Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Simulační nástroj železniční stanice založený na agentově orientovaném výpočetním jádru

Bc. Ladislav Ježek

Diplomová práce
2010

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2009/2010**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Ladislav JEŽEK**
Studijní program:	**N2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Simulační nástroj železniční stanice založený na agentově orientovaném výpočetním jádru**
Zadávající katedra:	**Katedra softwarových technologií**

Zásady pro vypracování:

1. Popis agentově orientovaného výpočetního jádra. 2. Vývoj simulačního nástroje založeného na agentově orientovaném výpočetním jádře se zaměřením na simulaci zjednodušeného provozu osobní železniční stanice. 3. Integrace rozhodovacích mechanismů pro problematiku rozhodování o přidělení náhradní nástupištní koleje pro zpožděné přijíždějící vlaky z různých vstupních směrů. 4. Vyhodnocení chování uvažovaného simulačního modelu pro různé vstupní parametry.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. KAVIČKA, Antonín, KLIMA, Valent, ADAMKO, Norbert. Agentovo orientovaná simulácia dopravných uzlov. Žilina: EDIS - vydavateľstvo ŽU, 2005. 206 s. Monografie. ISBN 80-8070-477-5. 2. ADAMKO, Norbert, KAVIČKA, Antonín, KLIMA, Valent. DAAAM International Scientific Book 2007. Branko Katalinic. Vienna : DAAAM International Publishing, 2007. ISBN 3-901509-60-7. Agent based simulation of transportation logistic systems, s. 407-422. 3. BAŽANT, Michael, KAVIČKA, Antonín. Artificial neural network as a support of platform track assignment within simulation models reflecting passenger railway stations. Journal of Rail and Rapid Transit. 2009, vol. 223, no. 5, s. 505-515. 4. VONKA, J., MOLKOVÁ, T., ŠIROKÝ J. Technologie a řízení dopravy II - GVD. Pardubice : Univerzita Pardubice, 2000. ISBN 80-7194-286-3. 5. VONKA, Jaroslav, et al. Osobní doprava. 1. vyd. Pardubice : Univerzita Pardubice, 2001. 170 s. ISBN 80-7194-320-7. 6. MAŘÍK M., ŠTĚPÁNKOVÁ O., LAŽANSKÝ J. a kol. Umělá inteligence (4). Praha : Academia, 2003, 475 str. ISBN 80-200-1044-0.

Vedoucí diplomové práce: **Ing. Michael Bažant, Ph.D.**
Katedra softwarových technologií

Datum zadání diplomové práce: **30. října 2009**

Termín odevzdání diplomové práce: **21. května 2010**

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

doc. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 10. listopadu 2009

Prohlašuji:
Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 20. 08. 2010

Bc. Ladislav Ježek

ANOTACE

Diplomová práce se zabývá vytvořením simulačního nástroje železniční stanice. Simulační nástroj je založen na agentově orientovaném diskrétním simulačním jádru. Aplikace má grafické zobrazení a používá svoje animační jádro, které je závislé na simulačním jádru. Řeší otázky ohledně problematiky animačního jádra.

KLÍČOVÁ SLOVA

simulace, animační jádro, železniční stanice, zobrazení infrastruktury

TITLE

Simulation tool of railway station based on agent oriented simulation engine

ANNOTATION

Theses is focused on development of tool to simulate tranportation processes within passenger railway station. Simulation tool is based on discrete agent oriented simulation engine. Application has graphic interface which uses its animate core that depends on simulation engine. Main purpose of this theses is to solve tasks regarding to animation core issues.

KEYWORDS

simulation, animation engine, railway station, Infrastructure view

Obsah

Seznam obrázků a tabulek

Seznam zkratek

C# – Objektové orientovaný programovací jazyk.
EC – EuroCity
Ex – Expres
IC – InterCity
Mn – Manipulační vlak
Os – Osobní vlak
Pn – Průběžný nákladní vlak
R – Rychlík
SC – SuperCity
Sp – Spěšný vlak
MDI – Multiple Document Interface (otevřeno více oken v rámcovém okně)
dll – Dynamic-link library (dynamicky linkovaná knihovna)
xml – Extensible Markup Language (rozšiřitelný značkovací jazyk)
žst – Železniční stanice

1 Úvod

V dnešní době se hojně používají simulace a simulátory ve všech odborných odvětvích. Pomocí simulace můžeme simulovat chování nějakého určitého sytému nebo prostředí. S využitím simulace se setkáme v letectví, v dopravě, v lékařství atd. Na základě získaných poznatků ze simulace se může člověk rozhodovat, zjišťovat jak daný systém funguje, provádět optimalizace simulovaných procesů. Stav beztíže lze například simulovat ponořením jedince nebo tělesa pod vodu, ale v současné době se spíše využívá počítačové simulace. Počítačová simulace je pokus o simulaci reálného sytému za pomoci počítače.

Diplomová práce se zabývá vytvořením simulačního nástroje, který simuluje pohyb vlaků v železniční stanici. Simulátor je založen na agentově orientovaném simulačním jádru, které se připojuje k aplikaci. Simulace je zaměřená na zjednodušený provoz vlaků v železniční stanici, konkrétně Hlavního nádraží v Praze. Uvažují se hlavně příjezdy a odjezdy vlaků, neřeší se malé posuny po nádraží nebo přemísťování lokomotivy.

Řeší se komunikace animačního jádra se simulačním jádrem, způsob animování události a práce s přiděleným simulačním časem pro animaci. Práce se zabývá počítáním délky kroku vlakové soupravy pro animaci, aby stav animaci odpovídal stavu simulačního jádra.

2 Základní pojmy

Projdeme si některé základní pojmy, které se týkají diplomové práce. Zaměříme se hlavně na oblast kolem simulace. Všechny informace v této kapitole pochází z publikace [1].

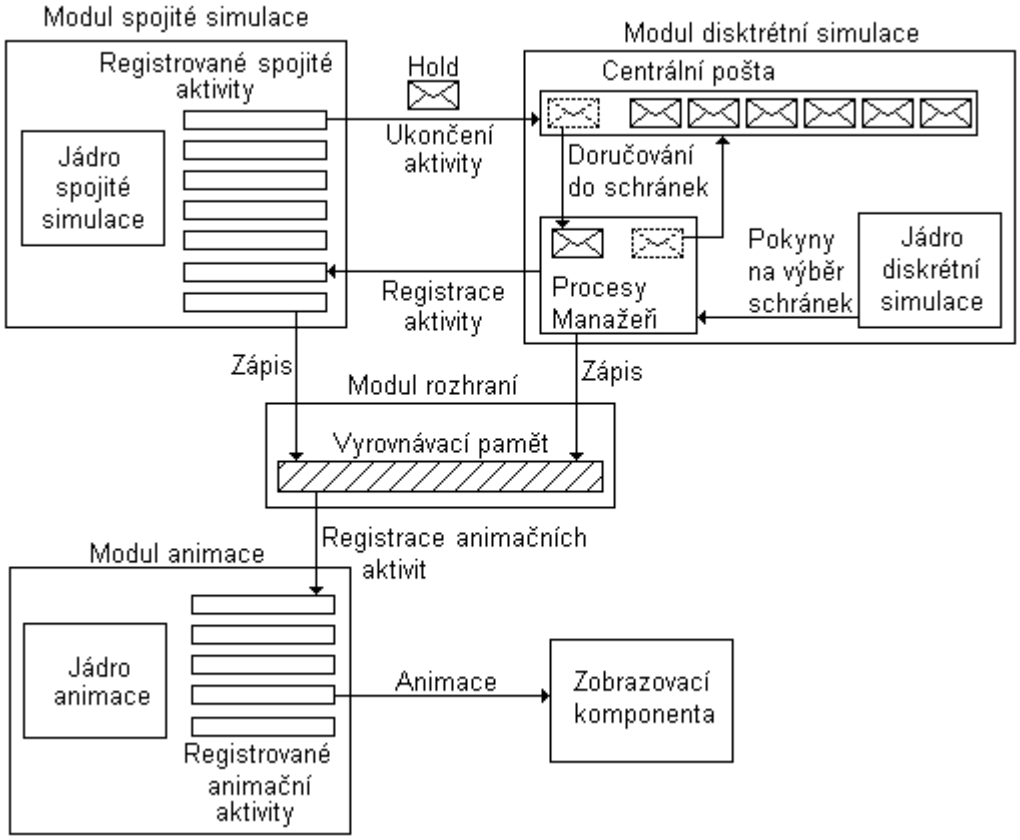
2.1 Simulace

Pomocí simulací zkoumáme objekty, které v reálném životě již existují - železniční stanice, výrobní linka, dopravní situace na křižovatkách, nebo mohou existovat v budoucnu - rozšíření železniční stanice, postavení obchvatu kolem města atd. Simulace napomáhá při realizaci rozsáhlejších projektů, které jsou finančně náročné, tím, že můžeme navrhované řešení otestovat a analyzovat, zda je navrhovaný projekt vhodným řešením. Dále lze pozorovat už existující projekt a tím zjišťovat informace, jak by se daný systém choval při různých situacích, které by mohly nastat.

Počítačovou simulaci rozdělujeme na spojité a diskrétní. Hlavním rozdílem je, že spojitá simulace používá spojité aktivity a diskrétní simulace diskrétní aktivity. Aktivita představuje základní simulační jednotku, která představuje určitou činnost v simulaci, ukrývá se pod ní pohyb určitého objektu, výrobní část celého procesu apod. Každá aktivita trvá danou dobu simulačního času a během změny simulačního času se mění stav simulace. U spojitých aktivit může nastat změna stavu kdykoliv při vykonávání aktivity, u diskrétní aktivity je změna stavu až při skončení činnosti, kde je vyvolaná změna stavu systému, která se nazývá událostí. Vykonáváním jednotlivých aktivit v určitém pořadí vytvářejí průběh simulačního programu.

Na následujícím obrázku (obr. 2.1) je vidět struktura kombinovaného simulačního modelu s animací. Hlavně je zobrazen komunikační tok mezi jednotlivými moduly a jejich hlavní činnost. Modul diskrétní simulace je nejdůležitější, protože řídí běh simulace a přiděluje časová kvanta ostatním modulům. Podrobněji se této

problematice věnuje kapitola 4 Synchronizace simulačního výpočtu, kde je popsána v jednotlivých krocích činnost každého modulu.



