorý realizuje optimalizáciu pomocou zákazu alternatívnej sekcie, ktorej prejazd by trval najväčší čas. Navrhované riešenia boli aplikované na úzke hrdlá holandskej železničnej trate a ukázalo sa, že je možné dosiahnuť takmer optimálne výsledky v krátkom čase.

Okrem priradovania jednotlivých koľají vlakom sa v rámci operačného výskumu v železničnej doprave kladie pozornosť aj tvorbe časových rozvrhov pre personál. Autori [32] v snahe priradiť personál jednotlivým vlakom presúvajúcim sa podľa vopred určeného časového rozvrhu zostavili matematický model a úlohu následne riešili metódou generovania stĺpcov. Pri riešení zadanej úlohy sa predpokladá, že prevádzka železnice funguje periodicky, a preto sa plán tvorí pre jednu periódu. Podmienky matematického modelu zabezpečujú priradenie jednej pracovnej úlohy jednej pracovnej skupine, ďalej to, aby v nasledujúcom stanovišti, do ktorého skupina príde bol potrebný rovnaký počet pracovných miest ako v tom, z ktorého vyšla, a v neposlednom rade zabezpečenie toho, že posádky sa nesmú vracaať ani v priestore, ani v čase. Účelová funkcia modelu sa snaží minimalizovať celkové náklady na prechody jednotlivých pracovných skupín železničnou sieťou, pričom tieto náklady môžu byť vyjadrené rôzne, napríklad skutočnými režijnými nákladmi alebo vzdialenosťami medzi jednotlivými stanovišťami, prípadne počtom použitých

pracovných skupín. Efektívnosť navrhovaného riešenia bola overená na skutočných dátach z dopravného systému na Slovensku a v Maďarsku. Experimenty ukázali, že exaktný prístup je v tom prípade použiteľný, avšak s rastúcim rozsahom úlohy sa rýchlo dosiahne strop možností komerčných IP riešiteľov. Naproti tomu, metóda generovania stĺpcov je použiteľná aj na úlohy väčšieho rozsahu, avšak nedokáže zaručiť optimálne riešenie.

Podobný prístup zvolili autori [25] a problém priradenia personálu efektívne riešia tak, že vopred generujú množinu všetkých pracovných úloh, ktoré sú následne použité ako nemenný vstup do matematického modelu. Pre tento matematický model je potrebné definovať súbor jednotlivých dep, medzi ktorými je potrebné zabezpečovať presuny. Okrem toho podmnožinu dep, disponujúcich obmedzeným počtom rušňovodičov, ďalej súbor kompletných úloh, súbor povinností a dvojice takých úloh, aby východisková poloha bola zároveň koncovou úlohou. Ako parametre do modelu vstupujú jednotlivé stanice, jednotlivé vlaky, čas začiatku úlohy, čas konca úlohy, doba trvania úlohy, čas trvania prípravných činností rušňovodičov pred každou službou, minimálny čas potrebný na prechod rušňovodiča z jedného vlaku do druhého, počet dostupných rušňovodičov v depe, počet potrebných rušňovodičov pre splnenie danej úlohy, minimálny/maximálny povolený pracovný čas rušňovodiča, minimálny celkový čas odpočinku a minimálny nepretržitý čas odpočinku. Model sa snaží minimalizovať celkový pracovný čas, ako aj čas potrebný na prípravné činnosti. Navrhovaný model bol aplikovaný na dataset získaný zo železničnej spoločnosti v Poľsku, obsahoval 7 vlakov a 670 úloh. Výsledky experimentov ukázali, že exaktným riešením danej úlohy je možné dosiahnuť redukciiu pracovného času rušňovodičov.

Snaha riešiť problémy komplexne je v rámci existujúcich výskumov nepostačujúca. Vzhľadom na komplexnosť celého systému sa výskum v tejto oblasti zameriava najmä na riešenie čiastkových, striktnie vymedzených úloh. Autori [44] predpokladali, že optimálne riešenia je možné nájsť pomocou rozdelenia problému na viacero menších subproblémov, založených na geografii, jednotlivých skupinách vlakov a časových intervaloch.

Na základe analýz realizovaných výskumov sa dá povedať, že optimalizačné úlohy v danej oblasti sú natoľko rozsiahle, že je náročné (najmä časovo) ich riešiť štandardnými exaktnými metódami. Použitie heuristických metód pri riešení takýchto úloh by mohlo byť vhodnejšie.

10.1.2 Heuristické metódy

Hoci heuristické metódy negarantujú dosiahnutie optimálneho riešenia, mnoho autorov ich aplikuje na riešenie problémov v železničnej doprave a to najmä z dôvodu ich časovej efektívnosti.

Pri minimalizácii meškania vlakov použili autori [55] heuristiku ant colony optimization. Ďalším výskumom v tejto oblasti bolo hľadanie vhodnej koľaje pre vlak pomocou heuristik local search a tabu search [10]. Základný prechod od jedného riešenia k druhému spočíva v zmene trasy jedného vlaku, pričom je však nevyhnutné kontrolovať, či je toto riešenie dosiahnuteľné a nespôsobí zablokovanie celej trate. Výsledky experimentov, uskutočnené na železničnej trati v reálnom svete, demonštrujú vysoký potenciál

presmerovania vlakov s cieľom minimalizovať po sebe idúce meškania.

Viacere výskumy ukázali, že pri riešení problémov s priradením koľaje prichádzajúcemu vlaku je efektívne aj použitie genetického algoritmu, prípadne rôznych jeho variánt. V článku [65] autori navrhujú použitie genetického algoritmu a porovnávajú ho s heuristickým prístupom „First-scheduled first-served“ (prvý bude obslužený vlak, ktorý prišiel ako prvý). Východisková populácie bolo definovaná ako náhodne vybraná podmnožina množiny všetkých kombinácii vlaku a koľaje. Každý gén však musí byť realizovateľný, a preto je potrebné kontrolovať viaceré podmienky, napríklad dodržanie dĺžky koľaje vzhľadom na dĺžku priradeného vlaku. Keďže cieľom algoritmu je minimalizácia meškania, algoritmus realizuje operácie kríženia a mutácií dovtedy, kým nie je dosiahnuté nulové meškanie, alebo kým nevyprší maximálny čas výpočtu. Aplikácia algoritmov na reálnych dátach ukázala, že navrhovaný genetický algoritmus dokáže znížiť čas meškania v porovnaní s „First-scheduled first-served“ prístupom. Autori [17] sa rozhodli porovnávať výsledky genetického algoritmu s použitím neurónovej siete. Jednotlivé chromozómy boli definované ako binárne vyjadrené rozvrhy vlakov, ktorých jednotlivé bity nesú informáciu o konfliktných situáciách a ich spôsobe riešenia. Tieto chromozómy tvoria populáciu, na ktorú sú aplikované operácie kríženia a mutácie, realizované náhodne a kvalita vytvorenej populácie je charakterizovaná celkovým vzniknutým meškaním. Algoritmus bol otestovaný na 51 hypotetických problémoch, pričom výsledky boli porovnané s výsledkami neurónovej siete. Táto neurónová sieť bola, rovnako ako genetický algoritmus implementovaná v prostredí Matlab a bola trénovaná na základe rozhodnutí skutočných dispečerov. Výsledky ukázali, že navrhovaný GA je schopný dosahovať lepšie výsledky ako neurónová sieť.

Autori [4] v snahe vyriešiť problém oneskoreného príchodu vlaku zostavili maticu 4 kritérií (stupeň obsadenosti trate v čase príchodu vlaku; obdobie obsadenosti trate vzhľadom na čas, ktorý strávi prichádzajúci vlak v stanici; obsadenie susednej trate spojovacím vlakom; ďalšie technické a technologické preferencie vzhľadom na prichádzajúci vlak). Tieto kritériá boli stanovené na základe skutočných skúseností dispečerov. Na základe tejto matice bol vytvorený dataset z údajov z hlavnej stanice v Prahe. Následne bola trénovaná dvojvrstvá neurónová sieť a výsledky ukázali využiteľnosť neurónovej siete pre takýto typ úloh. V roku 2019 [5] bola na tomto výskume založená aj štúdia využívajúca multikritériálnu analýzu. Pre použitie tejto metódy bola vytvorená multikritériálna matica, v ktorej jednotlivé riadky tvoria hodnotiace kritéria (v tomto prípade štyri) a jednotlivé stĺpce predmetné koľaje. Na základe multikritériálnej matice bola koľaj s maximálnou hodnotou posudzovaných kritérií priradená príslušnému vlaku. Následne bola táto metóda testovaná na modeli prevádzky osobnej stanice v Lysej nad Labem, ktorá má štyri železničné trate a 7 nástupísk. V štúdiu bol použitý cestovný poriadok z rannej špičky zahŕňajúci 29 vlakov. Pre realizovanie tohto experimentu bol využitý simulačný softvér Villon, v rámci ktorého bolo implementované rozšírenie využívajúce multikritériálnu analýzu. Ako optimalizačné kritérium pre vyhodnotenie experimentu bol stanovený celkový prírastok meškania vlakov. Na základe vykonaných experimentov sa ukázalo, že tento prístup prináša lepšie výsledky v porovnaní s metódou priority listu.

Heuristické metódy sú rovnako dobre použiteľné aj pri priradovaní personálu jednotlivým pracovným

úlohám, ktoré je potrebné riešiť v železničnej doprave. Autori [5] pri tvorbe časových rozvrhov zahŕňajúc aj priradenie jednotlivých pracovných skupín použili heuristiku simulované žíhanie, ktorej efektívnosť potvrdili aplikáciou na reálnych operatívnych dátach z východosibírskej železnice. Pri riešení problému priradenia personálu je možné použiť genetický algoritmus [38], prípadne ant colony optimization [75].

10.1.3 Metódy strojového učenia

Metódy strojového učenia, najmä algoritmy učenia posilňovaním boli taktiež použité vo viacerých výskumoch zaoberajúcich sa optimalizáciou v železničnej doprave. Autor [27] hľadal metódy na optimalizáciu rozmiestnenia nákladných vagónov v zriaďovacej stanici. Pri experimentoch vychádzal z predpokladu, že počítačové usporiadanie týchto vagónov je náhodné a následne sú usporiadané do poradia, ktoré je požadované pre odchádzajúci vlak. Usporiadanie a pohyb jednotlivých vagónov popisuje aktuálny stav v stanici a prechody z jedného stavu do druhého boli definované na základe Markovovho rozhodovacieho procesu. Následne bol aplikovaný algoritmus Q učenia, ktorý zohľadnil vzdialenosti potrebné na presun ako aj počet jednotlivých pohybov. Prístup popísaný v tomto článku autor využil v ďalšom výskume [28], kde navrhuje zlepšenie prístupu rozdelením problému na menšie oblasti, ktoré spolu vnútorne súvisia. Toto rozdelenie zlepšuje kvalitu dosahovaných výsledkov.

Autori [16] sa vo svojom článku zaoberajú použitím algoritmov učenia posilňovaním pri optimalizácii použitej energie pri presune jednotlivých vozňov.

V rámci zriaďovacej stanice je potrebné zabezpečiť správnu rýchlosť vozňov, prichádzajúcich na zväzky pahorok. To, že na určenie tejto rýchlosti je možné použiť učenie posilňovaním je popísané v článku [64], kde autori svoje výsledky porovnávajú s reálnymi dátami.

Učenie posilňovaním bolo využité aj pri tvorbe rozvrhu pre železničnú trať [37], pričom každý vlak bol popísaný vlastným stavovým vektorom. Na základe tohto vektora a prislúchajúcej Q hodnoty sa vyberie vhodná akcia. V snahe preskúmať širšie okolie riešení bol zahrnutý určitý stupeň randomizácie pri výbere vhodnej akcie. Výsledky navrhovaného algoritmu boli porovnané s výsledkami dvoch heuristik a ukázalo sa, že algoritmus učenia posilňovaním dosahuje lepšiu kvalitu rozvrhu a to aj pri testovaní na skutočnej železničnej trati.

Ďalší výskum v tejto oblasti [7] sa snažil minimalizovať čas zotrvania každého vlaku v stanici. V rámci aplikácie učenia posilňovaním bol použitý multiagentový systém a boli použité tri rozhodovacie metódy (FIFO, náhodné prechody a Q učenie). Ako účelová funkcia pre Q učenie bol použitý celkový čas zotrvania v stanici. Hoci multiagentový systém dosahuje dobré výsledky, táto štúdia potvrdila, že Q učenie nie je vhodné na rozsiahle úlohy.

10.1.4 Na simulácii založené optimalizačné metódy

Pri snahe zefektívniť spôsob riadenia v železničnom dopravnom uzle sa viacero autorov zaoberalo aj spojením simulačného modelu a optimalizácie.

Vo výskume [3] boli použité výsledky diskretnej simulácie pre tréning neurónovej siete. Týmto spôso-

bom bola realizovaná snaha o nájdenie optimálneho súboru pravidiel pre dispečerov.

Autori [29] použili kombinovaný prístup simulácie a optimalizácie na minimalizáciu meškania pri zostavovaní cestovných poriadkov železníc. Najskôr bol zostavený matematický model, ktorý definuje podmienky, ktoré je potrebné sledovať. Následne bol definovaný simulačný model v prostredí Railsys a bol navrhnutý algoritmus riešenia. Navrhovaný algoritmus pozostáva zo štyroch krokov:

- výber inicializačného rozvrhu a definovanie konštanty pre meškania,
- realizácia stanoveného počtu simulačných behov,
- nastavenie parametrov matematického modelu na základe výstupov zo simulácie,
- riešenie matematického modelu.

Použitelnosť navrhovaného riešenia bola overená na dátach zo švédskej železnice a vybraných bolo 87 vlakov a experimenty ukázali, že navrhované riešenie je schopné znížiť meškania vlakov až o 50 percent. Pri tomto výskume je dôležité spomenúť, že autori kladú dôraz aj na sekundárne meškania, ktoré sú generované konfliktnými situáciami.

Pri riešení problému priradenia koľaje zmeškanému vlaku použili autori [14] metódu vnorenej simulácie. V tejto metóde je hlavná simulácia pri konfliktnej situácii prerušená a naklonovaná. Následne je realizovaných viacero simulačných behov, z ktorých sa vyberie ten najlepší. Aby mohla byť vnorená simulácia aplikovaná je potrebné definovať si niekoľko parametrov a to:

- kritérium optimality, ktoré umožní výber najvhodnejšieho simulačného behu,
- trvanie výhľadu do budúcnosti pre jednotlivé vnorené simulácie,
- počet replikácií pre všetky vnorené simulácie,
- počet vnorených simulácií/alternatívnych scenárov.

Tieto parametre vplyvajú na časovú a výpočtovú náročnosť. Je zrejmé, že výpočtová náročnosť tohto prístupu je vysoká, a preto autori odporúčajú realizovať väčšinu výpočtov na grafickej karte, prípadne výpočty realizovať paralelne na viacerých počítačoch. V rámci výskumu boli realizované experimenty v simulačnom nástroji Mesorail. Ako už vyplýva z názvu, jedná sa o simulátor, ktorý vytvára simulačné modely na mezoskopickú úroveň a tým znižuje náročnosť v porovnaní s mikroskopickou simuláciou. Ako kritérium optimality bolo zvolené celkové meškanie všetkých vlakov v simulácii. Trvanie výhľadu vnorenej simulácie bolo nastavené na 5 minút, počet vnorených simulácií bolo maximálne množstvo vyplývajúce zo situácie a pre každú vnorenú simuláciu bola realizovaná jedna replikácia. Bol vytvorený model obsahujúci 5 nástupištných koľají, dve zjazdové koľaje a jednu dvojkolajovú trať. Vlaky do tohto modelu prichádzali v 10 minútovom intervale z oboch smerov. Výsledky ukázali, že táto metóda je ťažko aplikovateľná vzhľadom na výpočtovú náročnosť a citlivosť vzhľadom na nastavenie jednotlivých parametrov.