Obr. 2.1 – Struktura kombinovaného simulačního modelu s animací. Zdroj: [1]

2.2 Simulační modul

Simulační modul obsahuje simulační jádro. U kombinované simulace můžeme modul rozdělit na dva další moduly. Jeden modul by byl diskrétní simulace a druhý spojitá simulace. Modul diskrétní simulace kromě jádra ještě většinou obsahuje centrální poštu, která většinou představuje prioritní frontu, kde prioritou je čas. Modul spojité simulace má hlavně na starost uchovávat registrované spojité aktivity. Registrace aktivit je inicializována z diskrétního modulu, při zpracovaní události.

2.3 Animační modul

Animační modul se skládá z animačního jádra a registrovaných animačních aktivit. Hlavní činností modulu je registrování animačních aktivit, které řídí jádro modulu. Na základě těchto aktivit animační jádro vykonává procedury, které vytváří animační zobrazení objektů v simulaci. Zobrazuje většinou pohyb po určité trajektorii a tím se zobrazuje průběh simulace.

Aktivace animačního modulu je od jádra diskrétní simulace, které předává délku doby činnosti animačního jádra. Animační jádro nejdříve vybere všechny zprávy z vyrovnávací paměti, které se nachází v modulu rozhraní. Zprávy obsahují údaje o názvu animační aktivity, která se má vykonávat a důležitou informaci o době trvaní animace, dále může obsahovat i jiné informace pro animační jádro. Animační jádro na základě zpráv zaregistruje animační aktivity a po registraci začne vykonávat animace po dobu, kterou má zadanou od diskrétního jádra. Pokud některá aktivita skončí během této doby, tak je zrušená její registrace v animačním modulu. Po vyčerpání času pro animační modul předá animační jádro řízení diskrétnímu modulu.

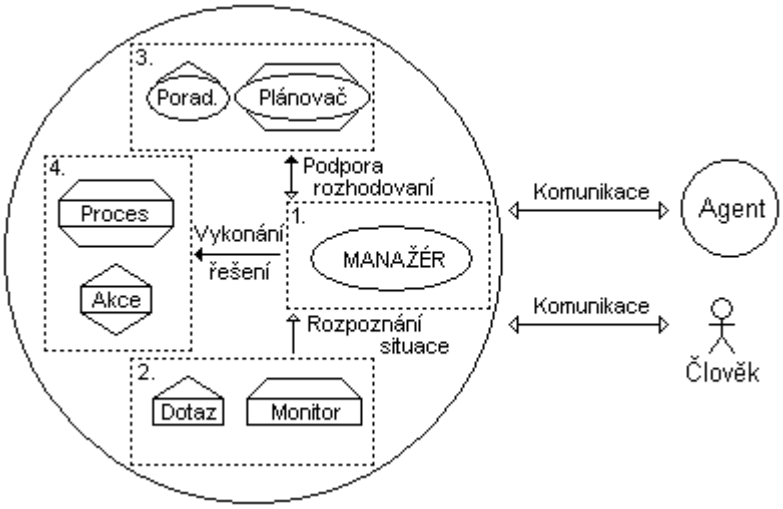
2.4 Modul rozhraní

Modul slouží na uložení zpráv pro animační modul. Zprávy do modulu vkládá simulační modul. Modul rozhraní si můžeme představit jako rozhraní mezi simulací a animačními výstupy na obrazovku.

2.5 Agent

Agenta si můžeme představit jako zapouzdřený systém, který dokáže existovat sám o sobě a reaguje na podněty nebo zprávy z okolního prostředí. Většinou má každý agent určité poslání a pokyny, jak reagovat na okolí. V dnešní době se hojně využívá agentů, například agenti hledající práci na internetu, kterým lze zadat požadavky, ale raději se zaměříme na agenty, kteří se používají v simulaci.

Aby agent fungoval, musí disponovat určitými schopnostmi. Jednou důležitou schopnosti je komunikace, zvládne komunikovat s ostatními agenty nebo s člověkem. Dále je agent v sytému identifikovaný, má vlastnosti, chová se podle definovaných pravidel, zvládne se rozhodovat a je soběstačný. Agent má definované cíle, které se snaží dosáhnout svými schopnostmi a způsobem chování.



Obr. 2.2 – Rozložení agenta. Zdroj: [1]

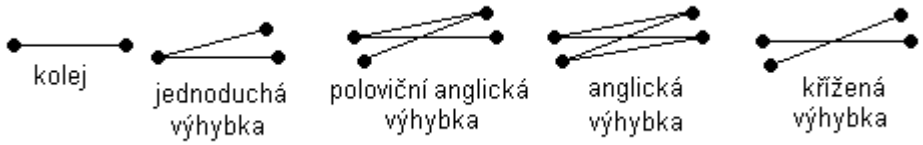
Agent se většinou skládá ze čtyř skupin interních komponent (obr. 2.2). Hlavní komponentou je manažer, který patří do skupiny řídicích a rozhodovacích komponent, ten má na starost rozhodování a komunikaci s okolím nebo s interními komponentami, které mezi sebou nekomunikují. Manažer pro svoji funkčnost využívá tedy interní komponenty, které se rozdělují do zbylých třech skupin. Jednou skupinou jsou senzory, do tohoto oddílu spadají komponenta dotaz a monitor. Dotaz vrací informaci hned na pokyn manažera, monitor monitoruje dlouhodoběji stav simulace. Komponenty poradce a plánovač patří do skupiny řešitelů. Manažerovi hlavně napomáhají při rozhodování. Poradce většinou hned vrací řešení daného problém, může to být optimalizační algoritmus nebo člověk. Plánovač funguje opačně, ten plánuje dopředu řešení problémů, které by mohly nastat a pokud najde závažný problém, informuje o tom manažera. Poslední skupinou jsou efektory, tam patří komponenta akce a proces. Akce zařídí okamžitou změnu stavu simulace, proces mění stav v časových okamžicích, třeba vykonává pohyb objektu.

3 Modelování železničních objektů

V této kapitole se budu věnovat grafickému vymodelování jednotlivých objektů, jak statických, tak dynamických. Při modelování kolejiště jsem čerpal z publikací [7] a [8]. Z publikace [11] se hlavně vycházelo při modelování vlakové soupravy.

3.1 Modelování kolejiště

U modelování kolejiště musíme zvolit vhodnou strukturu pro uchování informací. Nejvhodnější se jeví použití grafu, tato struktura se hodně často používá pro uchování dopravní sítě. Pro uchování železniční sítě můžeme využít z teorie grafů vrcholově ohodnocený orientovaný graf nebo hranově ohodnocený graf. Druhá volba lépe zachycuje skutečnou podobu kolejiště, kde kolej a výhybková kolej jsou představovány hranou grafu a ohodnocení hrany je délka koleje, potom vrcholy grafu jsou spoje mezi kolejemi. Jednoduchá výhybka je tvořena třemi vrcholy a křížená, poloviční anglická a anglická výhybka čtyřmi vrcholy (obr. 3.1). Tato struktura nám nabízí lehké ošetření zakázaného odbočení, stačí definovat pravidlo, které nedovoluje vlaku přejet z jedné výhybkové koleje na druhou.

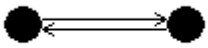


Obr. 3.1 – Ukázka zachycení kolejí u hranově ohodnoceného grafu

3.1.1 Modelování v simulačním modulu aplikace

Simulační modul, který je v diplomové práci využíván, používá hranově ohodnocený orientovaný graf. Všechny koleje se skládají ze dvou orientovaných hran a dvou vrcholů (obr. 3.2). Jednoduchá výhybka ze tří vrcholů a ostatní ze čtyř vrcholů. Diskrétní simulační jádro uchovává informace o koleji ve třídě Kolej<TIdentifikator, TDelka>, která obsahuje dva spoje Odkud a Kam, podle toho, jak byla kolej vytvořena. Pro uchování výhybky je použita

třída Vyhybka<TIdentifikator, TDelka>. Celá infrastruktura kolejiště je uchována ve třídě Infrastruktura<TIdentifikator, TDelka>.



Obr. 3.2 – Ukázka koleje s dvěma orientovanými hranami

3.1.2 Volba křivky

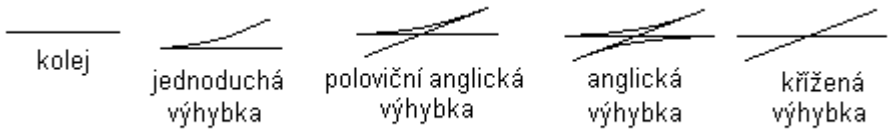
Existuje mnoho křivek, které se rozdělují na dvě základní skupiny, interpolační křivky a aproximační křivky [5]. Interpolační křivky prochází řídícími body, naopak aproximační křivky nemusí procházet řídícími body. Aproximační křivka se lépe hodí pro použití v diplomové práci, protože její řídící body méně ovlivňují tvar křivky než u interpolační křivky. Tím dochází k hladšímu průběhu křivky, tím nevznikne nežádoucí zvlnění křivky. Rozhodl jsem se pro Bézierovu křivku, která začíná v počátečním bodu a končí v koncovém bodu. Další důvodem bylo, že grafická třída Graphics v programovacím jazyku C#, už obsahuje metodu pro vykreslení Bézierovi křivky.

Jedná se o parametrickou křivku, která je pojmenovaná podle francouzského inženýra Pierru Bézierovi. Existují různé Bézierové křivky a to lineární, kvadratická a kubická. Lineární je určena jen dvěma body, proto křivka představuje úsečku. Kvadratická je už určená třemi body, kde druhý bod určuje ohyb křivky. V diplomové práci je využita kubická křivka, která je určena čtyřmi body a umožňuje křivku esovitě zkroutit. Křivka začíná v prvním bodě a končí ve čtvrtém bodě. Druhý a třetí bod určují tvar křivky a z pravidla jimi křivka neprochází. Druhý bod určuje směr, kterým křivka vychází z prvního bodu, a třetí bod určuje směr, kterým se křivka přibližuje k čtvrtému bodu.