V rámci iniciatívy Shift2rail bol realizovaný projekt OptiYard, ktorého hlavnou úlohou bolo navrhnuť systém na podporu rozhodovania pre správu terminálov zriaďovacích staníc [43]. Tento projekt bol založený na myšlienke skombinovať možnosti mikrosimulácie a optimalizačných algoritmov. Hlavným benefitom bola nepretržitá komunikácia medzi simulačným modelom a optimalizačnou metódou. Ostatné

výskumy realizovali túto komunikáciu len vo veľkých dávkach, zvyčajne po ukončení simulačného behu, avšak tento projekt ukázal, že komunikácia v reálnom čase môže byť prínosná pre podporu rozhodovania dispečerov. Pre vytvorenie simulačných modelov bol použitý simulačný softvér Villon, ktorý bol v tomto projekte použitý pre zostavenie neoptimalizovaných modelov železničných uzlov. Okrem toho bol vytvorený simulačný model okolitej železničnej siete pomocou TrackULA. TrackULA ponúka mikroskopický simulačný model, ktorý predstavuje pohyb jednotlivých vlakov. Je založená na diskretnej simulácii, kde sa aktualizuje stav vlaku v pevnom časovom intervale. Pre hľadanie optimálnych riešení bol vytvorený optimalizačný modul obsahujúci heuristický algoritmus. Základom tohto algoritmu je heuristika „local search“, ktorá prehľadáva okolie tak, že vymieňa poradie jednotlivých operácií priradených ku zdroju. V rámci navrhovaného systému bola zabezpečená komunikácia simulačných modelov s optimalizačným modulom pomocou vopred definovaných správ v štandardizovanom formáte JSON a statické dáta boli zdieľané pomocou XML súborov. Návrh riešenia bol testovaný na dvoch reálnych systémoch (Česká Třebová – 700 vagónov za 24 hodín, Terst – 8000 vlakov ročne). Simulačný model bol však v tomto projekte použitý len na generovanie vstupnej situácie nahrádzajúcej realitu. Kľúčovým pre hľadanie riešení bol optimalizačný modul, ktorý pracoval úplne nezávisle od simulačného modelu. To znamená, že pracoval s vlastným dátovým modelom a tento bol priebežne aktualizovaný na základe situácie v simulačnom modeli. Ak v simulácii došlo ku konfliktnej situácii, bola vyslaná požiadavka optimalizačnému modulu a tento rozhodol nezávisle na simulácii. Následne bolo toto izolované riešenie poslané pomocou komunikačného servera naspäť do simulačného modelu a simulačný beh pokračoval ďalej.

Realizované výskumy zväčša používali simuláciu ako generátor vstupných informácií pre optimalizačný proces, prípadne na demonštráciu dosiahnutých výsledkov optimalizácie.

10.2 Možnosti integrácie výsledkov optimalizačných metód do systémov na podporu rozhodovania

Aby bolo možné v nami navrhovanom systéme zabezpečiť prítomnosť sofistikovaných algoritmov, je potrebné tieto algoritmy implementovať ako mikroslužby reagujúce na požiadavky vychádzajúce zo simulačného modelu, tak ako je popísané v kapitole 7.2.

Tieto mikroslužby môžu zahrnúť optimalizačné algoritmy realizované v ľubovoľnom programovacom jazyku, čím sú zabezpečené viaceré možnosti pri implemetácii daného algoritmu. Je však potrebné definovať výstup z týchto mikroslužieb, ako správu vo formáte JSON, tak aby bol simulačný model schopný spracovať tento výstup a následne aplikovať výsledky optimalizačných algoritmov tak, že používateľ vidí výsledky simulačného behu, ktorý používa tieto optimalizačné metódy.

Simulačný model (v našom prípade postavený na architektúre ABAsim) musí byť schopný vytvoriť požiadavku smerom k optimalizačnému algoritmu. Takáto požiadavka sa zvyčajne zaoberá pridelovaním obmedzených zdrojov, s ktorými rôzne optimalizačné algoritmy pracujú rôznymi spôsobmi a teda pre svoje správne fungovanie očakávajú rôzne vstupné informácie.

Na základe realizovaných výskumov v oblasti použitia optimalizačných metód sme definovali tri základné typy optimalizačných metód:

- exaktný výpočet realizovaný pomocou matematického modelu,
- heuristický prístup v podobe rôznych primárnych heuristík (výmenná heuristika, vkladacia heuristika a iné) alebo metaheuristík (genetický algoritmus, simulované žihanie, tabu search a iné),
- strojové učenie (v prípade úloh riešených v železničných dopravných uzloch najmä učenie posilňovaním).

Pre tieto tri základné typy boli definované všeobecné postupy, ktorými by simulačný model mal definovať požiadavky, tieto požiadavky odoslať a následne prijať odpoveď a túto odpoveď spracovať.

10.2.1 Integrácia exaktného výpočtu pomocou matematického modelu

Exaktný výpočet realizovaný pomocou matematického modelu predpokladá existenciu takéhoto modelu, ktorý na základe vstupných informácií vytvorí riešenie. Vstup do matematického modelu je tvorený súborom viacerých typov premenných, zvyčajne sa jedná o binárne alebo celočíselné premenné.

V prípade, že chce tvorca navrhovaného systému na podporu rozhodovania využiť pri riešení konkrétného problému exaktný výpočet, musí poznať súbor týchto premenných. Keďže tieto premenné môžu obsahovať informácie o mnohých častiach systému (napríklad sú potrebné údaje o príchode jednotlivých vlakov a zároveň údaje o pracovných zmenách rušňovodičov), je potrebné tieto údaje zozbierať na jednom mieste. Keďže sa pri tomto type optimalizačného algoritmu (najmä pre jeho časovú náročnosť) predpokladá, že na základe jedného súboru vstupných údajov bude realizovaných viacero rozhodnutí (napríklad pri pridelovaní rušňovodičov možno na základe jedného algoritmu prideliť všetkých rušňovodičov) je vhodné realizovať simulačný beh dvakrát. Pri prvej realizácii sú postupne odosielané potrebné údaje o aktuálnom simulačnom behu smerom k optimalizačnému algoritmu (v podobe mikroslužby). Tento optimalizačný algoritmus vytvorí na základe prijímaných informácií potrebnú dátovú štruktúru. Následne realizuje výpočet a čaká na požiadavku od simulačného modelu o poslanie informácie, ktorý konkrétny zdroj má byť pridelený. Keď takúto požiadavku dostane, odošle správu s danou informáciou. Simulačný model musí vedieť takúto požiadavku spracovať.

V rámci simulačného modelu je teda potrebné:

- poznať dáta, ktoré vyžaduje optimalizačný algoritmus,
- realizovať inicializačný simulačný beh,
- počas neho postupne odosielať požadované dáta,
- realizovať ďalší simulačný beh,
- odosielať požiadavku na pridelenie konkrétného zdroja,
- očakávať odpoveď obsahujúcu informácie o konkrétnom zdroji,
- spracovať túto odpoveď a prideliť konkrétny zdroj.

10.2.2 Integrácia heuristického prístupu

Heuristický prístup sa v mnohom podobá na exaktný. Na strane optimalizačného algoritmu totiž stojí metóda, ktorá na základe vstupných údajov takisto prináša určitý výstup (v našom prípade rozhodnutie o pridelení zdroja). Preto môže byť zvolený obdobný prístup ako pri integrácii výsledkov exaktného výpočtu.

Výhodou heuristického prístupu je však väčšia rýchlosť výpočtu (aj keď za cenu potenciálnej straty optimálneho riešenia), preto je v niektorých prípadoch vhodnejšie realizovať výpočet pri každej požiadavke o pridelenie zdroja. Tento prístup umožňuje lepšie zachytiť náhodné deje, ktoré prebiehajú v simulačnom modeli (rovnako ako v reálnom systéme). V prípade, že by mal byť v rámci navrhovaného systému na podporu rozhodovania použitý heuristický prístup, možno ho realizovať rovnako ako exaktný výpočet, alebo zvoliť postup, pri ktorom je síce realizovaný inicializačný beh na získanie predpokladaných údajov, avšak počas druhého simulačného behu sú spolu s požiadavkou o pridelenie konkrétneho zdroja posielané aj informácie o aktuálnom stave jednotlivých parametrov systému, ktoré heuristický algoritmus potrebuje k svojmu výpočtu. Tento výpočet je potom realizovaný pre každú požiadavku zvlášť.

10.2.3 Integrácia učenia posilňovaním

Keďže učenie posilňovaním predpokladá existenciu simulačného prostredia, v ktorom môže byť rozhodovací algoritmus (agent) trénovaný na to, aby rozhodoval žiadúcim spôsobom [13], javí sa tento prístup ako vhodná optimalizačná metóda v oblasti optimalizácie obslužných systémov, akými sú aj železničné dopravné uzly. V prípade učenia posilňovaním, ktoré pre svoje fungovanie vhodne využíva neurónové siete je potrebné rozdeliť celý proces použitia tejto metódy do dvoch častí a to trénovanie rozhodovacieho algoritmu (agenta) a používanie natrénovaného agenta.

Pre použitie algoritmov učenia posilňovaním je potrebné definovať:

- stav prostredia v danom okamihu vo forme vstupného stavového vektora. V prípade železničného uzla môže stav definovať poloha jednotlivých vlakov, poloha personálu, vzdialenosti medzi nimi, zoznam prebiehajúcich obslužných činností a podobne. To, ktoré konkrétne ukazovatele prostredia sú zvolené vychádza z charakteru úlohy, ktorá je riešená. Napríklad, v prípade, že rozhodujeme o pridelení vhodnej koľaje je dôležité definovať stav obsadenosti jednotlivých koľají, ale menej dôležitý (prípadne úplne nepodstatný) je aktuálny pracovný stav personálu. Hodnoty tohto vektora je vhodné upraviť na hodnoty z intervalu 0 až 1,
- akcie, ktoré povedú k zmene stavu prostredia. V prípade železničného uzla môže byť takouto činnosťou ukončenie obslužnej činnosti a prechod na inú, pridelenie vhodnej vlakovej trate, pridelenie personálu k vlaku, pridelenie obslužnej činnosti vlaku a podobne,
- odmeny/pokuty, ktoré sa budú udeľovať za každé uskutočnenie akcie. Tieto odmeny môže definovať účelová funkcia, v ktorej je snahou minimalizovať/maximalizovať určitú súhrnnú hodnotu, ako napríklad celkové meškanie vlakov, celkové pracovné vyťaženie personálu, počet využitých/ne-

využitých zamestnancov, počet obsluhovaných vlakov a podobne. Môže byť určená aj ako súhrn viacerých ukazovateľov, ktoré sa na základe poznania daného uzla a riešenej úlohy ukazujú ako podstatné. Pri udeľovaní odmeny je potrebné zvážiť, či táto odmena bude udelená po každej akcii, alebo iba jedna (koncová) za celý simulačný beh. Toto taktiež vychádza z charakteru problému, pri niektorých problémoch je veľmi ťažké určiť čiastkovú odmenu.

V procese tréningu musí byť teda simulačný model schopný odosielať všetky definované informácie o stave prostredia vždy, keď je potrebné uskutočniť rozhodnutie. Následne musí simulačný model čakať na odpoveď, ktorá obsahuje informáciu, ktorý zdroj má byť pridelený. Po pridelení tohto zdroja je potrebné stanoviť odmenu podľa definovaných pravidiel a informáciu o tej odmene odoslať optimalizačnému algoritmu.

Je samozrejmé, že proces tréningu predpokladá veľké množstvo realizácií simulačného behu, a teda zaberie značný čas. V rámci použitia systému na podporu rozhodovania v reálnom čase však používateľ pracuje s už natrénovaným algoritmom učenia posilňovaním. V procese používania tohto typu optimalizačného algoritmu teda simulačný model odosiela celkový stav celého prostredia a dostáva odpoveď obsahujúcu informáciu, ktorý zdroj má byť pridelený.

11 Implementácia navrhovaného systému s využitím sofistikovaných metód

Ak dokážeme zabezpečiť existenciu systému, v ktorého jadre stojí simulačný model a tento model je neustále aktualizovaný informáciami zo skutočného systému, môžeme do takéhoto systému zahrnúť použitie rôznych sofistikovaných optimalizačných metód. Vzhľadom na rôznorodosť a komplexnosť problémov, ktoré musí dispečer riešiť v rámci operatívneho riadenia, sme sa rozhodli overiť použiteľnosť navrhovaného riešenia na vybraných izolovaných úlohách. Pre overenie použiteľnosti boli vybrané dve údržbové depá, najmä pre jednoduchšie technologické postupy v údržbovom depe v porovnaní s inými uzlami, akými sú napríklad zriaďovacie stanice. V rámci tejto dizertačnej práce sme sa zaoberali dvomi izolovanými úlohami a to pridelovaním personálu a stanovením poradia vykonávania údržbových činností.

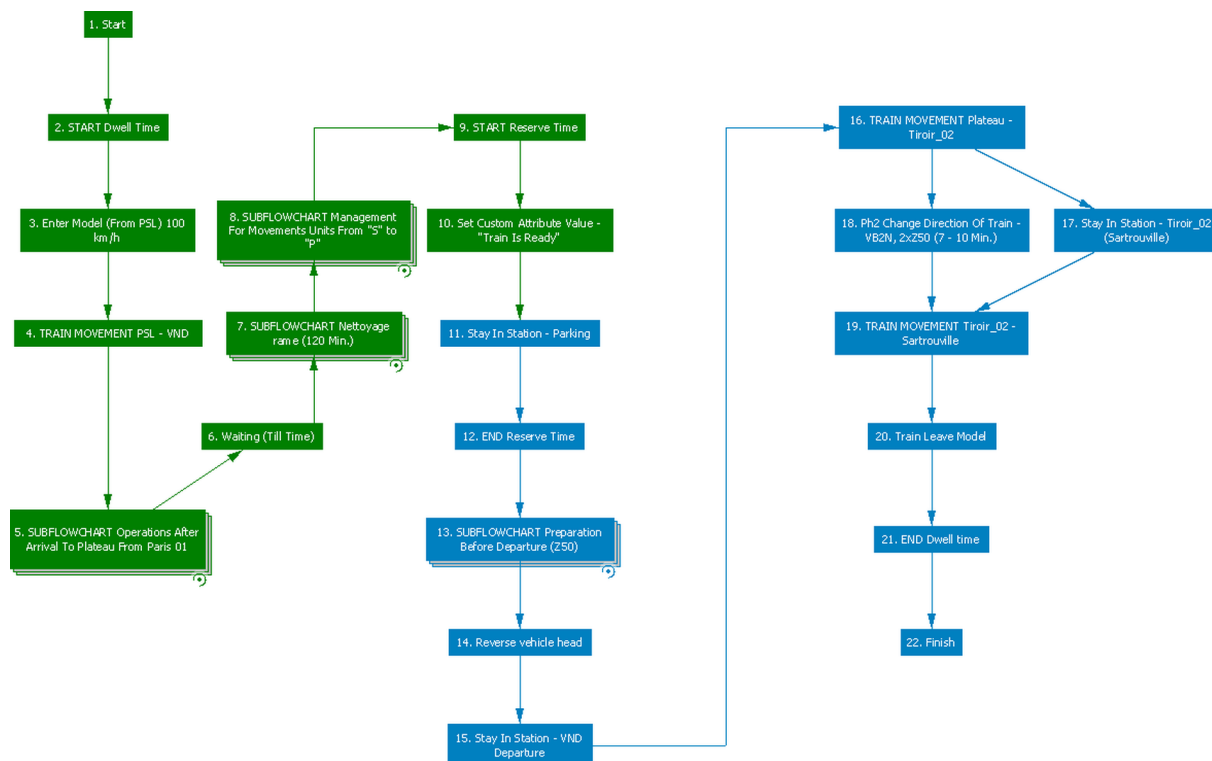
11.1 Úloha priradenia rušňovodičov

Údržbové depo je jedným z mnohých typov železničných dopravných uzlov. Vlaky do depa prichádzajú podľa presného, vopred daného časového rozvrhu a každý vlak má definovaný aj čas odchodu z depa. V tomto depe sa realizuje údržba vlakov v čase, keď nie sú potrebné v železničnej sieti, to znamená, že väčšina činností v tomto depe je vykonávaná počas noci. Každý vlak má definované činnosti, ktoré by na ňom mali byť vykonané a tieto činnosti sú popísané vopred definovanou technológiou. Príklad takejto technológie zobrazuje obrázok 11.1.