3.1.3 Modelování v animačním modulu aplikace

Animační modul výše zmíněné třídy podědil a tím vznikly nové třídy KolejView, VyhybkaView a InfrastrukturaView. Třídy vznikly pro uchování informací, které jsou potřeba pro vykreslení objektu na obrazovku. Za TIdentifikator je zvolena třída PointF, která uchovává souřadnice bodu, respektive počátek a konec koleje

v souřadnicovém systému XY. Pro délku koleje je zvolen datový typ double, který nahrazuje TDelka. Délka koleje je určená vzdáleností počátku koleje od konce. Kdyby byly vykreslovány koleje jen rovně, tak by se celá infrastruktura moc nepodobala skutečné infrastruktuře na nádraží, proto se některé koleje vykreslují zakřiveně. Pro vykreslení křivky se používá známá Bézierova křivka, která je jedna z mnoha parametrických křivek. Pro vytvoření Bézierovy křivky jsou potřeba dva body, jako počátek a konec křivky, a dva body řídící, jejichž hodnoty jsou uchovány ve třídě KolejView. Tato křivka se hlavně využívá na některé koleje, které tvoří výhybku, nebo koleje ohnuté do určitého poloměru (obr. 3.3).



Obr. 3.3 – Ukázka vykreslených kolejí z aplikace

3.2 Modelování nástupiště

Nejjednodušší geometrický tvar, který se podobá nástupišti, je vyplněný obdélník v 2D prostoru. Pro vykreslení obdélníku je potřeba znát umístění levého horního bodu, délku a šířku nebo polohu tří bodů tvořící obdélník. Pokud je nástupiště více členité, je vhodnější ho vykreslit jako vyplněný polygon, u kterého potřebujeme posloupnost bodů.

3.2.1 Modelovaní v animačním modulu aplikace

Animační modul pro uchování struktury nástupiště používá svoji vlastní třídu NastupisteView, která si pamatuje posloupnost bodů. Nástupiště se tedy vykresluje jako vyplněný polygon, pro lepší zachycení nástupiště na železničním nádraží (obr. 3.4).

Obr. 3.4 – Ukázka vykresleného nástupiště v aplikaci

3.3 Modelování vlakové soupravy

Je celá řada možností, jak vymodelovat vlakovou soupravu, přičemž záleží na tom, co od vykresleného modelu požadujeme. Zaměříme se na pohled z ptačí perspektivy, který je pro simulaci v dopravě nejvhodnější. Za nejjednodušší způsob by se mohlo uvažovat vykreslení silnější čáry, která by kopírovala trajektorii koleje a byla barevně odlišná (obr. 3.5). Tato metoda je také nejrychlejší, co se týče vykreslování, ale nenabízí detail vlakové soupravy, který je většinou požadován. Výhodou u tohoto modelování je, že stačí si uchovávat jen počáteční a koncový bod vlaku a potom kopírovat vykreslenou kolej mezi body.

Obr. 3.5 – Vlak vymodelovaný pomocí jednoduché čáry

Pro skutečnější zobrazení, které odpovídá realitě, vyhovuje vykreslit orámovaný obdélník. Pod ním si už lze představit jednotlivé vagóny a lokomotivu lze barevně odlišit. S tímto stylem vykreslovaní je spojeno plno dalších výpočtů. Nestačí si uchovávat jen střed obdélníku, sice by bylo potřeba si jen pamatovat polohu tohoto bodu na koleji, ale zobrazení by neodpovídalo skutečnosti (obr. 3.6).

Obr. 3.6 – Vlak vymodelovaný pomocí obdélníku

Pro reálnější zobrazení je potřeba obdélník natočit, abychom ho mohli otočit, je potřeba znát velikost úhlu, o který se má obdélník otočit. Úhel lze spočítat z počátečního a koncového bodu koleje, na kterém se střed obdélníku nachází. Tím se přiblížíme k věrohodnějšímu zobrazení celého vlaku, ale nastává problém při ohnuté koleji, nebo u výhybek, kde se mění úhel vykreslené koleje. Vlak se potom jeví jako rozpojený a při animaci dochází rychlému směru jízdy vagónu, který nevypadá moc dobře (obr. 3.7).

Obr. 3.7 – Vymodelovaný vlak s otáčením obdélníku podle směru koleje

3.3.1 Modelovaní v animačním modulu aplikace

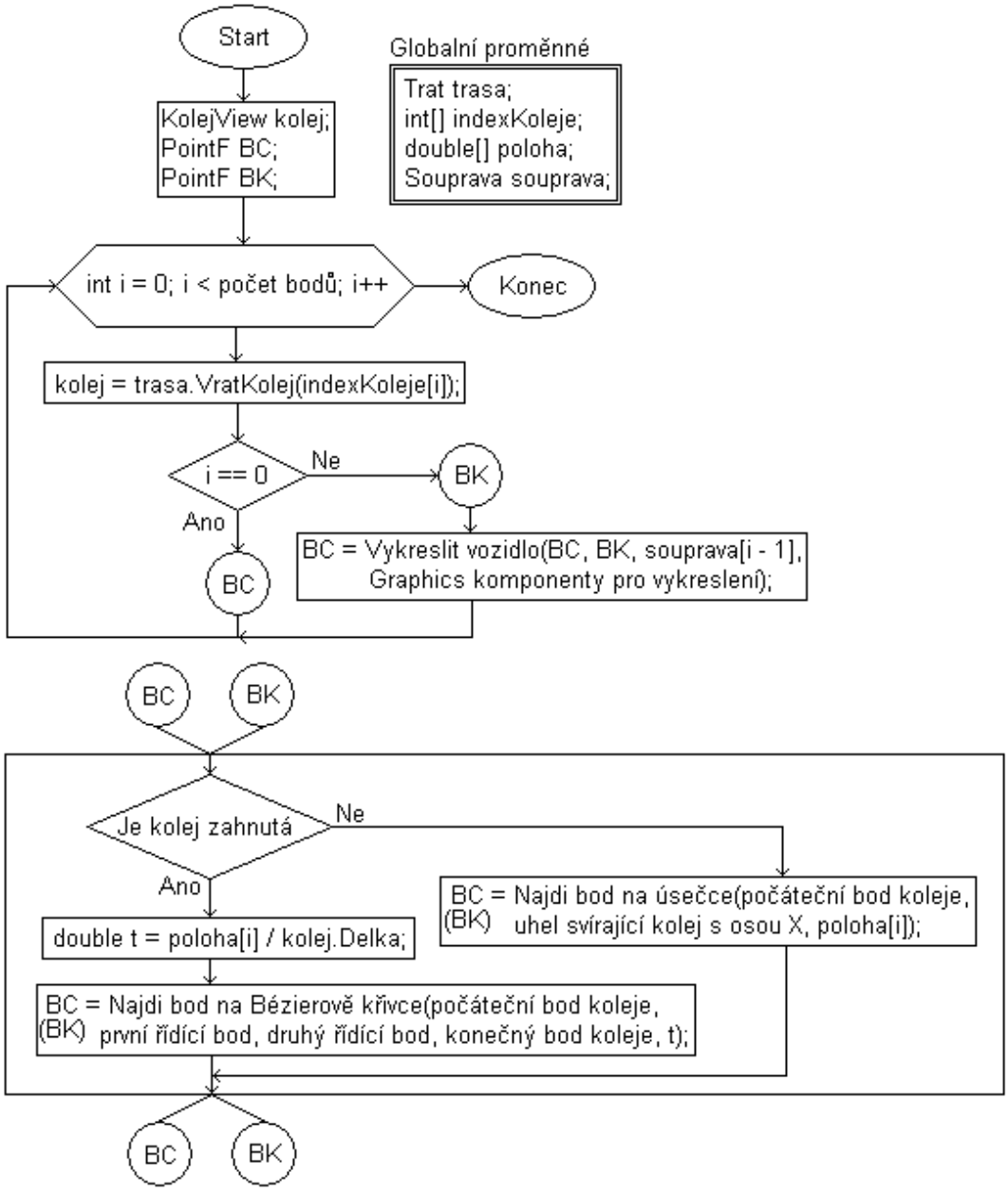
Jak je vidět na výše zobrazeném obrázku, tak vykreslování celého vlaku se podobá skutečnosti vyjma výhybek. Toto se dá zlepšit, tím že si u každého vagónu nebo lokomotivy pamatujeme polohu přední a zadní části. Tento způsob je použit i v diplomové práci. Všechny tyto informace o vlaku se uchovávají ve třídě Animace. Při uchování těchto bodů nám pomůže zjednodušení, které spočívá v tom, že kde končí vozidlo, tam začíná druhé vozidlo. To znamená, že pro lokomotivu a čtyři vagóny je zapotřebí šesti bodů, u kterých si budeme pamatovat polohu na dané koleji. Pro vykreslení vozidla je zapotřebí nejdříve spočítat úhel mezi body a osou X. Tím zjistíme směr lokomotivy či vagónu. Potom už stačí dané body natočit o daný úhel a vykreslit obdélník, podobně u ostatních vozidel. Tak docílíme vykreslení celého vlaku, který je v celém úseku spojený a neovlivní to ani ohnutá kolej nebo přejezd vlaku po výhybce. Na lepší zobrazení vlaku lze použít už připravené obrázky lokomotiv a vagónů při pohledu shora a tím vlak dostane reálnější podoby (obr. 3.8), než u barevného obdélníku s ohraničením.