Depo sa skladá z dvoch hlavných častí a to tzv. vstupnej časti a údržbovej časti. Vstupná časť je spojená s príjazdovými koľajami a je tvorená zväzkom koľají, na ktorých stoja vlaky po príjazde do depa čakajúce na údržbu, alebo vlaky po údržbe, ktoré čakajú na čas svojho odchodu z depa. Údržbová časť je tvorená stanovištami, na ktorých sú vykonávané rôzne údržbové činnosti alebo čistenie. Obrázok 11.2 zobrazuje vyššie popísanú infraštruktúru predmetného depa.

Aby mohol vlak vykonať presun v rámci depa, je potrebné, aby mu bol priradený interný rušňovodič. Títo rušňovodiči majú definované osemhodinové pracovné smeny, počas ktorých sa nachádzajú vo vlaku, presúvajú sa medzi vlakmi, presúvajú sa na vopred definované stanovište, prípadne v tomto stanovišti čakajú, kým budú priradení ďalšiemu vlaku. Rušňovodič teda môže nadobúdať stav:

- dostupný - nachádza sa na pracovisku, ale momentálne nie je priradený žiadnemu vlaku, ani sa sám nikam nepresúva,
- pracujúci - je priradený vlaku a tento vlak presúva, prípadne sú realizované činnosti potrebné pre začiatok pohybu vlaku,
- čakajúci - je priradený vlaku, ale vlak sa nehýbe, pretože čaká, kým sa uvoľní koľaj, po ktorej by



Obr. 11.1: Technológia obsluhy vlaku v simulačnom modeli údržbového depa

mal ísť,

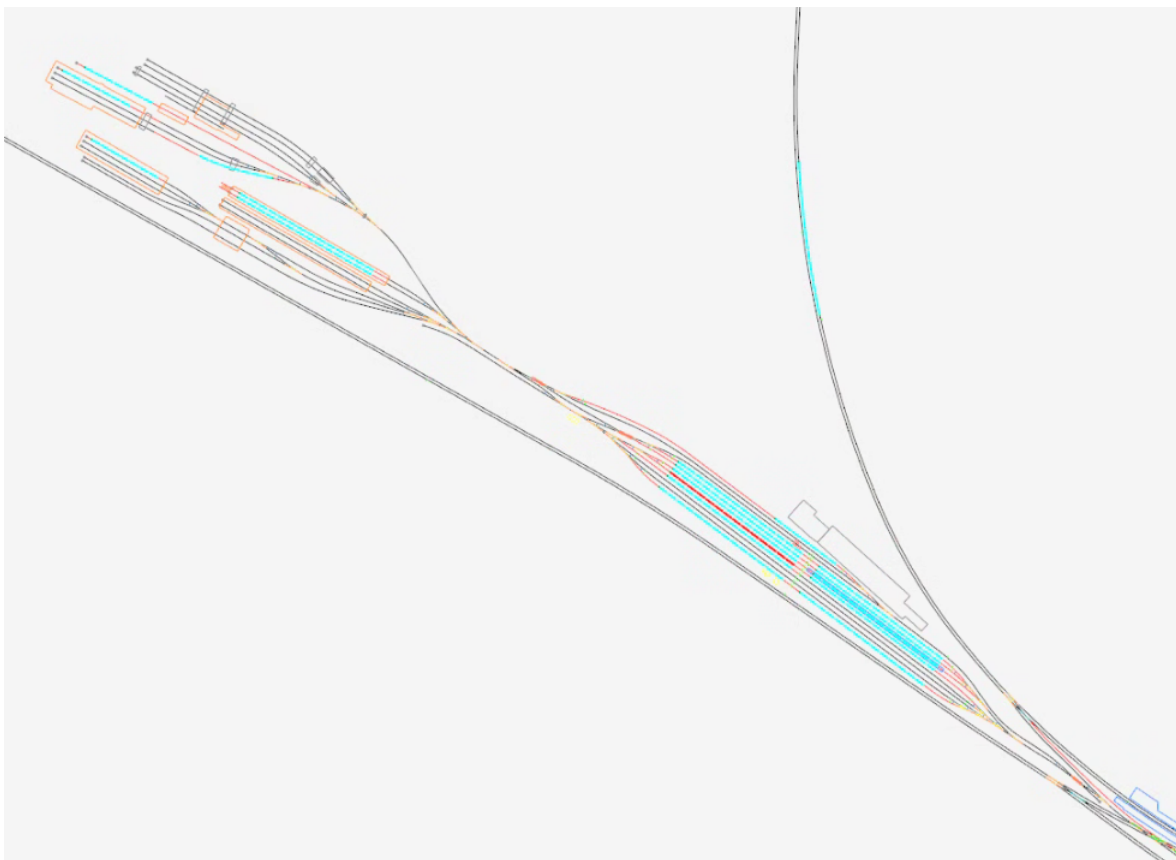
- presúvajúci sa - presúva sa medzi dvoma vlakmi,
- vracajúci sa - presúva sa od vlaku na stanovište,
- mimo práce - nie je v práci.

Pri poskytovaní podpory rozhodovania v danom údržbovom depe je potrebné zaoberať sa optimalizáciou priradovania rušnovodičov jednotlivým vlakom, ktoré je potrebné presúvať v rámci údržbového depa. Príchod vlakov do uzla je v rámci tejto úlohy zanedbaný a úloha sa zameriava výhradne na presuny v rámci železničného uzla medzi jednotlivými miestami, kde je potrebné vykonať údržbu. Vlak sa presúvajú relatívne krátke vzdialenosti, avšak vzhľadom k tomu, že v depe nemôžu prekročiť obmedzenú rýchlosť, tieto presuny v mnohých prípadoch trvajú významne dlhý časový úsek. Tieto časové úseky však nie je možné skrátiť, a preto sa ako priestor na optimalizáciu javia vzdialenosti, ktoré musia jednotliví rušnovodiči prejsť, keď sa presúvajú od jedného vlaku k druhému.

11.1.1 Použitie na simulácii založeného optimalizačného algoritmu

Výber sofistikovanej metódy

Po tom, ako bola definovaná vybraná úloha, prichádza na rad výber sofistikovanej metódy. V rámci overenia navrhovaného postupu sme sa rozhodli pre túto úlohu použiť exaktný prístup, avšak vzhľadom



Obr. 11.2: Simulačný model údržbového depa v softvéri Villon

na rozsah danej úlohy bol tento exaktný prístup použitý v rámci optimalizačného algoritmu založeného na simulácii. Výskumy popísané v kapitole 10.1 ukázali, že najmä v počiatočných fázach riešenia optimalizačného problému je vhodné zostaviť matematický model a pokúsiť sa nájsť optimálne riešenie. Pre potreby použitia optimalizačného algoritmu bol preto vytvorený matematický model, popisujúci problém priradenia rušňovodičov v údržbovom depe. Tento matematický model bol inšpirovaný všeobecne známym matematickým modelom, ktorý rieši úlohu okružných jász.

Účelová funkcia

$$\min \sum_{r=1}^R \sum_{i=1}^M \sum_{j \in F(i)} d_{ij} x_{ijr} \quad (11.1)$$

Účelová funkcia 11.1 minimalizuje vzdialenosti, ktoré musia rušňovodiči prejsť medzi vlakmi, ktoré obsluhujú (presúvajú v rámci železničného uzla).

Štruktúrne podmienky

$$\sum_{r=1}^R \sum_{i \in B(j)} x_{ijr} = 1 \quad \text{pre } j = 2 \dots M \quad (11.2)$$

$$\sum_{i \in B(j)} x_{ijr} = \sum_{k \in F(j)} x_{jkr} \quad \text{pre } j = 1 \dots M, r = 1 \dots R \quad (11.3)$$

$$\sum_{j \in F(1)} x_{1jr} \leq 1 \text{ pre } r = 1 \dots R \quad (11.4)$$

$$t_i^p \leq y_i \text{ pre } i = 2 \dots M \quad (11.5)$$

$$(y_i + t_i^o + d_{ij}) \leq y_j + M(1 - x_{ijr}) \text{ pre } i = 2 \dots M, j \in F(i), r = 1 \dots R \quad (11.6)$$

$$d_{1j} + t_r^z \leq y_j + M(1 - x_{1jr}) \text{ pre } j = 2 \dots M, r = 1 \dots R \quad (11.7)$$

$$t_r^k + M(1 - x_{i1r}) \geq y_i \text{ pre } i \in B(1), r = 1 \dots R \quad (11.8)$$

Podmienka 11.2 vyjadruje, že každý presun bude realizovaný práve jedným rušňovodičom. Zachovanie toku jednotlivých činností je zabezpečené v podmienke 11.3. Podmienka 11.4 vyjadruje skutočnosť, že každý rušňovodič môže v jednom čase obsluhovať len jeden vlak. Podmienka 11.5 vyjadruje, že začiatok aktivity i nemôže byť realizovaný skôr ako vlak, na ktorom je táto aktivita vykonávaná, príde do systému. V podmienke 11.6 je vyjadrený fakt, že ak obslužné aktivity i a j nasledujú za sebou a sú realizované jedným rušňovodičom, tak začiatok aktivity j nemôže byť skôr ako skončí aktivita i plus čas, ktorý trvá presun medzi vlakmi prislúchajúcimi k aktivitám i a j v prípade, že sa jedná o dva rôzne vlaky. Ak aktivity i a j nenasledujú za sebou alebo sú realizované rôznymi rušňovodičmi, táto podmienka je uvoľňovaná pomocou konštanty N . Podmienka 11.7 zabezpečuje dodržanie času začiatku pracovnej doby rušňovodiča a podmienka 11.8 zabezpečuje dodržanie času skončenia jeho pracovnej doby.

Obligatórne podmienky

$$y_i \geq 0 \text{ pre } i = 1 \dots M \quad (11.9)$$

$$x_{ijr} \in \{0, 1\} \text{ pre } i = 1 \dots M, j = 1 \dots M, r = 1 \dots R \quad (11.10)$$

Podmienka 11.9 definuje rozsah hodnôt, ktoré môže nadobúdať premenná y_i a podmienka 11.10 vyjadruje binárnosť rozhodovacích premenných x_{ijr} .

Popis konštánt a premenných

- M - počet všetkých obslužných činností realizovaných na vlakoch,
- R - počet všetkých rušňovodičov,
- B(i) - predchodcovia obslužnej činnosti i ,
- F(i) - nasledovníci obslužnej činnosti i ,
- y_i - čas začiatku obsluhy vlaku i ,
- x_{ijr} - binárna rozhodovacia premenná, či vlak i bude obsluhovaný pred vlakom j rušňovodičom r ,
- t_i^p - čas príchodu vlaku i (najskorší možný čas začiatku obsluhy vlaku i),
- t_i^o - doba obsluhy vlaku i ,
- t_r^z - začiatok pracovnej doby rušňovodiča r ,
- t_r^k - koniec pracovnej doby rušňovodiča r ,
- d_{ij} - vzdialenosť medzi miestom, kde rušňovodič opúšťa vlak i a miestom, kde začína obsluhu vlaku j .

Pomocou navrhovaného matematického modelu je možné vykonať presné výpočty a vyriešiť problém priradenia jednotlivých rušňovodičov. Riešenie celého problému pomocou exaktného prístupu je však časovo náročné, a preto nevhodné na podporu rozhodovania pri operatívnom rozhodovaní v reálnom čase. Vzhľadom na túto skutočnosť sme navrhli optimalizačný algoritmus založený na simulácii. Tento prístup je založený na možnosti redukovať stavový priestor (a tým aj výpočtový čas) obmedzením počtu budúcich úloh, ktoré sa berú do úvahy stanovením časového limitu (nazývaného horizont).

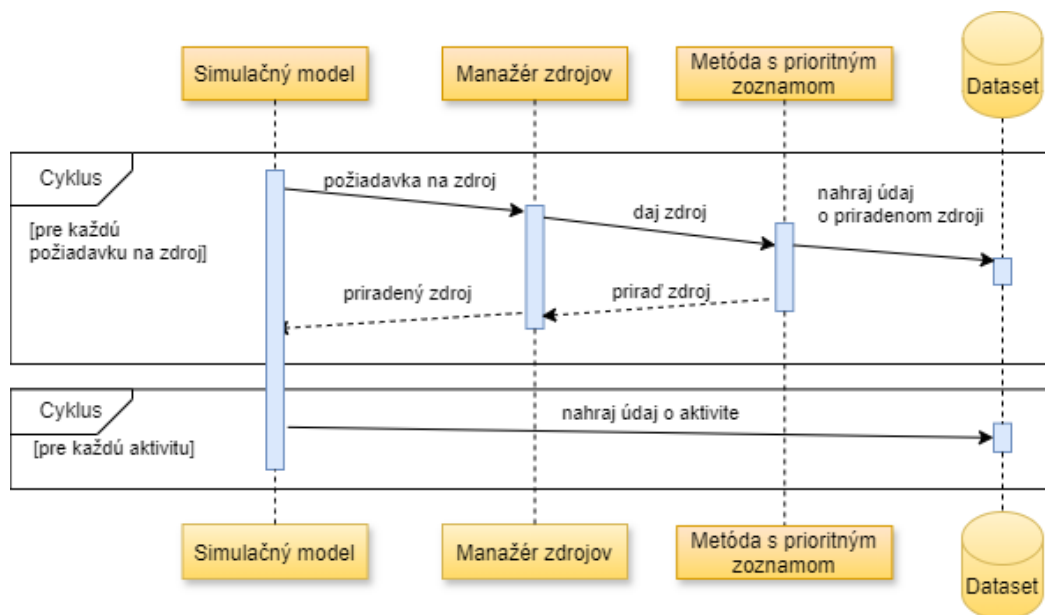
Na vykonanie navrhovaného algoritmu je potrebné vykonať inicializačný simulačný beh. Tento beh je dokončený pred začatím optimalizácie a využíva určitý typ jednoduchého algoritmu na alokáciu každého zdroja, ktorý je štandardný v simulačných modeloch tohto typu a zvyčajne je založený na implementácii prioritný zoznam.

Počas tohto simulačného behu sa vytvorí inicializačný dataset, ktorý obsahuje jednotlivé obslužné aktivity, ktoré sa vykonávajú v simulačnom modeli, ako aj pracovné časy jednotlivých rušňovodičov. Následne sa vykoná simulačný beh s použitím optimalizačnej metódy (v našom prípade s použitím exaktného výpočtu pomocou matematického modelu). Počas tohto behu sa každá požiadavka na priradenie zdroja (rušňovodiča) k vlaku spracuje nasledovne. Najprv sa odošle správa *daj zdroj* (obsahujúca požiadavku) optimalizačnému modulu. Keď optimalizačný modul (vrátane optimalizačných metód) prijme túto správu, vytvorí čiastkovú množinu údajov z inicializačnej množiny údajov na základe veľkosti zvoleného časového horizontu. Napríklad, ak je časový horizont 3 hodiny, čiastková množina údajov pozostáva zo všetkých obslužných činností, ktoré sa vyskytnú v priebehu troch hodín od príslušnej požiadavky. Tým sa, samozrejme, výrazne zmenší veľkosť údajov a zníži sa výpočtový čas použitých algoritmov. Optimalizačný modul potom vykoná výpočet na základe tohto čiastkového súboru a určí najvhodnejší zdroj a odošle správu s týmto zdrojom do simulácie. V simulačnom modeli je zdroj priradený požiadavke, požiadavka sa považuje za splnenú a simulačný beh pokračuje, kým nevznikne ďalšia požiadavka na priradenie zdroja. Tento cyklus sa opakuje až do ukončenia simulačného behu. Na obrázkoch 11.3 a 11.4 je znázornený sekvenčný diagram realizácie požiadaviek v navrhovanom algoritme.