Obr. 3.8 – Zobrazení vlaku z aplikace

3.3.2 Vývojový diagram vykreslení vlaku

Výše popsanou teorii lépe pochopíme na vývojovém diagramu (obr. 3.9). Proměnná trasa představuje seznam kolejí seřazené ve směru jízdy, kde se může indexově přistupovat k jednotlivým kolejím. Proměnná souprava je seznam, kde první je lokomotiva a zbytek vagóny. Je potřeba si pamatovat polohu výše zmíněných bo-

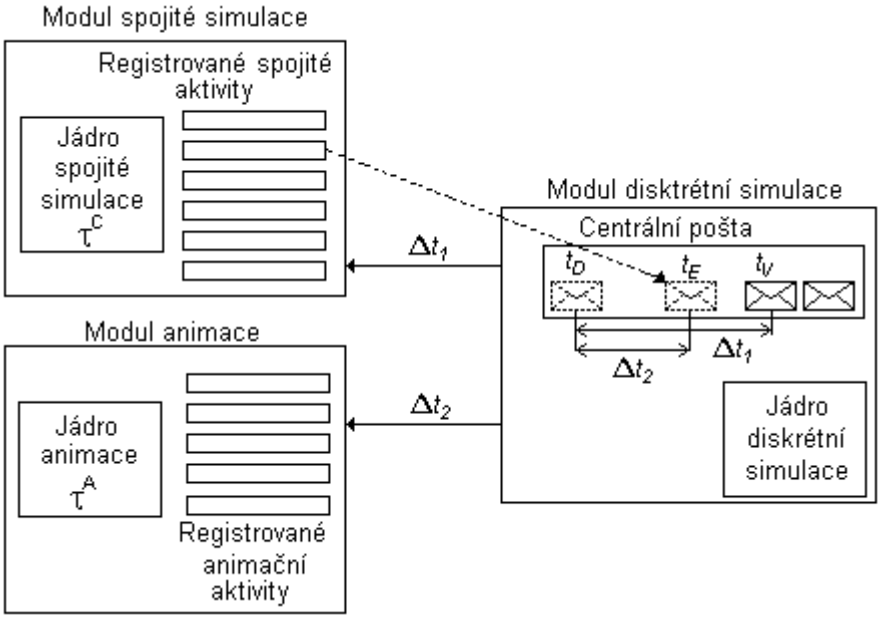
dů, k tomu nám slouží proměnná poloha, která přestavuje pole bodů, a proměnná indexKoleje, protože si pamatujeme polohu na jednotlivé koleji. Zkratka BC je bod čela lokomotivy nebo vagónu a BK je zas bod konce. Podmínka (i = 0) slouží že BC je potřeba spočítat jen napoprvé, protože vypočítáním BK vozidla získáme BC následujícího vozidla.



Obr. 3.9 – Vývojový diagram vykreslení vlaku

4 Synchronizace simulačního výpočtu

Kombinované simulační modely s animátorem vyžadují synchronizaci, protože jsou na něho kladeny dva požadavky. Prvním požadavkem je realizovat diskrétně-spojitou simulaci a druhým je vykonávat animaci některých aktivit, které chceme vidět. Strukturu takového modelu můžeme rozdělit na Modul diskrétní simulace, modul spojité simulace a modul animace. Mezi moduly simulace a modulem animace bývá rozhraní, které je také samo o sobě modulem viz obr. 2.1. V kapitole se vychází z publikace [1].



Obr. 4.1 – Přidělovaní časových kvant. Zdroj: [1]

Hlavním synchronizačním modulem je modul diskrétní simulace. Tento modul zahajuje činnost na začátku simulace a potom předává řízení ostatním modulům na definovanou část simulačního času (obr. 4.1). Modul, který dostal řízení, vrací zpět řízení diskrétnímu modulu po uplynutí času, který měl k dispozici nebo dřív. Celý koloběh synchronizace ukazuje níže uvedená tabulka.

Tab. 4.2 – Synchronizační koloběh kombinovaného simulačního modelu

Krok	**Řízení má**	**Činnost**
0	Diskrétní simulační modul	Inicializace simulačního času diskrétní simulace
1		Zjištění stavu centrální pošty, pokud není prázdná a ještě nevypršel čas pro běh simulačního programu, pokračuje se na krok 3
2		Ukončení simulace
3		Výběr zprávy s nejmenším časovým razítkem a následné zpracování zprávy
4		Zjištění časové kvanta mezi vybranou a následující zprávou, pokud se rovná nule, návrat na krok 1
5	Spojitý simulační modul	Vykonávaní spojitých aktivit, pokud nějaké existují. Svoji činnost provádí až do výskytu přerušení nebo vyčerpaní časového kvanta
6	Diskrétní simulační modul	Zjištění zbývajícího časového kvanta, pokud se rovná nule, návrat na krok 1
7	Animační modul	Provede registraci animačních aktivit, pokud jsou nějaké v modulu rozhraní
8		Provádí animační aktivity do vypršení časového kvanta, pokud jsou některé zaregistrované
9	Diskrétní simulační modul	Vydaní pokynu vyprázdnění modulu rozhraní
10		Aktualizace simulačního času diskrétní simulace na nejmenší časové razítko zprávy v poště, pokud není pošta prázdná
11		Návrat na krok 1

Zdroj: [1]

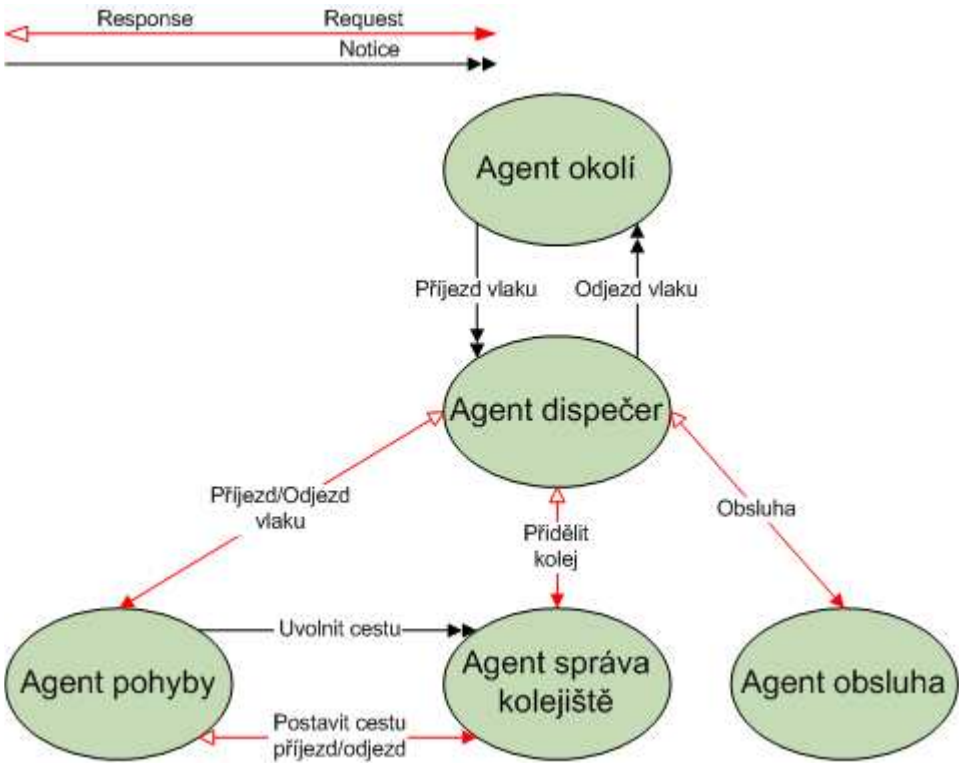
4.1 Synchronizace v aplikaci

Synchronizace v aplikaci je jednodušší, než je výše popsáno, protože neobsahuje modul spojité simulace. Simulační model obsahuje jen modul diskrétní simula-

ce, který je agentově orientovaný, dále modul rozhraní a modul animace. Všechny kroky jsou totožné, až na zrušené kroky pět a šest, jako v tabulce uvedené výše. Podrobnější činnost animačního modulu je popsána v další kapitole 6.2 Jeden krok animačního jádra.

5 Agentově orientované výpočetní jádro

Přesněji se jedná o diskrétní simulační jádro, které je postaveno na agentově orientované architektuře. Agenti mezi sebou komunikují pomocí zpráv, která obsahuje adresáta, příjemce a data. Zprávy agenti zasílají do centrální pošty, která je podle časového razítka zasílá k určeným příjemcům. Simulační jádro obsahuje několik agentů (obr. 5.1).



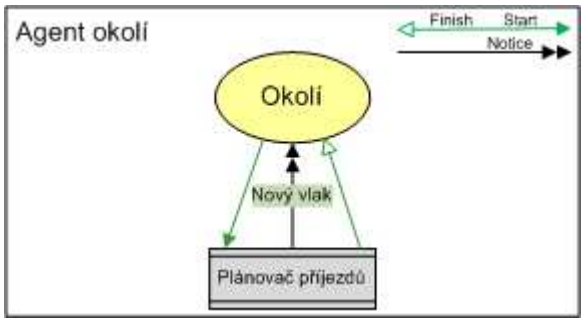
Obr. 5.1 – Hierarchie agentů v simulačním jádru. Zdroj: [6]

5.1 Popis jednotlivých agentů

Popíšeme si stručně jednotlivé agenty, které se vyskytuji v diskrétním simulačním jádru. Popíšeme si hlavně jeho činnost a komponenty, které obsahuje.

5.1.1 Agent okolí

Agent okolí je vstupní a výstupní brána do systému pro objekty, které představují vlaky přijíždějící do stanice nebo odjíždějící ze stanice. Hlavně využívá seznam vlaků, který tvoří prioritní fronta, kde prioritou je čas příjezdu. Hlavní činnost agenta je posílat ve správný čas vlaky do systému a potom informovat o opuštění vlaku ze systému.



Obr. 5.2 – Agent okolí. Zdroj: [6]

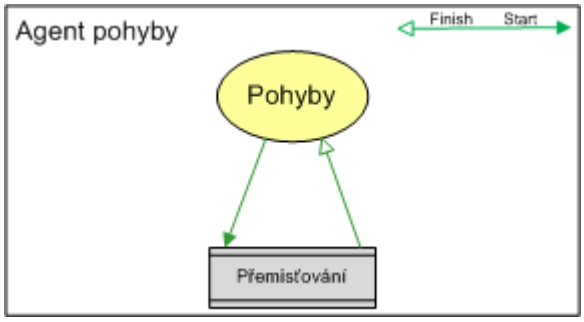
5.1.2 Agent správa kolejiště

Stará se o kompletní správu kolejiště. Agent rezervuje koleje pro daný vlak, pokud jsou koleje zarezervované, postaví cestu a tím dané koleje zablokuje, pokud je kolej součástí výhybky, tak zablokuje celou výhybku. K dispozici má aktuální přehled o volných kolejích u nástupišť. Samozřejmě když umí zablokovat koleje, tak i koleje uvolňuje. Informuje animační modul o rezervování, zablokování a uvolnění kolejí.

Obr. 5.3 – Agent správa kolejiště. Zdroj: [6]

5.1.3 Agent pohyby

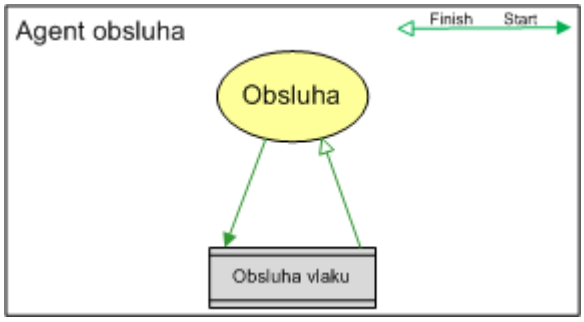
Tento agent obsluhuje pohyby vlaků v systému, pokud dostane zprávu o tom, že některý vlak má postavenou cestu, uvede tento vlak do pohybu. Předem určí, kdy vlak dojede na konec cesty a tím i ukončí pohyb vlaku ve stanici. Když uvede vlak do pohybu, informuje o tom animační modul, aby zaregistroval animaci. Informace obsahují kompletní údaje o vlaku, hlavní údaje jsou délka, číslo a druh vlaku. Dále zjistí pro vlak postavená cesta, a jak dlouho mu bude trvat danou vzdálenost překonat. Dále ještě informuje animační modul o zastavení vlaku na konci trasy, tato informace je jen testující, pro kontrolu, že stav animace odpovídá stavu simulačního jádra.



Obr. 5.4 – Agent pohyby. Zdroj: [6]

5.1.4 Agent obsluha

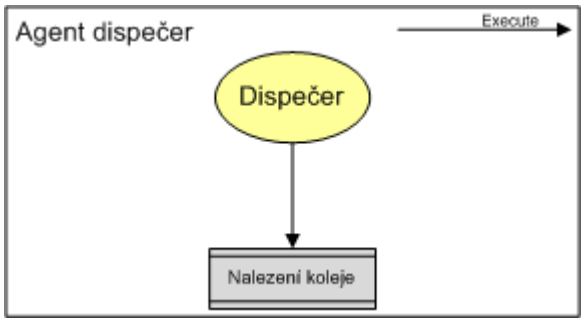
Když vlak zastaví u nástupiště, je o tom informován agent obsluhy. Tento agent hlavně zajišťuje spojení, rozpojení vlakové soupravy a hlavně přečíslovaní vlaku. Dále by mohl vykonávat činnost, která představuje kontrolu brzdných mechanismů vlaku, výstup a nástup cestujících apod.



Obr. 5.5 – Agent obsluha. Zdroj: [6]

5.1.5 Agent dispečer

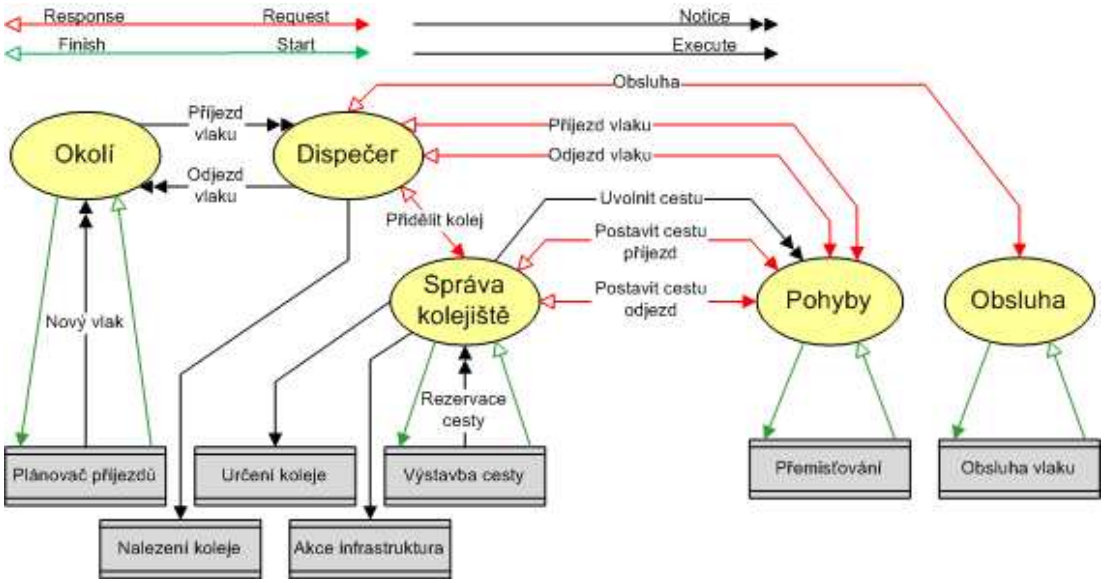
Každé větší nádraží má dispečink, kde se řídí chod celého nádraží. Pracovníci na dispečinku mají komplexní přehled o všech tratích, jak jsou postavené výhybky, o rozsvícených semaforech a vlacích v nádraží. Od dispečinku vycházejí požadavky pro ostatní systémy nebo pracovníky, kteří zajišťují dílčí činnosti pro chod železniční stanice. Takto si lze představit, jakou činnost vykonává agent dispečer. Agent je informován o novém příjezdu vlaku od agenta okolí a na základě této informace rozesílá požadavky dalším agentů v systému. Agent komunikuje se zbylými agenty. Komunikace mezi agenty je rozebrána v následující kapitole.



Obr. 5.6 – Agent dispečer. Zdroj: [6]

5.2 Komunikace mezi agenty ohledně jednoho vlaku

Agenti komunikují mezi sebou předáváním zpráv. Zprávy posílají do centrální pošty, která rozesílá zprávy určeným agentům. Každá zpráva obsahuje příjemce a odesílatele. V následující tabulce (tab. 5.8) je výpis komunikace mezi agenty od příjezdu vlaku až po zastavení u nástupiště a obsloužení vlaku. Jedná se jen o kladnou komunikaci, kdy probíhá komunikace bez překážek. Za překážky považuji například, že nelze postavit cestu nebo nástupištní kolej je obsazená atd.



Obr. 5.7 – Zjednodušený vrstvový MPE/ABAsim model železniční stanice.
Zdroj: [6]

Tab. 5.8 – Kladná komunikace mezi agenty ohledně příjezdu vlaku

Odesílatel	Příjemce	Činnost a zpráva
Agent Okolí	Agent Dispečer	Agent Okolí informuje agenta o příjezdu vlaku na vstupní kolej do systému.
Agent Dispečer	Agent Správa kolejiště	Agent ve zprávě posílá požadavek na přidělení koleje.
Agent Správa kolejiště	Agent Dispečer	Agent Správa kolejiště zjistí, jestli je kolej volná a pokud je, tak ji rezervuje. Informuje agenta Dispečera o přidělené koleji.
Agent Dispečer	Agent Pohyby	Informuje agenta o příjezdu koleje.
Agent Pohyby	Agent Správa kolejiště	Posílá zprávu s požadavkem na postavení cesty.
Agent Správa kolejiště	Agent Pohyby	Agent postaví cestu a informuje agenta Pohyby o postavené cestě.

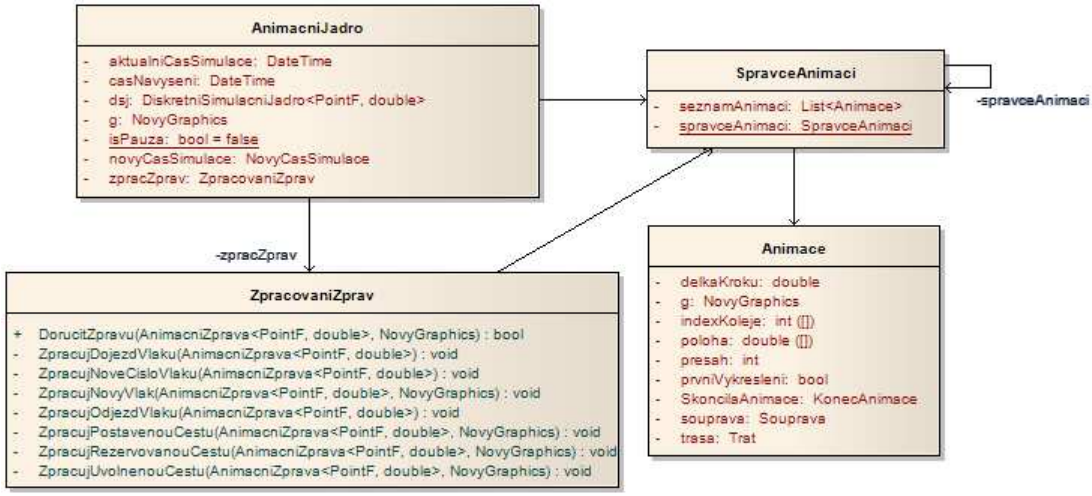
Agent Pohyby	Agent Správa kolejiště	Agent přemístí vlak k nástupišti a posílá zprávu s požadavkem o uvolnění cesty.
Agent Pohyby	Agent Dispečer	Informuje agenta Dispečera o příjezdu vlaku k nástupišti.
Agent Dispečer	Agent Obsluhy	Agent informuje agenta Obsluhy o vlaku stojícím u nástupiště.
Agent Obsluhy	Agent Dispečer	Agent po obsloužení vlaku (např. připojení, rozpojení nebo přečíslovaní vlaku) informuje agenta Dispečera o ukončení obsluhy.

Po obsloužení vlaku je komunikace mezi agenty velice podobná, zas je potřeba přidělit odjezdovou kolej a rezervovat ji. Potom postavit cestu a přemístit vlak. Po odjezdu vlaku ze systému zas uvolnit koleje.

6 Animační jádro

Jádro je celé naprogramováno pomocí programovacího jazyka C#, tento jazyk patří do vyšších programovacích jazyků, které jsou objektově orientované. Animační jádro v aplikaci představuje dynamickou knihovnu, pro prostředí Microsoft Windows (soubor s příponou .dll, který se k aplikaci připojuje, AnimacniJadroLib.dll). Jádro používá dynamickou knihovnu SimulacniJadro.dll, která představuje diskrétní jádro simulace a simulační jádro je také naprogramováno v C#.