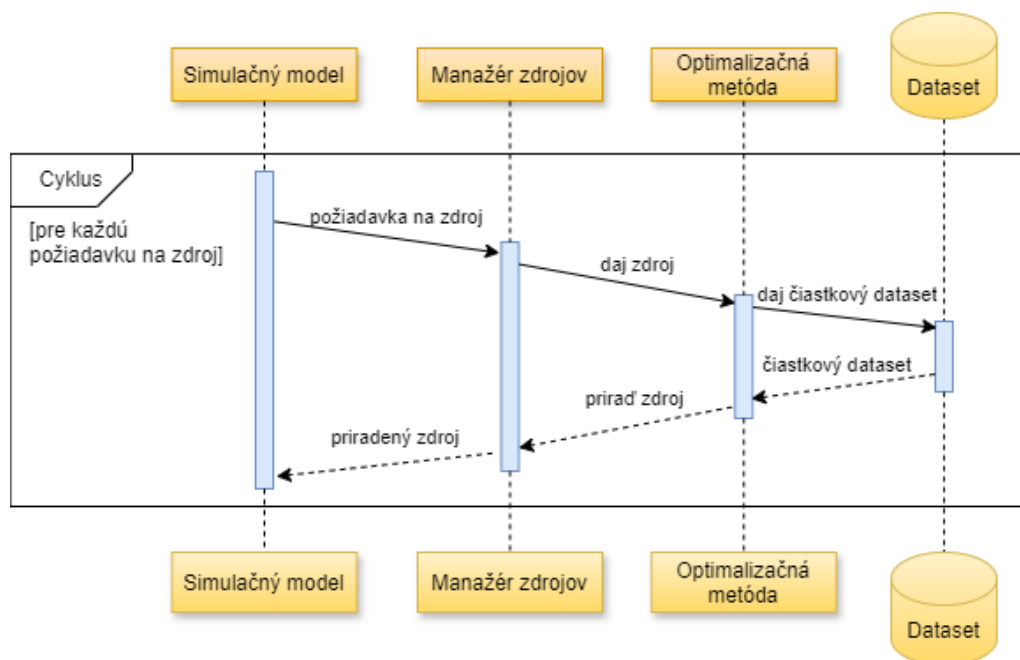
Definovanie dát, potrebných pre použitie vybranej metódy

Na získanie údajov zo simulačného modelu bol vykonaný inicializačný simulačný beh, v ktorom boli jednotliví rušňovodiči priradení k vlakom pomocou funkcie „First Fit“ (aj keď simulačný softvér Villon poskytuje niekoľko algoritmov na priradenie personálu, v tomto prípade sa použila funkcia „First Fit“, pretože algoritmus použitý v skutočnom depe nebol pri vývoji modelu známy). Táto funkcia vždy vyberie prvého rušňovodiča, ktorý je v dostupnom stave. Ak v tomto stave nie je žiadny rušňovodič, vlak čaká, kým jeden z rušňovodičov bude k dispozícii.

Súbor údajov vytvorený v inicializačnom simulačnom behu pozostáva z jednotlivých obslužných činností (pohyb vlaku z bodu A do bodu B) a rušňovodičov. Jednotlivé obslužné činnosti (presuny vlakov) sa vzťahujú na konkrétny vlak. Časové zmeny jednotlivých rušňovodičov sú modelované ako noví rušňovodiči,



Obr. 11.3: Sekvenčný diagram inicializačného behu



Obr. 11.4: Sekvenčný diagram optimalizačného algoritmu

hoci v skutočnosti pracuje jeden rušňovodič vo viacerých zmenách. Pre 7-dňový simulačný beh (7-dňová prevádzka v údržbovom depe) bol vytvorený základný súbor údajov s 207 činnosťami a 84 rušňovodičmi, čo však neznamená 84 rušňovodičov, ale 84 jednotlivých pracovných zmien.

Implementácia na simulácii založenej optimalizačnej metódy

Výpočty pomocou matematického modelu s použitím tohto súboru údajov boli vykonané pomocou štandardného IP riešiteľa Gurobi[24]. Celá metóda bola implementovaná v jazyku Java.

Definovanie komunikačného rozhrania

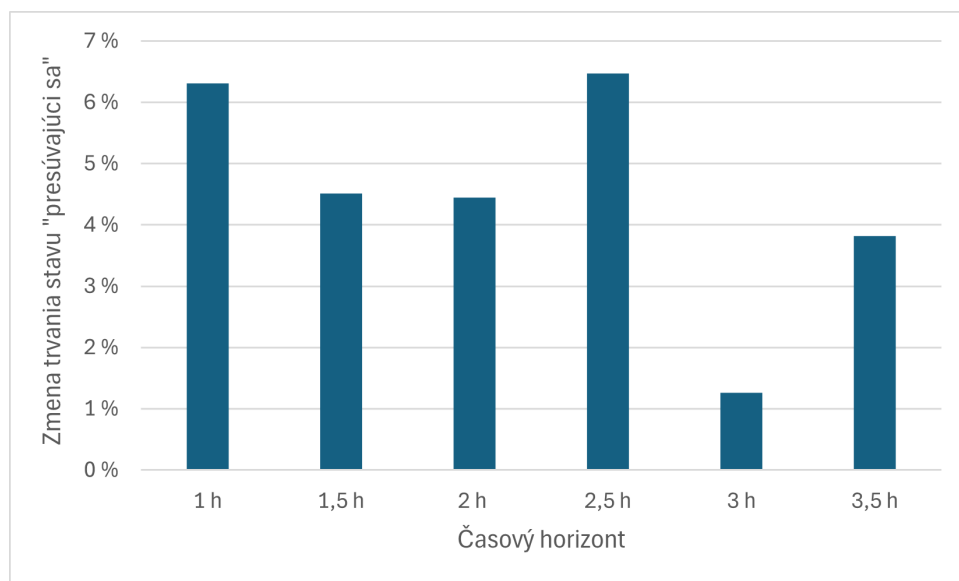
Optimalizačný algoritmus bol v tomto prípade implementovaný ako externá aplikácia a to najmä z dôvodu, že bol použitý štandardný IP riešiteľ, preto bolo v tomto kroku navrhnuté komunikačné rozhranie, ktoré umožňuje komunikáciu po sieti priamym spojením simulačného modelu a externej aplikácie za pomoci socketov. Správy sa posielajú vo formáte JSON.

Overenie funkčnosti navrhovaného riešenia

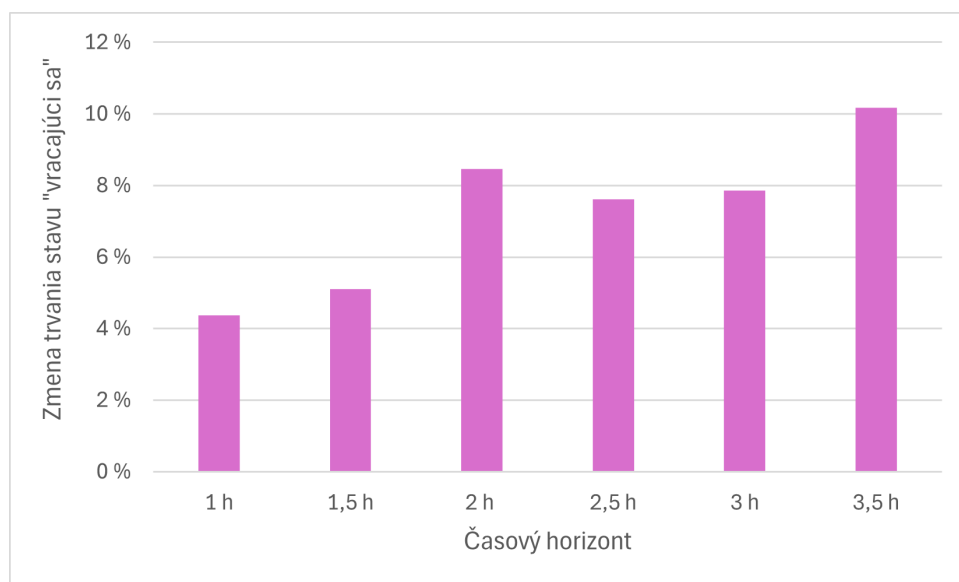
Cieľom použitia navrhovaného algoritmu pomocou matematického modelu (minimalizáciou jeho účelovej funkcie) je skrátiť trvanie stavov „vracajúci sa“ a „presúvajúci sa“, pretože tieto priamo súvisia so vzdialenosťami, ktoré musí rušňovodič prejsť pešo medzi vlakmi. Zníženie trvania týchto presunov má priamy vplyv na trvanie obsluhy vlakov. Sekundárnym účinkom optimalizácie by mohlo byť zvýšenie trvania stavu „dostupný“ a zníženie trvania stavu „pracujúci“ a „čakajúci“. Súčasne, musí byť zachovaný výpočtový čas, počas ktorého sa môže rozhodovanie v depe realizovať. V rámci nášho výskumu sa za hraničnú hodnotu pre výpočtový čas považuje 1 hodina.

Experimenty sa uskutočnili s rôznymi dĺžkami časového horizontu (1 hodina, 1,5 hodiny, 2 hodiny, 2,5 hodiny, 3 hodiny a 3,5 hodiny) pre 7-dňový simulačný beh s cieľom preskúmať význam vhodných nastavení časového výhľadu. Čiastkové datasety obsahovali v priemere 14 činností a 10 rušňovodičov. V tabuľke 11.1 sú uvedené hodnoty najdôležitejších stavov „presúvajúci sa“ a „vracajúci sa“, ostatné stavy sa menili nevýznamne.

Keďže simulačný model použitý na vykonanie experimentov je deterministický, experimenty boli vykonané v rámci jedného simulačného behu. Výsledky získané pomocou modelu používajúceho First Fit (pôvodne použitá funkcia) a výsledky získané pomocou navrhovaného prístupu s daným časovým horizontom boli porovnané určením času, o koľko sa trvanie každého stavu znížilo (zvýšilo) v percentách. Na obrázkoch 11.5 a 11.6 sú znázornené rozdiely medzi funkciou First Fit a navrhovaným optimalizačným algoritmom s rôznym časovým horizontom vyjadrené v percentách. Ako vidieť, použitie navrhovaného algoritmu prinieslo zlepšenie v porovnaní s funkciou First Fit pre obidva pozorované stavy. Na základe výpočtu Pearsonovho korelačného koeficientu sa skúmala korelácia medzi trvaním stavu návratu a veľkosťou časových horizontov, ale takáto korelácia sa nezistila pre stav „presúvajúci sa“. Je to pravdepodobne



Obr. 11.5: Percentuálna zmena stavu „presúvajúci sa“



Obr. 11.6: Percentuálna zmena stavu „vracajúci sa“

Tabuľka 11.1: Porovnanie výsledkov experimentov s rôznym časovým výhľadom

	Presúvajúci sa (hod.)	Vracajúci sa (hod.)
Prvý vhodný	30,12	21,52
1 hodina	28,22	20,58
1,5 hodiny	28,76	20,42
2 hodiny	28,78	19,70
2,5 hodiny	28,17	19,88
3 hodiny	29,74	19,83
3,5 hodiny	28,97	19,33

preto, že časové rozdiely medzi časovými horizontmi nie sú dostatočne veľké. Experimenty však s väčšími časovými horizontmi nebolo možné vykonať, pretože výpočtový čas prekročil stanovenú hraničnú hodnotu. Tento problém by sa dal v budúcnosti odstrániť použitím heuristických metód namiesto exaktného prístupu.

11.1.2 Použitie algoritmov učenia posilňovaním

Výber algoritmu učenia posilňovaním

Iným prístupom, ktorý umožňuje poskytnúť riešenia definovaného výskumného problému je učenie posilňovaním. Tento prístup sme sa rozhodli použiť na základe predpokladu, že tento prístup je určený na riešenie komplexných problémov a mohol by dokázať vyriešiť problémy danej veľkosti.

Pre overenie navrhovaného postupu bol vybraný Proximal Policy Optimization (PPO) algoritmus. Tento algoritmus vychádza z algoritmu Simple Advantage Actor Critic (A2C). A2C algoritmus je tvorený dvomi sieťami, a to aktorom a kritikom. Aktor aproximuje politiku, to znamená, že určuje pravdepodobnosť s akou od určitého vstupu nastane daný výstup. Kritik aproximuje hodnotovú funkciu, to znamená, že určuje ako dobrá daná akcia je [30]. Nevýhodou tohto algoritmu je, že má sklon ku oscilácii. Aby sa predišlo tejto oscilácii, PPO funguje na princípe aplikácie malej vzorky skúseností na zmenu politiky, a tým zabezpečuje, že sa nová politika nebude príliš líšiť od tej predchádzajúcej. Vzhľadom na väčšiu stabilitu tréningu sa preto dá predpokladať, že PPO agent bude dobre použiteľný pri riešení vyššie popísanej úlohy.

Definovanie dát, potrebných pre použitie vybranej metódy

Vychádzajúc z postupu, ktorý je popísaný v metodike pre integráciu učenia posilňovaním boli definované tieto parametre:

- prostredie - zoznam času začiatku plánovaných aktivít (v prípade, že aktivita už prebehla, plánovaný čas začiatku je -1), poloha všetkých vlakov v systéme (počet týchto vlakov sa môže meniť, preto v prípade, že nie sú v systéme všetky vlaky, ktoré sa v ňom môžu v jednom čase nachádzať, je vektor

doplnený pomocou núl, tak aby dosiahol veľkosť, ktorá pokryje všetky vlaky), poloha rušňovodičov od vlaku, ktorému je potrebné priradiť rušňovodiča,

- akcie - priradený rušňovodič,
- odmeny/pokuty - sú priradené na konci simulačného behu na základe počtu obslužených vlakov, čím viac sa tento počet blíži k maximálnemu počtu obslužených vlakov, tým je odmena vyššia. Táto odmena je normalizovaná na intervale (0,1).

Vzhľadom na časovú náročnosť tréningu neurónovej siete, ktorú používa algoritmus učenia posilňovaním bol tréning realizovaný na trojdňovom simulačnom behu, v rámci ktorého bolo potrebné pridelovať rušňovodičov aktivitám presunu. V rámci tohto simulačného behu bolo plánovaných 92 aktivít presunu, maximálny počet vlakov, ktoré sa nachádzajú v jednom momente v systéme je 36, v simulačnom modeli sú traja rušňovodiči. Jednotlivé aktivity presunu, súradnice vlakov a vzdialenosť jednotlivých rušňovodičov od konkrétneho vlaku tvoria vektor popisujúci prostredie. Odmena bola nastavená na základe dokončených technológií jednotlivých vlakov - vlak odišiel obslužený zo systému. Maximálny počet takýchto obslužených vlakov je 65. Odmena bola nastavená nasledovne:

- ak bolo obslužených menej ako 10 vlakov, odmena je 0 (menej ako 10 obslužených vlakov sa počas realizácie experimentov nikdy nevyskytlo),
- ak bolo obslužených viac ako 10 a menej ako 65 vlakov, odmenu vypočítame vzťahom:
 $(x-10)/(65-10)$, kde x je počet obslužených vlakov,
- ak bolo obslužených 65 vlakov, odmena je 1.