6.1 Struktura jádra



Obr. 6.1 – Hlavní struktura animačního jádra

Hlavní struktura animačního jádra je zobrazena na obrázku nad odstavcem. Základní kamenem je třída AnimacniJadro, která obsahuje metodu KrokAnimacnihoJadra(TimeSpan casNaAnimaci), podrobněji se budeme metodě věnovat v další kapitole 6.2 Jeden krok animačního jádra.

6.1.1 Třída AnimacniJadro

Toto je nejdůležitější třída animačního jádra. Uchovává aktuální čas simulace pro zobrazení v simulátoru. Aktuální čas získává z diskrétního jádra, které je atributem třídy. Atribut novyCasSimulace je ukazatel na metodu, která v aplikaci zobrazuje čas simulace. Tento atribut se už předává jako parametr při inicializaci objektu. Funkce pro zobrazení času musí obsahovat parametr typu DateTimu, který patří do systémového balíčku programovacího jazyka C# a slouží pro uchování časového údaje. Druhým parametrem je ukazatel na funkci, která vrací Graphics komponenty na niž se mají vykreslovat animační prvky. Třída obsahuje hlavní metody pro pozastavení a zpuštění animačního jádra, tento stav se uchovává v atributu isPauza. Atribut nabývá dvou hodnot a to logickou jedničkou, pokud je animační jádro pozastaveno, nebo logickou nulou při opaku. Dále používá instanci třídy ZpracovaniZprav, kde využívá metodu DorucitZpravu. V atributu metody se předává animační zpráva.

6.1.2 Třída SpravceAnimaci

Třída je napsaná podle návrhového vzoru Singleton, tento vzor má výhodu v tom, že v celé aplikaci bude existovat jen jedna instance této třídy, víc jich není zapotřebí. Slouží pro veškerou činnost s jednotlivými animacemi. Používá atribut pro uchování animací. Jednotlivé animace jsou uloženy v seznamu. Pro práci s daným seznamem využívá metody. Nabízí metodu pro přidání, smazání a vracení animace. Poskytuje informaci o tom, jestli stojí všechny animace, to znamená, že všechny vlaky stojí u nástupiště a žádný není v pohybu. Obsahuje metodu pro posunutí animací, které jsou v pohybu.

6.1.3 Třída Animace

Instance třídy uchovává veškeré informace o dané animaci. Má atribut souprava, kde jsou veškeré informace o vlaku, jako je délka, číslo, druh atd. Pamatuje si trasu animace, která představuje seznam kolejí seřazených sestupně. Při aktualizaci trasy se počítá délka kroku, která je pro každou animaci odlišná. Délka kroku se uchovává v atributu delkaKroku. Dalším důležitým atributem je poloha, představující polohu bodů na daných kolejích. Index bodu v poli odpovídá indexu v poli indexKoleje, kde je informace, na které koleji se nachází daný bod. Pomocí těchto bodů se vykresluje celý vlak, vykreslení vlaku je popsáno v předešlé kapitole 3.3.1 Modelování v simulačním modulu aplikace a 3.3.2 Volba křivky. Třída obsahuje metodu pro posunutí animace, která posune všechny body o danou délku kroku a zavolá metodu pro vykreslení vlakové soupravy. Metodu volá instance třídy SpravceAnimaci, pokud daná animace nestojí, pro tuto informaci slouží vlastnost Stoji, která vrací pravdivostní hodnotu. Další důležitá vlastnost je JeKonec, která se hlavně nastavuje u odjezdu vlaku na hodnotu TRUE, aby vlak po dojezdu na konec trasy byl smazán ze seznamu animací.

6.1.4 Třída ZpracovaniZprav

Třída obsahuje jen metody, které slouží pro zpracování animační zprávy. Je jen jedna veřejná zpráva DorucitZpravu, kde se v parametru zprávy předává animační zpráva. Každá animační zpráva obsahuje kód, podle kterého metoda zavolá pří-

slušnou funkci na zpracování zprávy a předá zprávu dál. Třída rozlišuje devět kódů animačních zpráv.

Tab. 6.2 – Seznam kódů animační zprávy

Kód anim. zprávy	**Popis zpracování zprávy**
PrijezdVlaku	Metoda nejdříve sestaví vlak pomocí třídy SestaveniVlaku, která vrátí instanci třídy Souprava. Dále se najde pro vlak postavená cesta ve třídě InfrastrukturaView. Na základě těchto informaci se vytvoří animace, která vypočítá polohy jednotlivých bodů na trati pro vykreslování vlaku. Animace se přidá do seznamu všech animací.
OdjezdVlaku	Funkce zaprvé získá instanci třídy Animace ze seznamu animací. Vyhledá postavenou cestu, kterou přidá do animace. Nastaví vlastnost Stoji na hodnotu FALSE a vlastnost JeKonec zas na hodnotu TRUE. Pokud vlak nepokračuje stejným směrem jako u příjezdu, potom se otáčí celá souprava.
DojezdVlaku	Ověří se, zda už vlak dojel. Má povolenou velmi malou toleranci. Slouží to ke kontrole, jestli stav simulačního jádra odpovídá stavu animace.
PostavenaCesta	Metoda vykreslí koleje z cesty barvou, která je určená pro postavenou cestu. Pokud kolej je součástí výhybky, tak danou barvou vykreslí celou výhybku. Nakonec vykresli vlak, kterému cesta patří, jestliže existuje.
RezervovanaCesta	Zde to je podobný jako u zprávy o postavení cesty, jen se koleje vykreslí barvou pro rezervování cesty.
UvolnenaCesta	Vykreslí koleje původní barvou, vlak už neexistuje.
NoveCisloVlaku	Metoda vyhledá patřičnou animaci ze seznamu a aktualizuje informace v objektu Souprava, který je jeden z atributů třídy Animace.

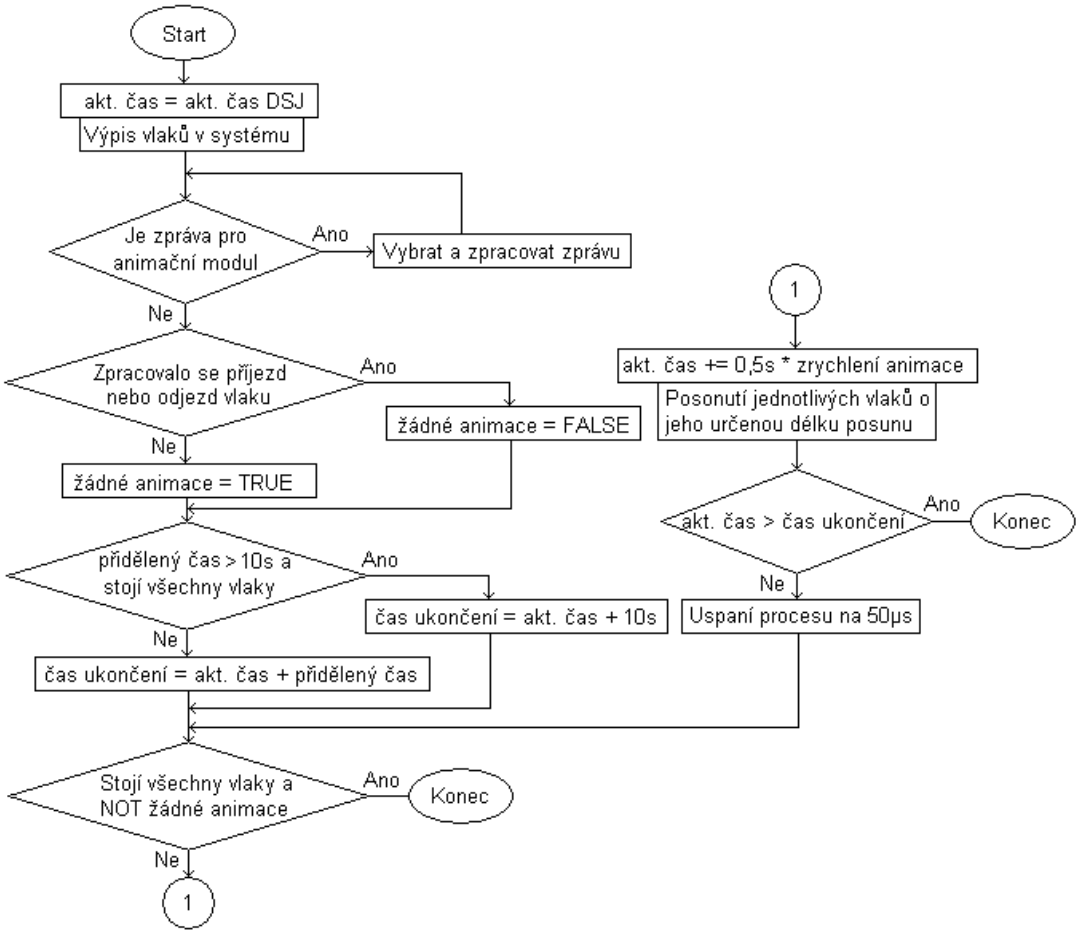
SpojeniVlaku	Najdou se dvě příslušné animace, které jsou ze seznamu animací vymazány. Získají se koleje, na kterých vlaky stály. Dále se zjistí poloha předku a zadku vlaku na koleji u obou animací. Porovnává se vzájemné postavení obou vlaků, můžou nastat tři možnosti (obr. 6.3). Na základě těchto informací se určí bod pro nový vlak a vytvoří se animace, která se přidá do seznamu animací. Nový vlak stojí na kolejích předešlých vlaků.
RozpojeniVlaku	Najde správnou animaci, která je ze seznamu animacích vymazána. Potom se získají koleje, na kterých vlak stojí a poloha předku a poloha zadku na kolejích. Dále se zjišťuje vzájemné postavení rozpojených souprav. Na základě těchto informací se vytvoří dvě nové animace a přidělí se jim koleje, na kterých stál původní vlak. Vypočítají se jednotlivé body pro animace a vlaky se vykreslí. Obě animace se přidají do seznamu animací.

Obr. 6.3 – Možnosti postavení vlaků před spojením

6.2 Jeden krok animačního jádra

Krok animačního jádra zajišťuje metoda KrokAnimacnihoJadra, v atributu metody se předává doba určená pro animační jádro. Tato funkce je ve třídě AnimacniJadro a volána z diskrétního simulačního jádra. Nejdříve se zpracují všechny animační zprávy, které jsou ve vyrovnávací paměti. Po zpracování zpráv se aktualizuje čas simulace podle diskrétního simulačního jádra. Pokud se zpracovaly jen zprávy mimo těch, které se týkají příjezdu nebo odjezdu vlaku, a doba pro animaci je delší než deset sekund a zároveň není žádný vlak v pohybu, tak doba animace je zkrácená na dobu 10 sekund. Potom se provádí cyklus, kde se postupně navyšuje čas simulace,

a animují se vlaky, které jsou v pohybu. Cyklus je ukončen, pokud už není žádný vlak v pohybu nebo vypršel čas určený pro simulační jádro. Po skončení cyklu je metoda ukončena a vrací se řízení simulačnímu jádru.



Obr. 6.4 – Vývojový diagram animačního kroku

6.3 Pohyb animace v aplikaci

Pohybující těleso nebo obrázek docílíme pravidelným posunutím o určitou délku kroku. Tyto dva faktory ovlivňují rychlost a plynulost animace. Posunutí objektu docílíme tak, že posuneme jeho body a vykreslíme objekt znova. Ještě je potřeba starý objekt vymazat. Není vhodné vymazávat celý objekt, ale jen část, o kterou jsme se posunuli. Protože se většinou posouvá o necelou délku objektu, aby byla animace plynulá, tak by nám vznikalo při animaci nežádoucí blikání objektu.

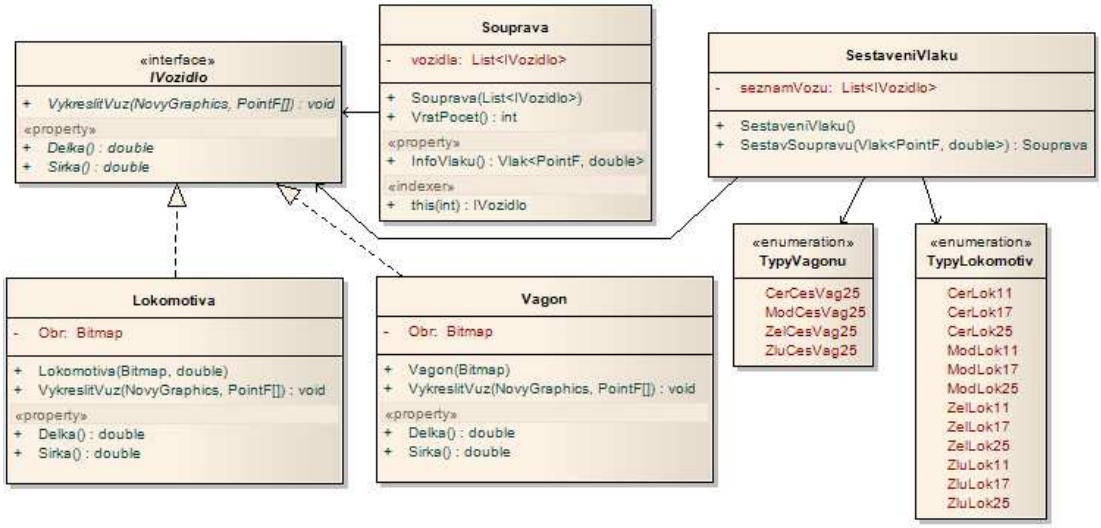
6.3.1 Vypočet délky kroku

Zaměříme-li se na skutečnost, že vlaky jezdí po koleji, tak výpočet bude určovat velikost úseku, o který se vlak posune na koleji. Pro správný výpočet potřebujeme znát délku trasy, kterou musí vlak urazit, a dobu na překonání vzdálenosti trasy. Při vytváření animace a přidání trasy se předává trasa, která už tyto informace má. Další údajem pro výpočet je změna času simulace při jednom kroku animačního jádra. V aplikaci se změní simulační čas o půl sekundy. Tímto známe všechny potřebné údaje. Vlastní výpočet se podobá výpočtu fyzikální veličiny rychlosti, která se počítá podílem délky úseku a doby překonání úseku. Délka kroku se počítá jako podíl délky trasy a počtem sekund na překonání trasy, celá hodnota se ještě dělí dvěma, protože při změně délky se čas mění jen o půl sekundy.

6.3.2 Rychlost animace

Zrychlení a zpomalení animačních prvků se už očekává u simulačních nástrojů. První, co každého napadne, je zrychlit nebo zpomalit čas animace. Ale nastává problém u vytvořených animací, které mají spočítanou délku kroku, že se délka musí zvětšit nebo zmenšit. Nejdříve bylo uvažováno, že se bude měnit interval mezi jednotlivými kroky animačního jádra, ale aby animace vypadala plynule, je tato doba 50 mikrosekund. V tomto případě již ale není možné dosáhnout velkého zrychlení. Nejjednodušší volbou, která mě napadla, je přidávat násobek rychlosti. Tato hodnota je jako konstanta, kterou se násobí časová změna simulačního času při jednom kroku animace a délka kroku, o kterou se animace posune.

6.4 Sestavení vlaku



Obr. 6.5 – Struktura tříd kolem třídy Souprava

Na obrázku výše lze vidět strukturu tříd, které souvisejí se sestavení vlaku pro animaci. Souprava vlaku se skládá z lokomotivy a vagónů. Lokomotiva určuje směr jízdy vlaku.

6.4.1 Rozhraní IVozidlo

Rozhraní využívá třídy Lokomotiva a Vagon. Tím můžeme oba objekty rozlišit. Třídy mají společnou vlastnost pro uchování délky a šířky vozidla, dále metodu pro vykreslení vozidla. Díky implementaci rozhraní, můžou být třídy uchovaný v jednom seznamu, který představuje celý vlak. Třídy Souprava a SestaveniVlaku používají toto rozhraní.

6.4.2 Třída SestaveniVlaku

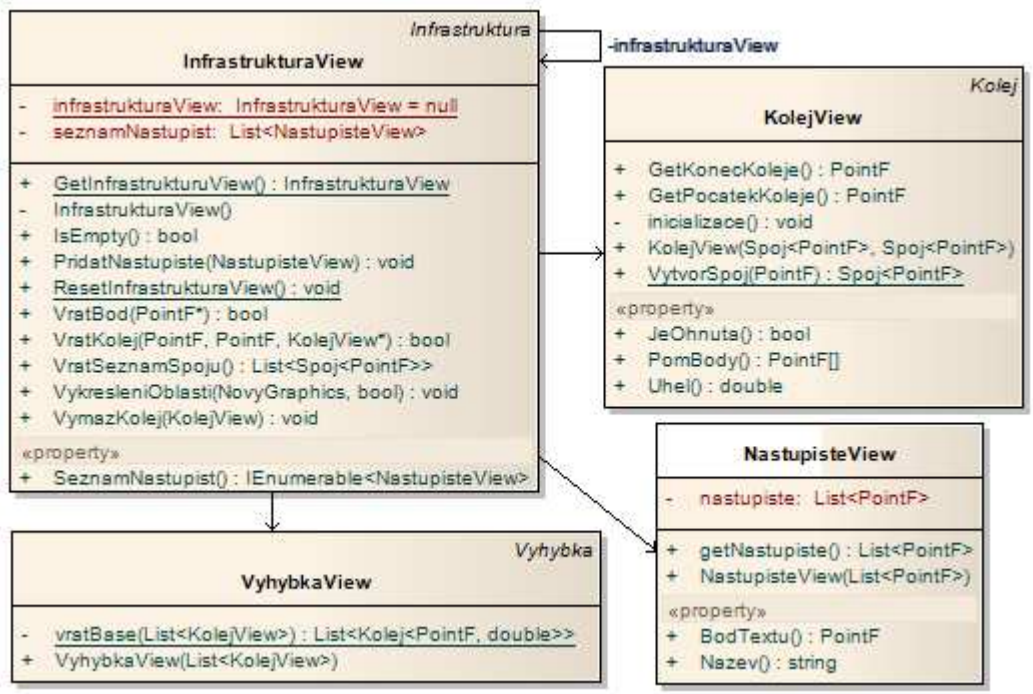
Třída pomocí metody SestavSoupravu, kde se v parametru předává objekt Vlak, který nese veškeré informace o vlaku. Zde je potřeba znát jen dvě důležité informace, jedna z nich je délka vlaku a druhá druh vlaku. Třída obsahuje dva výčtové typy TypyVagonu a TypyLokomotiv. Vagóny jsou stejně dlouhé, proto obsahuje jen čtyři hodnoty, které rozlišují barvu vagónu. Lokomotiv je více, protože nabývají různé délky. Celý vlak může být vykreslen jednou barvou ze čtyř barev. Každá barva je určená pro určitý druh vlaku. Žlutá barva je pro Pn a Mn, modrá barva je rezervová-

na pro EC, IC a SC, červenou barvu mají Ex, R a Sp a zelenou barvu vlastní Os. Na základě druhu vlaku se vybere barva vlaku a začne se vlak sestavovat. Nejdříve se připojují vagóny, které mají stejnou délku, a odečítá se délka vagónu z celkové délky. Pokud už zbývá délka rovna nebo menší než 25, tak se připojí lokomotiva o velikosti zbylé délky. U lokomotivy se pamatuje skutečná zbylá délka, proto aby odpovídala délka soupravy skutečné délce. Používá víc rozměrů délky lokomotivy jedné barvy, aby při vykreslování nedocházelo k velké deformaci obrázku.

6.4.3 Třída Souprava

Objekt této třídy se ukládá do animace, obsahuje seznam vozidel, který tvoří celý vlak. Třída Souprava nabízí indexový přístup k seznamu, kde na první pozici je lokomotiva a za ni následují vagóny. Dále uchovává veškeré informace o vlaku.

6.5 Uchování statických prvků v simulaci



Obr. 6.6 – Struktura pro uchování kolejí a nástupišť

Všechny třídy jsou poděděné od tříd v diskrétním simulačním jádru až na třídu NastupisteView, protože simulační jádro nepotřebuje do svého systému zahrnovat nástupiště v železniční stanici.

6.5.1 Třída KolejView

Třída je poděděná od třídy Kolej z diskrétního simulačního jádra. Navíc obsahuje funkce pro vrácení počátku a konce koleje v souřadnicovém systému XY, uchovává pomocné body pro Bézierovu křivku, pokud je kolej ohnutá. Pamatuje si úhel svírající s vodorovnou osou.