Definovanie komunikačného rozhrania

Aby bolo možné použiť učenie posilňovaním, musí simulačný model spĺňať požiadavky na použitie učenia posilňovaním, to znamená, že musí byť schopný komunikovať s externou aplikáciou, ktorá realizuje výpočty pomocou algoritmu učenia posilňovaním (agentom). Použitý simulačný model bol rozšírený o možnosť komunikácie s externými modulmi a preto bolo možné využiť simulačný model železničného dopravného uzla tak, ako to požaduje algoritmus učenia posilňovaním.

Algoritmus učenia posilňovaním má slúžiť ako podporný prvok pre rozhodovací proces dispečera, a teda sa predpokladá, že po natrénovaní dokáže takýto agent poskytnúť návrh rozhodnutia pri pridelení zdroja, konkrétne rušňovodiča v údržbovom depe.

Implementácia algoritmu učenia posilňovaním

Pre overenie navrhovaného riešenia bol vybraný PPO agent, teda jeden z algoritmov učenia posilňovaním.

Tento algoritmus bol implementovaný v jazyku Python, za použitia knižnice PyTorch.

Parametre PPO agenta boli nastavené nasledovne:

- počet epoch - 4,
- faktor zrážania gamma - 0,999,
- veľkosť kroku učenia alfa - 0,003,
- veľkosť dávky - 500.

Neurónová sieť použitá ako aktor je tvorená 170 neurónmi na vstupnej vrstve, následne dvomi vrstvami s 256 neurónmi a 3 neurónmi na výstupnej vrstve. Tieto výstupné neuróny určujú, ktorý rušňovodič má byť priradený danej úlohe. Neurónová sieť použitá ako kritik je tvorená 170 neurónmi na vstupnej vrstve, dvomi vrstvami s 256 neurónmi a jedným neurónom na výstupnej vrstve.

Nasadenie algoritmu učenia posilňovaním

Sofistikovaná metóda bola spolu so simulačným jadrom nasadená na štandardnom počítači, pričom na tomto počítači prebiehala aj prípravná fáza. Parametre zariadenia boli:

- RAM - 16 GB,
- Procesor - Intel(R) Core(TM) i5-9300HF CPU @ 2.40GHz 2.40 GHz.

Prípravná fáza (trénovanie)

Na tréning PPO algoritmu bolo realizovaných 15 000 epizód. Keďže trénovacia fáza prebieha pred nasadením systému a teda nemusí byť dosiahnutý nízky čas tréningu, tréning bol realizovaný na bežnom počítači a čas jednej epizódy bol priemerne 20 sekúnd. Tento tréning je zobrazený na grafe na obrázku 11.7.

Z grafu vyplýva, že algoritmus PPO je možné použiť pri realizácii rozhodnutí vo vyššie popísanej úlohe, nakoľko natrénovaný algoritmus dosahuje maximálnu odmenu a teda je zabezpečené obsluženie maximálneho počtu vlakov.

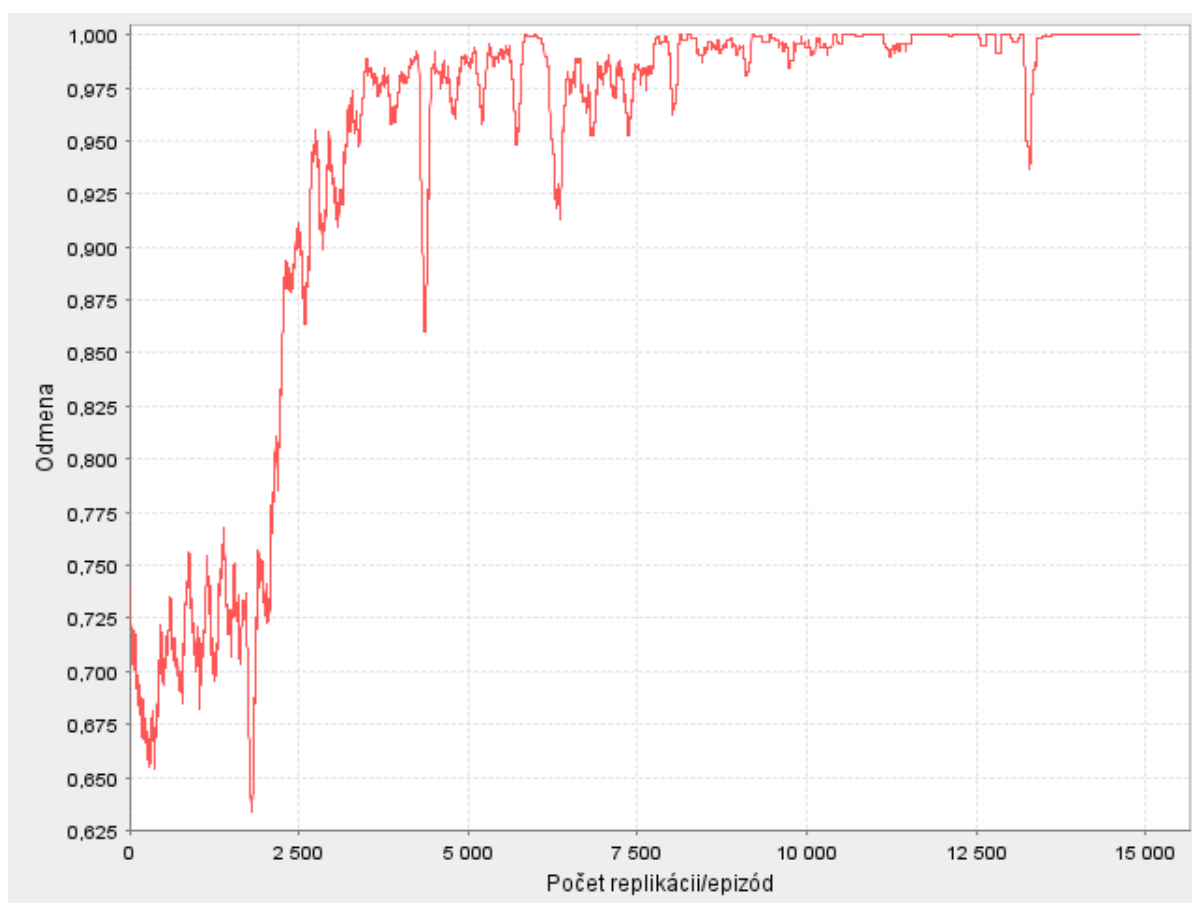
Realizovaný experiment bol vykonaný na inej konfigurácii simulačného modelu ako experiment popísaný v kapitole 11.1.1, a preto nie je vhodné porovnávať dosiahnuté výsledky. Na rovnakej konfigurácii simulačného modelu, na ktorej bol trénovaný PPO algoritmus bol použitý algoritmus First Fit. Tento však nedokázal dosiahnuť počet obslužených vlakov 65, dosiahol len 40 obslužených vlakov. Táto situácia nastala kvôli vzniku deadlocku (dva vlaky idú oproti sebe po jednej koľaji). Na daný problém bol použitý aj presný výpočet založený na matematickom modeli. Tento výpočet (najmä kvôli abstrakcii mnohých vlastností železničného uzla) tiež viedol k deadlockom. Je teda možné povedať, že PPO algoritmus dosahuje lepšie výsledky ako First Fit algoritmus.

Dosiahnuté výsledky ukazujú, že algoritmus PPO by mohol byť vhodný na riešenie úloh vo veľmi zložitom systéme, akým je železničný dopravný uzol. V takomto systéme je potrebné vyhodnotiť mnoho dôsledkov jednotlivých zmien, a to je možné pri použití algoritmov učenia posilňovaním. Použitelnosť

na širšom spektre úloh je však potrebné overiť na ďalších úlohách. Je dôležité uviesť, že vykonané experimenty sa uskutočnili v deterministickom prostredí. V stochastickom prostredí nedosahuje natrénovaný algoritmus dobré výsledky, a preto je potrebné rozšíriť proces trénovania na stochastické prostredie.

Overenie funkčnosti použitého algoritmu

Pri spustení simulačného behu s algoritmom, ktorý používa natrénovanú neurónovú sieť môžeme potvrdiť, že bol obslužený maximálny počet vlakov, čiže 65. Tento výsledok je zlepšením o 62,5 percenta v porovnaní s algoritmom First Fit.



Obr. 11.7: Tréningový proces algoritmu učenia posilňovaním

11.2 Úloha určenia poradia úloh

Pre dokázanie použiteľnosti sofistikovaných algoritmov v rámci systému na podporu rozhodovania s použitím simulačných modelov bola okrem úlohy priradovania rušňovodičov vybraná aj úloha určenia poradia vykonávania údržbových činností. Pre demonštráciu riešenia tejto úlohy bolo použité údržbové depo, ktorého simulačný model je zobrazený na obrázku 11.8.



Obr. 11.8: Simulačný model údržbového depa 2 v softvéri Villon

Vlaky, ktoré prichádzajú do údržbového depa majú vopred definovanú sériu údržbových činností, ktoré trvajú určitý čas. Každá údržbová činnosť sa môže realizovať len na určenom stanovišti. Ak na danom stanovišti prebieha údržba jedného vlaku, je samozrejmé, že v rovnakom čase nemôže prebiehať údržba iného vlaku na tom istom stanovišti. Vlaky majú určený čas odchodu a pre efektívnu prevádzku údržbového depa je potrebné minimalizovať počty vlakov, ktoré v tomto čase nie sú schopné odísť, pretože na nich ešte prebieha údržba. Aby bolo možné tieto počty vlakov minimalizovať, dôležité sa zaoberať poradím, v akom budú jednotlivé činnosti vykonávané na jednotlivých stanovištiach.

11.2.1 Použitie na simulácii založeného optimalizačného algoritmu

Výber sofistikovanej metódy

Pre riešenie úlohy určenia poradia obslužných úloh sme sa rozhodli použiť nami vytvorený optimalizačný algoritmus založený na simulácii, ktorý podrobný popis sa nachádza v kapitole 11.1.1.

Definovanie dát, potrebných pre použitie vybranej metódy

Aby mohol byť použitý navrhovaný algoritmus, musia byť definované jednotlivé dáta, ktoré bude potrebné získať zo simulačného modelu. Tieto dáta sú popísané v rámci matematického modelu vybranej úlohy, ktorý bol zostavený. Tento model bude neskôr použitý, aj pri implementácii daného algoritmu.

Matematický model danej úlohy bol inšpirovaný riešením úlohy o batohu a bol zostavený nasledovne:

Účelová funkcia

$$\min \sum_{i=1}^N c_i y_i \quad (11.11)$$

Účelová funkcia 11.11 minimalizuje časy začiatkov všetkých obslužných aktivít. Hodnota účelovej funkcie je závislá aj na hodnote koeficientov c_i . Koeficient c vyjadruje prioritu aktivity a táto priorita sa zvyšuje v závislosti na plánovanom čase odchodu daného vlaku (čím skorší odchod vlaku, tým vyššia priorita).

Štrukturálne podmienky

$$y_i \geq a_t \quad \text{pre } t = 1 \dots T, i \in C(t) \quad (11.12)$$

$$y_i \leq l + M(1 - v_i) \quad \text{pre } t = 1 \dots T, i \in C(t) \quad (11.13)$$

$$\begin{aligned} y_i + d_i &\leq y_j + M(3 - v_i - v_j - x_{ij}) \\ \text{pre } t &= 1 \dots T, i \in C(t), j \in C(t) \end{aligned} \quad (11.14)$$

$$\begin{aligned} y_j + d_j &\leq y_i + M(2 - v_i - v_j + x_{ij}) \\ \text{pre } t &= 1 \dots T, i \in C(t), j \in C(t) \end{aligned} \quad (11.15)$$

$$\begin{aligned} y_i + d_i &\leq y_j \\ &+ M(5 - v_i - v_j - x_{ij} - z_{ip} - z_{jp}) \\ \text{pre } p &= 1 \dots P, i \in P(p), j \in P(p) \end{aligned} \quad (11.16)$$

$$\begin{aligned} y_j + d_j &\leq y_i \\ &+ M(4 - v_i - v_j + x_{ij} - z_{ip} - z_{jp}) \\ \text{pre } p &= 1 \dots P, i \in P(p), j \in P(p) \end{aligned} \quad (11.17)$$

$$\sum_{p \in P} z_{ip} = 1 \quad \text{pre } i = 1 \dots T, t \in C(t) \quad (11.18)$$

$$\begin{aligned} y_i &\geq b_p - M(2 - z_{ip} - v_i) \\ \text{pre } i &= 1 \dots T, p = 1 \dots P \end{aligned} \quad (11.19)$$

Podmienka 11.12 vyjadruje, že začiatok aktivity i nemôže nastať pre príchodom vlaku t do údržbového depa (ak táto aktivita je realizovaná na tomto vlaku). Keďže je navrhovaný algoritmus založený na čiastkových datasetoch ohraňovaných časovým výhľadom, podmienka 11.13 vyjadruje, že začiatok aktivity i musí byť pred koncom časového výhľadu, ak aktivita i bude v tomto časovom okne vykonaná (keď používame prístup s časovými oknami, nie všetky aktivity z čiastkového datasetu musia byť realizované). Podmienky

11.14 a 11.15 vyjadrujú, že súčet času začiatku aktivity i a trvania tejto aktivity i musí byť nižší ako čas začiatku aktivity j , ak obe aktivity sú realizované a aktivita i je vykonaná pred aktivitou j na rovnakom vlaku a vice versa. Podmienka 11.16 a 11.17 hovorí, že dve aktivity sú realizované na rovnakom stanovišti a aktivita i je vykonaná pred aktivitou j , potom súčet času začiatku aktivity i a jej trvania musí byť pred začiatkom aktivity j , a vice versa. Podmienka 11.18 vyjadruje, že každá aktivita musí byť priradená stanovištu, kde bude vykonaná. Podmienka 11.19 vyjadruje, že ak aktivita i je vykonaná na stanovišti p , potom čas začiatku aktivity i musí byť väčší ako najskorší možný čas začiatku realizovania úloh na stanovišti p .

Obligatórne podmienky

$$y_i \geq 0 \text{ pre } i = 1 \dots T \quad (11.20)$$

$$x_{ij} \in \{0,1\} \text{ pre } i = 1 \dots T, \quad j = 1 \dots T \quad (11.21)$$

$$v_i \in \{0,1\} \text{ pre } i = 1 \dots T \quad (11.22)$$

$$z_{ip} \in \{0,1\} \text{ pre } i = 1 \dots T, \quad p = 1 \dots P \quad (11.23)$$

Obligatórne podmienky definujú rozsah jednotlivých premenných: podmienka 11.20 definuje rozsah hodnôt, ktoré môže nadobúdať premenná y_i a podmienky 11.21, 11.22, 11.23 vyjadrujú binárnosť rozhodovacích premenných x_{ij} , v_i a z_{ip} .

V tomto modeli bolo definovaných niekoľko premenných a konštánt.

x_{ij} – binárna rozhodovacia premenná rozhodujúca či aktivita i je realizovaná pred aktivitou j .

y_i – premenná nadobúdajúca kladné hodnoty a vyjadruje skutočný čas začiatku aktivity i .

v_i – binárna rozhodovacia premenná, ktorá určuje, či aktivita i je vykonávaná alebo nie. Toto je potrebné, pretože sú použité časové výhľady a nie každá aktivita musí byť realizovaná.

z_{ij} – binárna rozhodovacia premenná, rozhodujúca či aktivita i je realizovaná na stanovišti j .

$C(t)$ – zoznam aktivít na vlaku t .

$P(p)$ – zoznam aktivít, ktoré môžu byť vykonané na stanovišti p .

d_i – konštanta reprezentujúca trvanie aktivity i .

a_t – čas príchodu vlaku t , najskorší bod v čase, v ktorom môžu byť vykonané aktivity na vlaku t .

b_p – najskorší čas začiatku aktivít na stanovišti p .

c_i – koeficient vyjadrujúci prioritu aktivity i , závislý na plánovanom čase odchodu vlaku, na ktorom je aktivita i realizovaná.

l – súčasný časový výhľad.

M – konštanta, definovaná ako veľmi veľké číslo, použitá k uvoľňovaniu podmienok.

P – počet stanovišť.

T – počet aktivít.

Definovanie spôsobu získavania dát

V rámci inicializačného behu boli zo simulačného modelu pomocou funkcie "First Fit" získané potrebné údaje. Celkový dataset zahŕňa 15 vlakov a 63 údržbových aktivít, ktoré je potrebné vykonať.

Pre realizáciu danej metódy boli zozbierané všetky údaje, ktoré sú potrebné pre riešenie matematického modelu, to znamená počet vlakov, počet stanovišť, reálne príchody vlakov, plánované odchody vlakov, zoznam úloh na vlaku, zoznam úloh, ktoré je možné vykonať na danom stanovišti.

Definovanie komunikačného rozhrania

Ako už bolo spomenuté, táto optimalizačná metóda je implementovaná ako externá aplikácia a to najmä z dôvodu, že bol použitý štandardný riešiteľ, preto bolo v tomto kroku navrhnuté komunikačné rozhranie, ktoré umožňuje komunikáciu po sieti priamym spojením simulačného modelu a externej aplikácie za pomoci socketov. Správy sa posielajú vo formáte JSON.

Implementácia na simulácii založenej optimalizačnej metódy

Výpočty pomocou matematického modelu s použitím tohto súboru údajov boli vykonané pomocou štandardného riešiteľa IP Gurobi [24]. Celá metóda bola implementovaná v jazyku Java.

Overenie funkčnosti vybranej metódy

Pre overenie použiteľnosti navrhovaného riešenia boli realizované experimenty s rôznym časovým výhľadom. Boli teda použité čiastkové datasety, ktoré boli vytvorené pomocou rôznych časových horizontov od 1 hodiny do 3,5 hodiny. Čiastkové datasety obsahovali v priemere 7 vlakov a 30 úloh. Je dôležité poznamenať, že presný výpočet pracuje s čiastkovým súborom údajov obmedzeným konkrétnym časovým horizontom, a preto nie je potrebné zahrnúť všetky servisné úlohy.

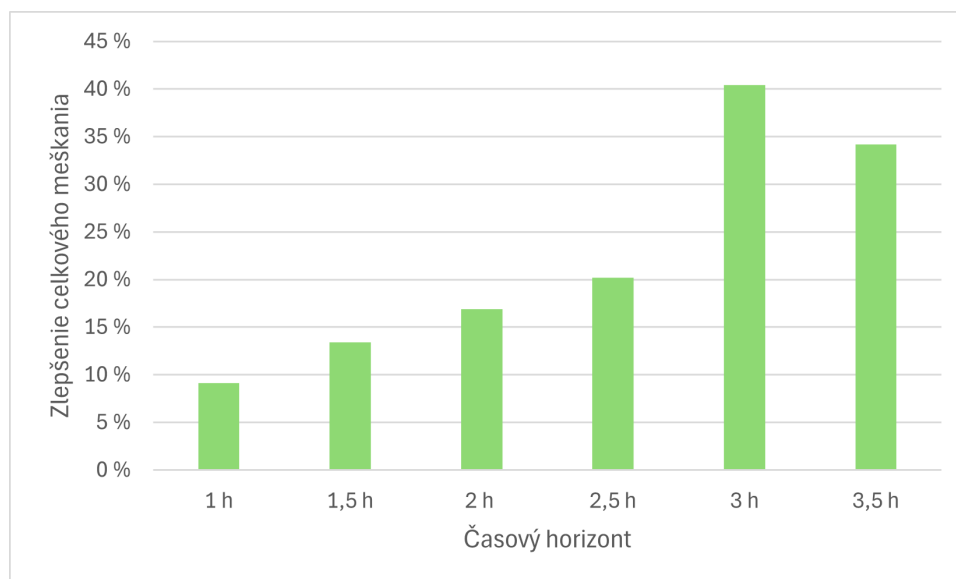
Výsledky experimentov sú uvedené v nasledujúcej tabuľke. Celkový čas meškania zahŕňa aj meškania vlakov, ktoré v príslušný deň neodišli. Oneskorenie nedokončeného vlaku sa vypočíta ako rozdiel medzi plánovaným časom odchodu a časom ukončenia simulácie.

Ako je možné vidieť, navrhovaný algoritmus priniesol zlepšenie oproti algoritmu "First Fit". Aby bolo možné posúdiť mieru tohoto zlepšenia, bola určená percentuálna zmena a táto zmena pre jednotlivé časové horizonty je znázornená na obrázku 11.9.

Z tohto obrázku je vidieť, že medzi zvyšujúcim sa časovým výhľadom a kvalitou riešenia je závislosť, ktorá bola potvrdená pomocou určenia Pearsonovho koeficientu. Na základe výsledkov experimentov popísaných v tejto kapitole môžeme povedať, že použitie navrhovanej optimalizačnej metódy založenej na simulácii ako podporného nástroja v rámci systému na podporu rozhodovania v reálnom čase je možné.

Tabuľka 11.2: Celkové meškanie pri použití rôznych časových horizontov

Dĺžka časového horizontu	Meškanie (minúty)	Počet nevypravených vlakov
First Fit	5234	4
1 hod.	4755	4
1.5 hod.	4533	4
2 hod.	4350	4
2.5 hod.	4176	3
3 hod.	3120	3
3.5 hod.	3445	2



Obr. 11.9: Percentuálna zmena súčtu oneskorení

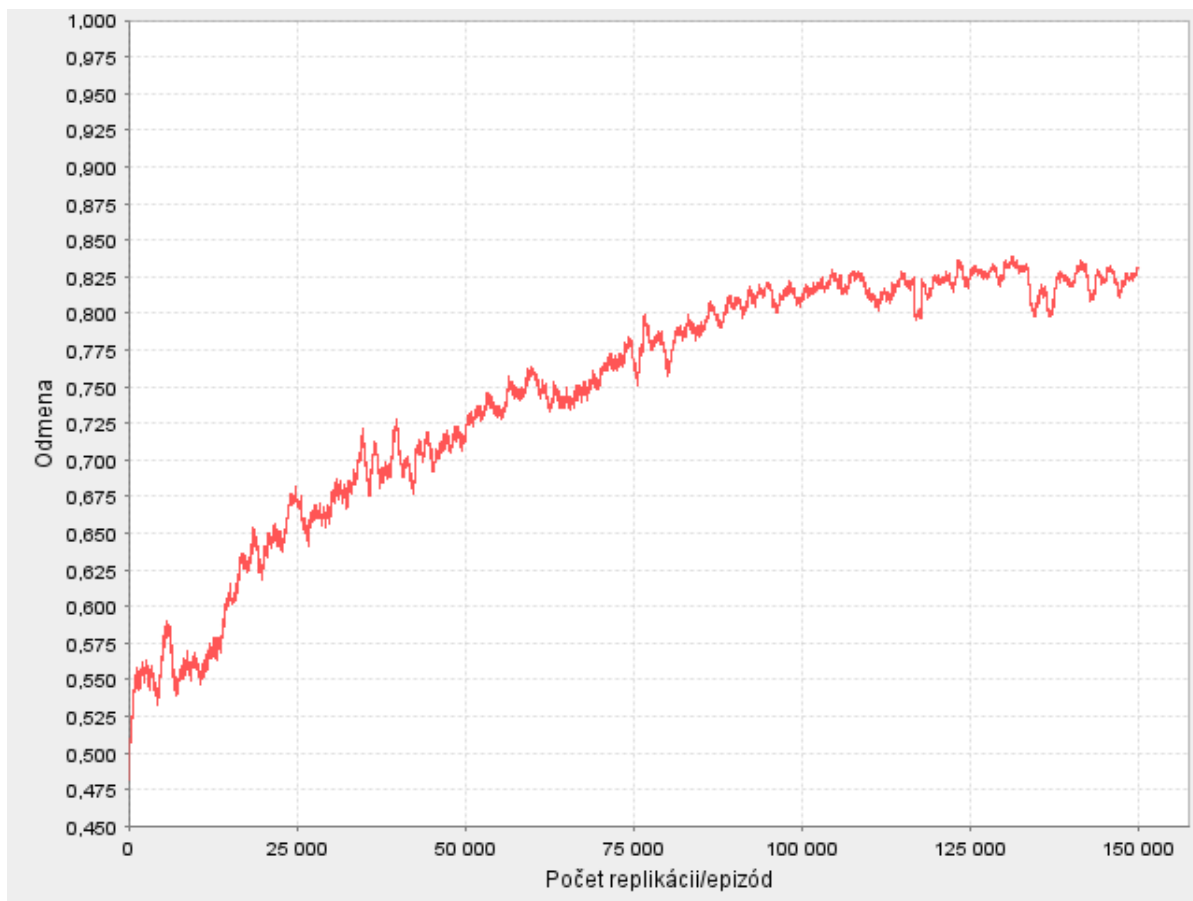
11.2.2 Použitie algoritmov učenia posilňovaním - určenie poradia úloh

Výber algoritmu učenia posilňovaním

Podobne ako pri riešení úlohy priradenia personálu, aj v tomto prípade sme sa rozhodli navrhnúť použitie algoritmov učenia posilňovaním. Najprv sme skúšali realizovať experimenty na deterministickom systéme, tu sme pri rovnakej konfigurácii algoritmu ako pri riešení úlohy priradenia personálu dosahovali výsledky, uvedené na grafe na obrázku 11.10.

Okrem použitia PPO algoritmu, sme sa rozhodli na tento systém použiť aj A2C algoritmus, pri použití ktorého boli najlepšie výsledky dosahované s parametrami:

- faktor zrážania gamma - 0,998,

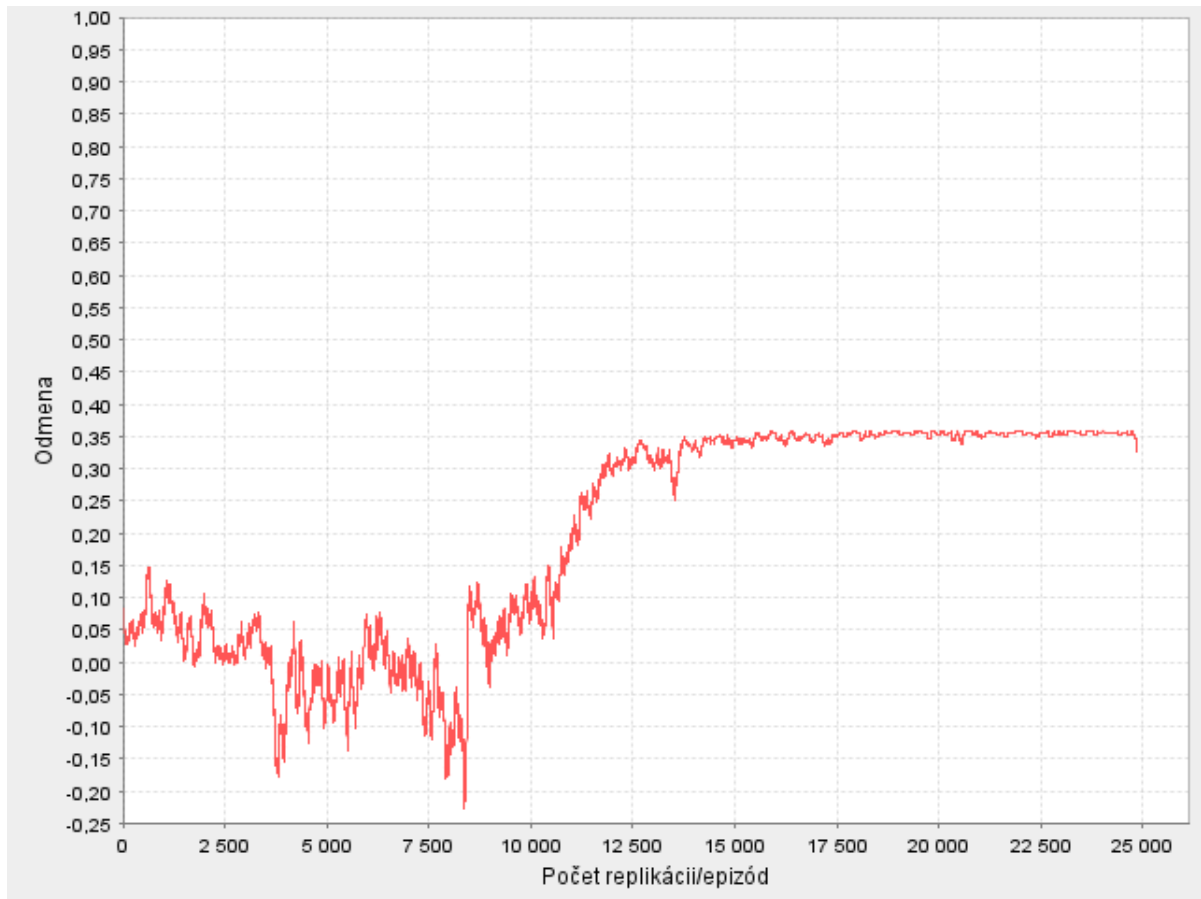


Obr. 11.10: Tréning PPO agenta v deterministickom prostredí

- veľkosť kroku učenia α - 0,001,
- veľkosť dávky - 1000.

Veľkosť použitej siete pre aktora je 202 neurónov na vstupnej vrstve, následne jedna vrstva so 64 neurónmi, jedna vrstva s 32 neurónmi a 2 neuróny na výstupnej vrstve. Kritik je tvorený 202 neurónmi na vstupnej vrstve, následne jednou vrstvou so 128 neurónmi, jednou vrstvou s 64 neurónmi a jedným neurónom na výstupnej vrstve. Výsledky tohto tréningu zobrazuje graf na obrázku 11.11 a bližšie sú popísané v bakalárskej práci [40].

Keďže však reálny systém nie je deterministický, zamerali sme sa aj na použitie PPO algoritmu v stochastickom prostredí.



Obr. 11.11: Tréning A2C agenta v deterministickom prostredí

Definovanie dát potrebných pre použitie algoritmu

Pre použitie algoritmu učenia posilňovaním bolo potrebné pre vybranú úlohu definovať nasledujúce parametre:

- prostredie

- index aktivity, o ktorej sa aktuálne rozhoduje,
 - plánovaný odchod vlaku,
 - trvanie jednotlivých aktivít vlaku, o ktorom rozhodujeme,
 - najskorší možný čas začiatku jednotlivých aktivít vlaku, o ktorom rozhodujeme,
 - časy vyjadrujúce, ako dlho budú obsadené jednotlivé koľaje,
 - trvanie aktivít plánovaných na vlakoch, ktoré sa nachádzajú v modeli,
 - najskorší možný čas začiatku aktivít plánovaných na vlakoch, ktoré sa nachádzajú v modeli,
 - aktuálny simulačný čas,
- akcie - rozhodnutie o tom, či bude daná úloha realizovaná,
 - odmeny / pokuty - sú priradované na konci simulačného behu na základe celkového meškania vlakov, pričom táto hodnota je ešte upravená o meškania vlakov, na ktorých nebola dokončená údržba v rámci simulačného behu. Táto odmena je normalizovaná v rozmedzí 0 a 1. V deterministickom systéme bola odmena určená pomocou vzorca $\text{odmena} = (1 - (\text{súhrnné meškanie}) / 3) * 0,5 - \text{nevypravené vlaky} * 0,2$. Pri stochastickom systéme, sa však tento výpočet ukázal ako nedostačujúci, a preto sme ho modifikovali na výpočet v tvare: $\text{odmena} = (1 - (\text{súhrnné meškanie}) / 3) * 0,5 + (5 - \text{nevypravené vlaky}) * 0,1$.

Definovanie komunikačného rozhrania

Pre použitie navrhovaného prístupu bol použitý rovnaký komunikačný mechanizmus, ako pri riešení úlohy priradenia personálu v kapitole 11.1.2.

Implementácia algoritmu učenia posilňovaním

Pre overenie navrhovaného riešenia bol vybraný PPO agent, teda jeden z algoritmov učenia posilňovaním. Tento algoritmus bol implementovaný v jazyku Python, za použitia knižnice PyTorch. Parametre PPO agenta boli nastavené nasledovne:

- počet epoch - 4,
- faktor zrážania gamma - 0,99,
- veľkosť kroku učenia alfa - 0,003,
- veľkosť dávky - 900,

Neurónová sieť použitá ako aktor je tvorená 170 neurónmi na vstupnej vrstve, následne dvoma vrstvami s 256 neurónmi a 3 neurónmi na výstupnej vrstve. Neurónová sieť použitá ako kritik je tvorená 170 neurónmi na vstupnej vrstve, dvoma vrstvami s 256 neurónmi a jedným neurónom na výstupnej vrstve.

Pri rozmere neurónovej siete, ako aj pri nastavení vstupných parametrov sme vychádzali zo skúsenosti s riešením deterministickej úlohy priradenia personálu, pričom veľkosť dávky musí byť väčšia ako počet rozhodnutí v jednom simulačnom behu.

Nasadenie algoritmu učenia posilňovaním

Obdobne, ako pri úlohe priradenia personálu bol algoritmus učenia posilňovaním spolu so simulačným jadrom nasadený na štandardnom počítači, pričom na tomto počítači prebiehala aj prípravná fáza. Parametre zariadenia boli:

- RAM - 16 GB,
- Procesor - Intel(R) Core(TM) i5-9300HF CPU @ 2.40GHz 2.40 GHz.

Prípravná fáza (trénovanie)

Pre natrénovanie neurónovej siete bolo realizovaných 32 500 simulačných behov (replikácií), nakoľko sme predpokladali, že počet replikácií pri stochastickej úlohe musí byť väčší, ako pri deterministickej úlohe, pri ktorej bolo realizovaných 15 000 replikácií a pri určení presného počtu replikácií sme vychádzali z toho, v akej fáze tréningu je pozorovateľné ustálenie dosahovaných výsledkov.

Overenie funkčnosti vybraného algoritmu

V tejto fáze by mala byť dostatočne natrénovaná neurónová sieť, a malo by byť možné ju použiť v systéme na podporu rozhodovania. Toto sa však v tejto fáze nepodarilo, nakoľko algoritmus učenia posilňovaním nevykazoval známky natrénovania (dosahovanie lepších výsledkov, ako v prvotných fázach tréningu). Vzhľadom k tomu bolo potrebné použiť pôvodné nastaveniu jednotlivých parametrov.

Ako prvý krok sme rozšírili informácie o stave prostredia. Zahrnuli sme obsadenosť jednotlivých koľají, ako aj plánovaný príchod jednotlivých vlakov. Bohužiaľ, ani toto sa neukázalo ako postačujúce. Preto sme pristúpili k rozšíreniu veľkosti siete a to na sieť pre aktora s 202 neurónmi na vstupnej vrstve, potom tri vrstvy s 512 neurónmi a výstupná vrstva s 2 neurónmi, ktoré reprezentujú jednotlivé akcie a sieť pre kritika s 202 neurónmi na vstupnej vrstve, potom tri vrstvy s 512 neurónmi a výstupná vrstva s 1 neurónom. Ani pri takejto zmene však stále nedochádzalo k trénovaniu algoritmu. Následne sme teda rôzne menili parametre alfa, gamma a počet epoch. Výsledky jednotlivých experimentov neuvádzame v tejto práci vzhľadom k tomu, že nedochádzalo k žiadnemu zlepšeniu a teda výsledky boli podobné, ako by sme použili netrénovanú sieť.

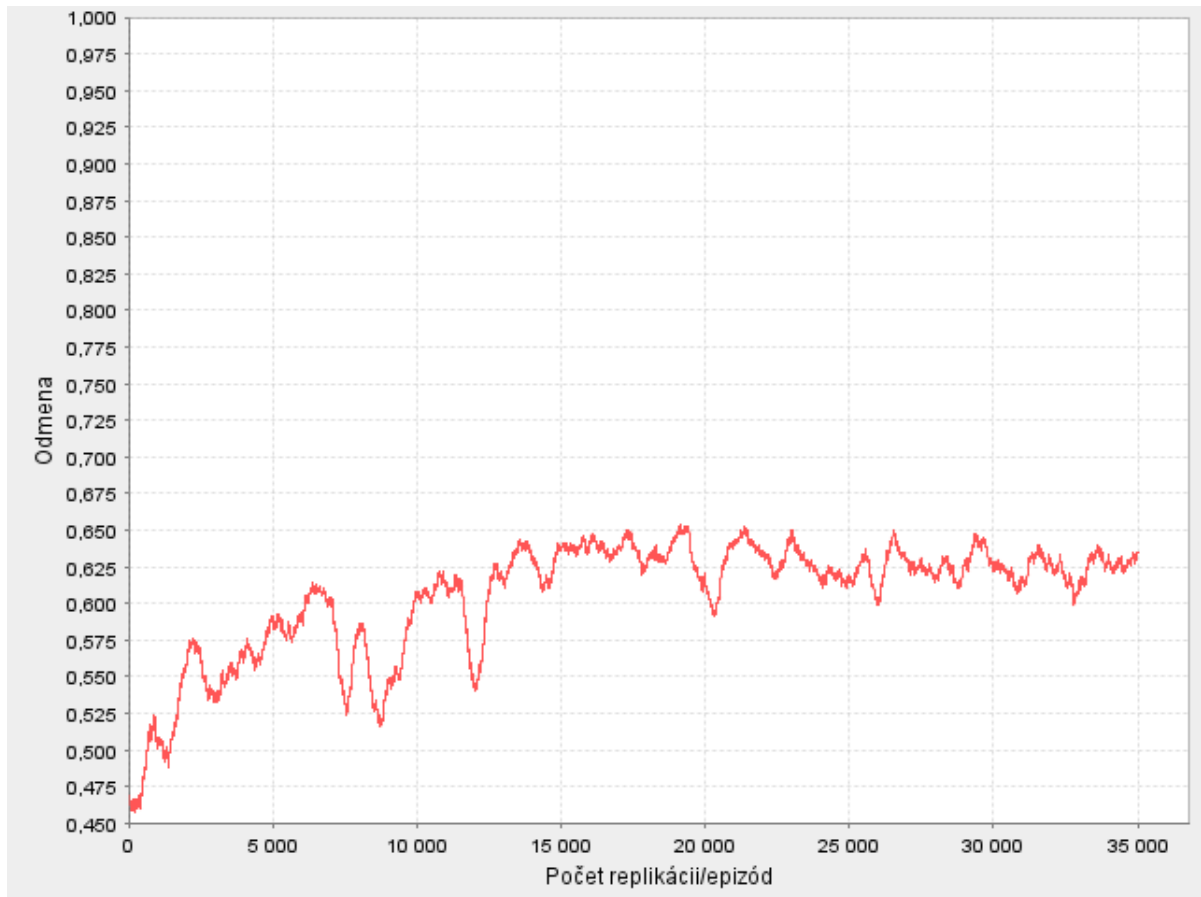
Vďaka týmto experimentom sa však podarilo dosiahnuť také nastavenia, že algoritmus učenia posilňovaním dosahuje lepšie výsledky ako pevne stanovené poradie úloh. Jednotlivé experimenty boli časovo náročné, a teda sme sa pri jednotlivých zmenách nevracali k východiskovému nastaveniu, ale snažili sme sa k zmenám jednotlivých parametrov pristupovať s vedomím, že jednotlivé parametre ovplyvňujú dosahovaný výsledok spoločne a nie izolovane. Vzhľadom k tomu, je problematické určiť, ktorá zmena priniesla zlepšenie.

Parametre, ktorých použitie viedlo k dosiahnutiu požadovaných výsledkov sú:

- počet epoch - 12,

- faktor zrážania γ - 0,998,
- veľkosť kroku učenia α - 0,001,
- veľkosť dávky - 900.

Nasledujúci graf zobrazuje dosiahnutý vývoj tréningu algoritmu PPO. Dosahované výsledky začínajú byť stabilnejšie od 15 000 replikácií. Bolo vykonaných viac ako 30 000 replikácií, pričom väčšiu stabilitu tréningu sa nepodarilo dosiahnuť.



Obr. 11.12: Tréning PPO agenta v stochastickom prostredí

Takto natrénovaný algoritmus je možné nasadiť ako sofistikovaný algoritmus v systéme na podporu rozhodovania. Nasledujúca tabuľka zobrazuje dosiahnuté výsledky pre 30 replikácií v porovnaní s pôvodne používaným algoritmom s pevne stanoveným počtom úloh, ako aj v porovnaní s netrénovaným algoritmom PPO (náhodný výber). Pre dosiahnuté celkové meškania bol určený 90 percentný interval spoľahlivosti. Tabuľka 11.3 zobrazuje dosiahnuté celkové meškania. V prípade nevypravených vlakov je zahrnuté meškanie do 6:00 ráno nasledujúceho dňa (dĺžka simulačného behu).

Na základe výsledkov môžeme povedať, že algoritmus PPO so správne nastavenými parametrami je vhodný na riešenie danej úlohy v stochastickom prostredí. Z vykonaných experimentov je zrejmé, že úloha

Tabuľka 11.3: Porovnanie výsledkov experimentov s použitím algoritmu PPO pri určení poradia úloh

	Pevne určené poradie (hod.)	Náhodný výber (hod.)	Natrénovaný algoritmus (hod.)
Celkové meškanie	<87,9;171,1>	<130,1;201,2>	<35,7;55,3>

je veľmi citlivá na nastavenie jednotlivých parametrov. V budúcnosti by sme sa chceli v tejto oblasti venovať ďalšiemu výskumu a to najmä použitiu grafových neurónových sietí [39]. Viacerí autori tento prístup využili pri realizácii výskumu v oblasti železničnej dopravy [73] [26] a preto predpokladáme, že by mohol byť vhodný aj na riešenie typov úloh, akými sa zaoberáme v tejto dizertačnej práci.

Záver

Železničný dopravný uzol je komplexným systémom, ktorý je súčasťou železničnej siete a ktorého plynulá prevádzka je nutná pre zabezpečenie plynulého chodu celej železničnej dopravy. Hoci je riadenie železničného dopravného uzla náročné na všetkých úrovniach, najmä operatívne riadenie je komplikované a vo veľkej miere závislé na schopnostiach, vedomostiach a skúsenostiach jednotlivých dispečerov. Dispečeri musia prinášať rozhodnutia, tak aby bola zabezpečená efektívna prevádzka celého železničného dopravného uzla.

Uvedomujúc si náročný rozhodovací proces dispečerov je potrebné zaoberať možnosťami poskytovania podpory rozhodovania a to najmä v reálnom čase. Dizertačná práca ponúka analýzu niektorých existujúcich informačných, a najmä dispozičných systémov v oblasti železničnej dopravy. Takýmito systémami sú napríklad Prevádzkový informačný systém, systémy spoločnosti Siemens (dôležitým je najmä systém VICOS), Graficko-technologická nadstavba a mnohé ďalšie. Keďže tieto systémy nie sú v súčasnosti dostatočné a hlavne sa zväčša zaoberajú čiastkovými problémami a nie železničným dopravným uzlom ako celkom, bol navrhnutý nový systém na podporu rozhodovania. Vzhľadom na skúsenosti s vytváraním simulačných modelov navrhujeme systém, ktorý plne využíva výhody, ktoré ponúkajú simulačné modely a to tak, že simulačný model stojí v jadre navrhovaného systému.

V dizertačnej práci bola zostavená metodika popisujúca postup tvorby systému na podporu rozhodovania v reálnom čase postaveného na simulačných modeloch železničných dopravných uzlov, ktorá bude nápomocná pri tvorbe a nasadení takéhoto systému. V rámci tejto metodiky sú detailne popísané jednotlivé kroky, ktoré musí tvorca systému nevyhnutne vykonať, ako aj návrhy riešení viacerých problémov, ktoré môžu pri tvorbe takéhoto systému vzniknúť. Realizácia takéhoto systému môže napomôcť dispečerom v ich náročnej práci, zefektívniť operatívne riadenie a tým celé fungovanie železničnej dopravy.

Za teoretický prínos dizertačnej práce považujeme návrh architektúry systému na podporu rozhodovania dispečera s využitím simulačných modelov v jadre tohto systému. Architektúra tohto systému je navrhnutá tak, aby sa jednotlivé časti dali implementovať nezávisle. Navrhovaná architektúra zahŕňa popis komunikačných protokolov a rozhrania umožňujúceho prenos informácií medzi jednotlivými časťami systému, popis jednotlivých komponentov. V práci sú definované informácie o reálnom systéme, ktoré sú potrebné pre fungovanie systému na podporu rozhodovania, ako aj možnosti zberu takýchto informácií. Ďalej je dôležitým teoretickým prínosom zostavenie metodiky, v rámci ktorej je popísaný postup, akým systém na podporu rozhodovania vytvoríť a nasadiť v železničnom uzle.

Za praktický prínos považujeme rozšírenie agentovo-orientovanej architektúry ABAsim o možnosť komunikácie s externými systémami (rôznymi mikroslužbami, modulmi aj komplexnými softvérovými aplikáciami) pomocou navrhnutého komunikačného agenta využívajúceho služby websocket komunikácie, nakoľko takéto rozšírenie môže byť v budúcnosti využité rôznymi spôsobmi.

V súčasnosti je značnou komplikáciou závislosť riadenia na ľudskom faktore - skúsenostiach dispečera. Navrhovaný systém na podporu rozhodovania umožňuje čiastočne nahradiť ľudský faktor pri tvorbe návrhov rozhodnutí, tým že umožňuje využívať sofistikované algoritmy (exaktný prístup, heuristiky, učenie posilňovaním a iné). V práci boli vybrané metódy použité pre overenie navrhovaného postupu na vybraných úlohách pridelovania personálu a určovania poradia servisných úloh v reálnych železničných dopravných uzloch. Ďalším z praktických prínosov dizertačnej práce je teda aplikácia výsledkov takýchto metód, ktoré okrem využitia v systéme na podporu rozhodovania, môžu pomôcť zlepšiť samotné simulačné modely, pretože zvyšujú presnosť modelovania vzhľadom na reálne rozhodnutia v praxi. V rámci použitia sofistikovaných metód sme sa zaoberali najmä použitím algoritmov učenia posilňovaním. Nakoľko sa jedná o inovatívny prístup, ktorý zatiaľ nie je v širokej miere v rámci riadenia železničných dopravných uzlov používaný, považujeme za prínosné, že v rámci tejto práce bola vypracovaná metodika nasadenia takýchto algoritmov v rámci systémov na podporu rozhodovania. Veríme, že dizertačná práca bude prínosom v oblasti poskytovania podpory rozhodovania dispečerov v reálnom čase a môže prispieť k budúcemu zefektívneniu operatívneho riadenia v železničnom dopravnom uzle.

Zoznam použitej literatúry

- [1] N. Adamko. „Agentovo orientovaná simulácia zložitých obslužných systémov“. In: *Habilitačná práca* (2013), s. 114.
- [2] N. Adamko, V. Klima a P. Marton. „Designing railway terminals using simulation techniques“. In: *International Journal of Civil Engineering* 8.1 (2010), s. 57–67.
- [3] A. Azadeh, A. Negahban a M. Moghaddam. „A hybrid computer simulation-artificial neural network algorithm for optimisation of dispatching rule selection in stochastic job shop scheduling problems“. In: *International Journal of Production Research* 50.2 (2012), s. 551–566.
- [4] M. Bažant a A. Kavička. „Artificial neural network as a support of platform track assignment within simulation models reflecting passenger railway stations“. In: *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* 223.5 (2009), s. 505–515.
- [5] M. Bažant et al. „Simulation-based rail traffic optimisations applying multicriterial evaluations of variants“. In: *Mendel* 25.1 (2019), s. 139–146.
- [6] M. Botts, A. Robin a E. Hirschorn. *OGC SensorML: Model and XML Encoding Standard*. [cit.2020-08-10]. Dostupné na internete: <http://docs.opengeospatial.org/is/12-000r2/12-000r2.html>.
- [7] A. Bretas et al. „Modelling railway traffic management through multi-agent systems and reinforcement learning“. In: (2019), s. 291–297.
- [8] CAPACITY4RAIL. „Data notation and modeling“. In: *Public deliverable D 3.4.1* (2016).
- [9] G. Cecchetti et al. „Communication platform concept for virtual testing of novel applications for railway traffic management systems“. In: *Transportation Research Procedia*. Zv. 62. 2022, s. 832–839.
- [10] F. Corman et al. „A tabu search algorithm for rerouting trains during rail operations“. In: *Transportation Research Part B: Methodological* 44.1 (2010), s. 175–192.
- [11] J. Čamaj a J. Gašparík. *Informačné a komunikačné technológie v železničnej doprave*. prvé vydanie. Žilinská univerzita v Žiline: EDIS, 2011, s. 179. ISBN: 978-80-554-0307-6.
- [12] A. D’Ariano, D. Pacciarelli a M. Pranzo. „A branch and bound algorithm for scheduling trains in a railway network“. In: *European Journal of Operational Research* 183.2 (2007), s. 643–657.
- [13] F.L. Da Silva a A.H. Reali Costa. „A survey on transfer learning for multiagent reinforcement learning systems“. In: *Journal of Artificial Intelligence Research* 64 (2019), s. 645–703.
- [14] R. Diviš a A. Kavička. „The method of nested simulations supporting decision-making process within a mesoscopic railway simulator“. In: (2016), s. 100–106.

- [15] N. Dragoni et al. *Microservices: How to make your application scale*. Zv. 10742 LNCS. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2018, s. 95–104.
- [16] D. Drungilas et al. „Deep reinforcement learning based optimization of automated guided vehicle time and energy consumption in a container terminal“. In: zv. 67. 2023, s. 397–407.
- [17] S. Dünder a T. Şahin. „Train re-scheduling with genetic algorithms and artificial neural networks for single-track railways“. In: *Transportation Research Part C: Emerging Technologies* 27 (2013), s. 1–15.
- [18] R. T. Fielding. *REST: Architectural Styles and the Design of Network-based Software Architectures*. [cit.2022-05-06]. 2000. Dostupné na internete: <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.
- [19] M. Fischetti a M. Monaci. „Using a general-purpose Mixed-Integer Linear Programming solver for the practical solution of real-time train rescheduling“. In: *European Journal of Operational Research* 263.1 (2017), s. 258–264.
- [20] J. Foster et al. „Chapter 8 - Using XML-based RPC“. In: *Developing Web Services with Java APIs for XML (JAX Pack)*. [cit.2023-01-03]. Burlington: Syngress, 2002, s. 303–332. ISBN: 978-1-928994-85-5. Dostupné na internete: <https://www.sciencedirect.com/science/article/pii/B9781928994855500275>.
- [21] J. Gašparík a J. Kolář. *Železniční doprava - technologie, řízení, grafikony a dalších 100 zajímavostí*. 1. vydanie. U Pruhonu 22, Praha 7: Grada Publishing, a.s., 2017, s. 432. ISBN: 978-80-271-0058-3.
- [22] J. Gašparík, D. Lichner a P. Blaho. „Železničná dopravná prevádzka - základy dopravnej prevádzky“. In: *ISBN 978-80-554-0996-2* 1 (2015), s. 407.
- [23] Oltis Group. *IT systémy pro dopravu, spedici a logistiku*. [cit.2021-08-02]. Dostupné na internete: <https://www.oltis.cz/>.
- [24] Gurobi. *Gurobi Optimizer*. [cit.2022-05-06]. Dostupné na internete: www.gurobi.com.
- [25] P. Hanczar a A. Zandi. „A novel model and solution algorithm to improve crew scheduling in railway transportation: A real world case study“. In: *Computers and Industrial Engineering* 154 (2021).
- [26] J. S. W. Heglund et al. „Railway Delay Prediction with Spatial-Temporal Graph Convolutional Networks“. In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. 2020, s. 1–6.
- [27] Y. Hirashima. „A reinforcement learning method for train marshaling based on movements of locomotive“. In: *IAENG International Journal of Computer Science* 38.3 (2011), s. 242–248.

- [28] Y. Hirashima. „A reinforcement learning with group-based candidate-extraction for container marshalling at marine ports“. In: *Lecture Notes in Engineering and Computer Science*. Zv. 2233. 2018, s. 11–16.
- [29] J. Högdahl, M. Bohlin a O. Fröidh. „A combined simulation-optimization approach for minimizing travel time and delays in railway timetables“. In: *Transportation Research Part B: Methodological* 126 (2019), s. 192–212.
- [30] T. Chu et al. „Multi-agent deep reinforcement learning for large-scale traffic signal control“. In: *IEEE Transactions on Intelligent Transportation Systems* 21.3 (2020), s. 1086–1095.
- [31] P. Jamshidi et al. „Microservices: The journey so far and challenges ahead“. In: *IEEE Software* 35.3 (2018), s. 24–35.
- [32] J. Janáček et al. „Optimization of periodic crew schedules with application of column generation method“. In: *Transportation Research Part C: Emerging Technologies* 83 (2017), s. 165–178.
- [33] L. Jánošíková, A. Kavička a M. Bažant. „Optimal operation scheduling and platform track assignment in a passenger railway station“. In: *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* 228.3 (2014), s. 271–284.
- [34] E. Jánošíková a M. Krempl. „Train platforming problem“. In: *Lecture Notes in Geoinformation and Cartography* (2015), s. 149.
- [35] D. Jorgensen. *Chapter 4 - Information Exchange Using the Simple Object Access Protocol (SOAP)*. 2002. Dostupné na internete: <https://www.sciencedirect.com/science%20/article/pii/B9781928994817500071>.
- [36] A. Kavička, V. Klima a N. Adamko. „Agentovo orientovaná simulácia dopravných uzlov“. In: *EDIS - Vydavateľstvo Žilinskej univerzity* (2005).
- [37] H. Khadilkar. „A Scalable Reinforcement Learning Algorithm for Scheduling Railway Lines“. In: *IEEE Transactions on Intelligent Transportation Systems* 20.2 (2019), s. 727–736.
- [38] E. Khmeleva et al. „Fuzzy-Logic Controlled Genetic Algorithm for the Rail-Freight Crew-Scheduling Problem“. In: *KI - Kunstliche Intelligenz* 32.1 (2018), s. 61–75.
- [39] T. N. Kipf a M. Welling. „Semi-supervised classification with graph convolutional networks.“ In: *International Conference on Learning Representations (ICLR)* (2017).
- [40] M. Kováč. „Využitie algoritmov učenia posilňovaním pri riešení dopravných úloh“. In: *Bakalárska práca* (2024).
- [41] E. Kršák, H. Bachratý a V. Polach. „GTN - Information system supporting the dispatcher and remote tracks control“. In: *Communications - Scientific Letters of the University of Žilina* 12.3 A (2010), s. 65–74.

- [42] P. Kučera a P. Drdla. „Geografická metoda pro Automatické stavění vlakových cest“. In: *Vědecko-technický sborník Správy železnic* 3/2020 (2020), s. 3–18.
- [43] R. Licciardello et al. „Integrating yards, network and optimisation models towards real-time rail freight yard operations“. In: *Ingegneria Ferroviaria* 75.6 (2020), s. 417–447.
- [44] X. Luan et al. „Decomposition and distributed optimization of real-time traffic management for large-scale railway networks“. In: *Transportation Research Part B: Methodological* 141 (2020), s. 72–97.
- [45] P. Lubbers a F. Greco. *HTML5 WebSocket: A Quantum Leap in Scalability for the Web*. [cit.2023-03-03]. 2023. Dostupné na internete: www.websocket.org.
- [46] J. Majerčák et al. *ŽELEZNIČNÁ DOPRAVNÁ PREVÁDZKA. Technológia železničných staníc*. 1. vydanie. Žilinská univerzita v Žiline: EDIS, 2008, s. 230. ISBN: 978-80-8070-887-0.
- [47] Berghof Process Management. *Dispo Management system for rail vehicles in the depot area*. [cit.2021-01-05]. 2021. Dostupné na internete: <https://www.berghof-processmanagement.com/en/railway/dispo/>.
- [48] P. Martón. *VLAHOTVORNÉ STANICE prístupy k zvyšovaniu efektívnosti prevádzky*. 1. vydanie. Žilinská univerzita v Žiline: EDIS, 2016, s. 137. ISBN: 978-80-554-1246-7.
- [49] W. Mei a Z. Long. „Research and Defense of Cross-Site WebSocket Hijacking Vulnerability“. In: *Proceedings of 2020 IEEE International Conference on Artificial Intelligence and Computer Applications, ICAICA 2020*. 2020, s. 591–594.
- [50] J. Melich. „Náhrada informačného systému MIS 2 novým systémom vlakotvorná stanice“. In: *Vědecko-technický sborník ČD* 26 (2008).
- [51] Siemens AG Mobility Division. *Controlguide OCS / TMS*. [cit.2021-07-02]. Dostupné na internete: <https://www.mobility.siemens.com/global/en/portfolio/rail/automation/operations-control-systems/controlguide-ocs.html>.
- [52] Siemens AG Mobility Division. *Efficient rail transport with the Vicos OC operations control system family*. [cit.2021-08-02]. Dostupné na internete: <https://www.mobility.siemens.com/global/en.html>.
- [53] T. Molková. „Postavení místního informačního systému v systému řízení dopravního provozu na ČD“. In: *Scientific papers of the university of Pardubice series B The Jan Perner Transport Faculty* (1995), s. 93–99.
- [54] M. Morley. *JSON - RPC*. [cit.2022-05-06]. Dostupné na internete: <https://www.jsonrpc.org/>.
- [55] F. Naldini, P. Pellegrini a J. Rodriguez. „Real-Time Optimization of Energy Consumption in Railway Networks“. In: *Transportation Research Procedia*. Zv. 62. 2022, s. 35–42.

- [56] NeTEx. *NeTEx - Network and Timetabling Exchange*. [cit.2015-05-06]. Dostupné na internete: <http://netex-cen.eu>.
- [57] OpenTrack. *OpenTrack railway Technology*. [cit.2021-08-02]. Dostupné na internete: <https://www.opentrack.cz>.
- [58] P. Pellegrini, R. Pesenti a J. Rodriguez. „Efficient train re-routing and rescheduling: Valid inequalities and reformulation of RECIFE-MILP“. In: *Transportation Research Part B: Methodological* 120 (2019), s. 33–48.
- [59] V. Polach. „Plán vlakové dopravy v GTN zcela nově“. In: *Technologie* 4 (2021), s. 52–55.
- [60] A. Radtke a D. Hauptmann. *Automated planning of timetables in large railway networks using a microscopic data basis and railway simulation techniques*. Zv. 15. Advances in Transport. 2004, s. 615–625.
- [61] InterTech Rail. *Train Positioning System (TPS) using RFID System*. [cit.2023-01-03]. 2023. Dostupné na internete: <https://www.intertechrail.com/articles/train-positioning-system-tps-using-rfid-system>.
- [62] A. Robin. *OGC® SWE Common Data Model Encoding Standard*. [cit.2011-01-04]. Dostupné na internete: <http://www.opengis.net/doc/IS/SWE/2.0>.
- [63] I. Salvadori a F. Siqueira. „A Maturity Model for Semantic RESTful Web APIs“. In: *Proceedings - 2015 IEEE International Conference on Web Services, ICWS 2015*. 2015, s. 703–710.
- [64] A. N. Shabelnikov, I. A. Olgeizer a A. V. Sukhanov. „Machine learning in calculating speeds in a railway sorting yard“. In: *IOP Conference Series: Materials Science and Engineering*. Zv. 918. 2020. Kap. 1.
- [65] B. B. W. Schasfoort et al. „A dynamic model for real-time track assignment at railway yards“. In: *Journal of Rail Transport Planning and Management* 14 (2020).
- [66] Berkeley simulation software. *Real Traffic Controller*. [cit.2023-01-03]. 2023. Dostupné na internete: <https://berkeleysimulation.com/rtc.php>.
- [67] M. Šturma. „Automatické stavění vlakových cest poprvé na IV. tranzitním koridoru“. In: *Technologie* 3 (2021), s. 58–63.
- [68] M. Šturma. „Automatické stavění vlakových cest v CDP Přerov“. In: *Technologie* 1 (2021), s. 34–39.
- [69] P. Šulko, L. Pečený a J. Čamaj. „Traťová poloha vlaků - nová informační podpora pro výpravcov“. In: *Perner's contacts* 1.XIX (2019), s. 106–111.
- [70] D. Taibi et al. „Microservices in agile software development: A workshop-based study into issues, advantages, and disadvantages“. In: *ACM International Conference Proceeding Series*. Zv. Part F129907. 2017.

- [71] ZSSK Cargo (ISP tím). „Zákaznícky portál ISP - Používateľská príručka“. In: *ZSSK Cargo* Verzia 1.0 (2017).
- [72] J. Tkáč. „Predpis PREVÁDZKOVÝ INFORMAČNÝ SYSTÉM“. In: *Železnice Slovenskej republiky* (2018).
- [73] R. Wang et al. „The Method about the Prediction and Analysis of Causes for Railway Accidents based on Graph Convolutional Neural Network“. In: *2023 7th International Conference on Transportation Information and Safety (ICTIS)*. 2023, s. 496–501.
- [74] M. Woolfridge a N.R. Jennings. „Intelligent agents: theory and practice“. In: *The Knowledge Engineering Review* 10.2 (1995), s. 115–152.
- [75] W. Zhou et al. „Crew scheduling considering both crew duty time difference and cost on urban rail system“. In: *Promet - Traffic - Traffico* 28.5 (2016), s. 449–460.

Zoznam vlastných publikácií

- [1] A. Galadíková a N. Adamko. „Simulation-based methods to support the real-time management of railways nodes“. In: *Transportation Research Procedia* 55 (2021), s. 1345-1352
- [2] A. Galadíková. „Využitie algoritmov učenia posilňovaním pri optimalizácii železničných uzlov“. In: *Využitie kvantitatívnych metód vo vedeckovýskumnej činnosti a v praxi. Zborník zo seminára. - 1. vyd. - Bratislava: Vydavateľstvo EKONÓM (2021), ISBN 978-80-225-4831-1., s. 25-31*
- [3] A. Galadíková a N. Adamko. „Simulation-based optimization of personnel assignment in railway nodes“. In: *2022 IEEE 16th International Scientific Conference on Informatics, Informatics 2022 - Proceedings* (2022), s. 100–105
- [4] A. Galadíková. „Communication architecture for real-time decision support systems in railway nodes“. In: *TRANSCOM 2023: 15th International scientific conference of young scientists on sustainable, modern and safe transport. - 1 vyd. - Amsterdam: Elsevier, 2023. - s. 1203-1210*
- [5] A. Galadíková a N. Adamko. „Usage of proximal policy optimization algorithm for personnel assignment in railway nodes“. In: *Proceedings of The International Conference on Information and Digital Technologies (IDT), 2023. - 1 vyd. - Danvers: Institute of Electrical and Electronics Engineers, 2023. - ISBN 979-8-3503-0586-9. - s. 37-43*