6.5.2 Třída VyhybkaView

Tato třída po podědění je úplně totožná, že by se dědit nemuselo, ale animační jádro pracuje s třídou KolejView, tak i výhybka musela být poděděná, protože uchovává seznam kolejí tvořící danou výhybku. Díky dědění je práce s výhybkami jednoduší.

6.5.3 Třída NastupisteView

Třída používá seznam bodů, které tvoří polygon nástupiště. Dále uchovává název a bod levého-horního rohu, kde začíná text.

6.5.4 Třída InfrastrukturaView

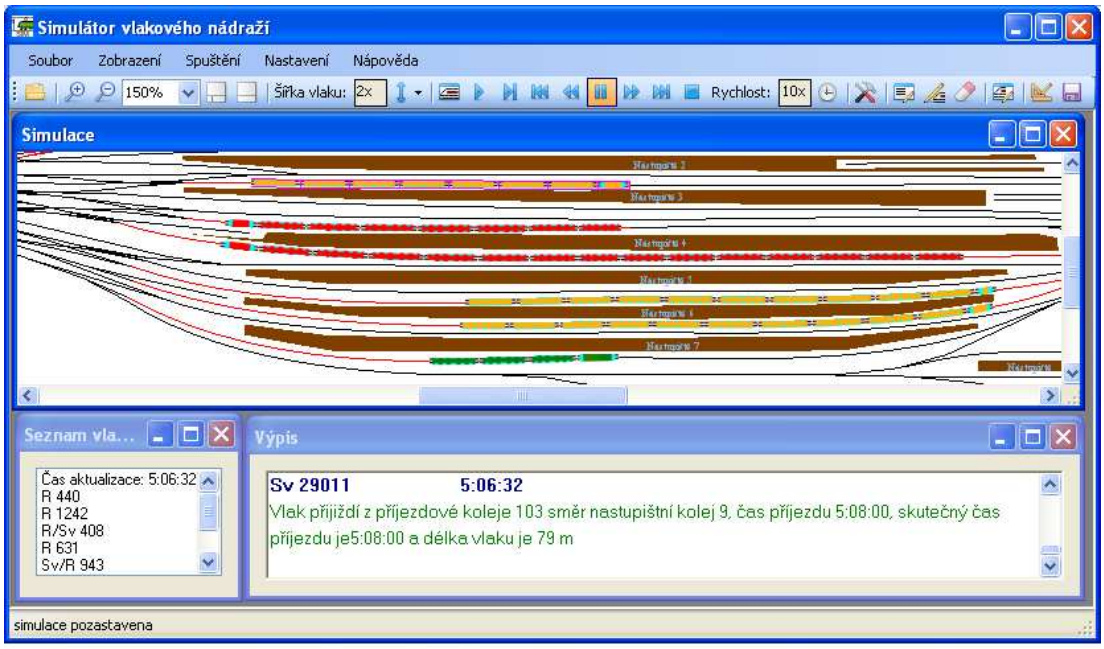
Třída je potomkem třídy Infrastruktura z diskrétního jádra. Díky podědění už obsahuje seznamy kolejí, rezervovaných kolejí, obsazených kolejí a výhybek. Navíc jsem implementoval seznam nástupišť, který není potřeba v simulačním jádru. InfrastrukruraView je naprogramován podle návrhového vzoru singleton, ten zaručuje, že instance třídy bude jen jedna v celé aplikaci.

7 Simulátor

Popíšeme si hlavní cíl diplomové práce, kterým je naprogramovat simulační nástroj železniční stanice založený na agentově orientovaném výpočetním jádru. Simulátor nabízí grafické zobrazení simulačních procesů.

7.1 Popis aplikace

Aplikace je naprogramována v jazyce C#, je formulářového typu a nabízí grafické zobrazení simulace. Program disponuje moderním rozhraním MDI (Obr. 7.1), které umožňuje v hlavním okně otevřít další okna, konkrétně uživatel může mít otevřené jedno okno, ve kterém se animuje stav simulace, a druhé okno vypisuje průběh simulace. Program zobrazuje v modálním okně čas simulace, toto okno má vlastnost, že je vždy navrchu všech oken, to dovoluje uživateli si okno kamkoliv posunout pro přehlednost. Aplikace je jednoduchá na ovládání, hlavně díky horní nabídce tlačítek a voleb zobrazení.



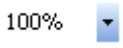
Obr. 7.1 – Zobrazení oken v MDI aplikaci

7.1.1 MDI aplikace

Zkratka MDI (Multiple Document Interface) znamená, že můžete v aplikaci otevřít více dokumentů nebo oken najednou. Aplikace neumožňuje spuštění více simulací, ale dovoluje mít otevřené zároveň okno simulace a výpisu. S okny se dobře pracuje, protože jsou otevřeny v jednom hlavním okně. Program umožňuje uživateli jedním kliknutím na tlačítko srovnat okna nad sebe a tím zpříjemňuje práci. Tlačítko bude popsáno v následující kapitole.

7.1.2 Nabídka horní lišty

Tab. 7.2 – Popis jednotlivých položek v nabídce.

	Otevře se dialogové okno pro výběr souboru s příponou .xml, cesta k soubor je předána metodě NacistInfrastrukturu(string soubor). Metoda celý soubor projde a uloží do paměti celou infrastrukturu nádržní stanice. Soubor je uložen pomocí značkovacího jazyka XML, který je po otevření čitelný v jakémkoliv textovém editoru.
  	Přibližuje nebo oddaluje zobrazení simulace. Kliknutím na plus nebo mínus se změní procentní zobrazení o 25%. Dále si můžeme zvolit hned vlastní hodnotu. Aktuální hodnoto je uložená v uživatelském nastavení aplikace. Hodnota v paměti je uložena pomocí datového typu integer. Veškeré objekty jsou vykreslovaní pomocí bodů v souřadnicovém systému XY. Hodnota procentního zobrazení je dělená stem a touto hodnotou se násobí jednotlivé souřadnice bodu. To nám umožňuje mít infrastrukturu a animované objekty uloženy v původních souřadnicích, které nejsou potřeba přepisovat a při změně zobrazení se musí jenom znova vykreslit.

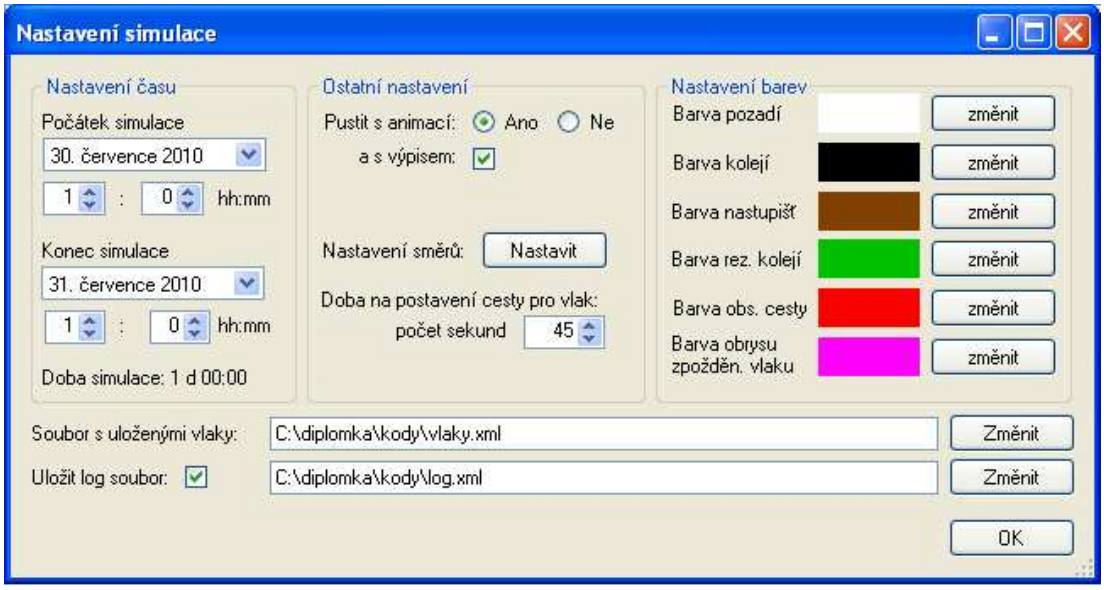
 	Okna se srovnají nad sebe, nahoře je okno pro zobrazení simulace a pod ním okno na výpis animačních zpráv. Metoda nejdříve maximalizuje hlavní okno, potom se zjišťuje prostor pro zobrazení vnitřních oken. Jednoduchá je šířka prostoru, kde zjistíme šířku hlavního okna a odečteme určitou menší hodnotu (třeba 15), že každé okno má kolem sebe nepatrný rám. Výška se získá podobným způsobem, musí se jen odečíst výška celé horní nabídky. Výška se pak rozdělí tak, aby zobrazovací okno zabíralo 2/3 prostoru a druhé okno 1/3. Druhá volba srovná okna s dalším oknem pro výpis aktuálních vlaků v sytému, které umístí dole vlevo.
Šířka vlaku: 	Zde se určuje násobek šířky vlaku. Skutečná šířka je velice malá k poměru infrastruktury a vlaky by mohli být pro někoho špatně viditelné. Proto lze vlak rozšířit až o trojnásobek své šířky. Bohatě postačuje dvojnásobek.
	Okno pro animaci se maximalizuje a tím se ostatní okna zakryji nebo se zobrazí v původní velikosti.
 	Obě volby spouští běh simulace, jen je mezi nimi jeden rozdíl a to, že první volba volá pořád dokola funkci KrokDSJ() dokud diskrétní simulační jádro nesignalizuje konec. Druhá volba ho jen zavolá jednou a zas musíte klinout. Umožňuje si to trochu krokovat simulaci. Funkce krokuj je volána ve vytvořeném novém vláknu aplikace, protože kdyby byla volána ve vláknu, v kterém běží hlavní aplikace, tak by hlavní okno přestalo reagovat. Aplikace by vypadala jako zamrzlá a to je nežádoucí.
     	Tlačítka ovlivňují průběh simulace, první dvě zpomalují animaci. Třetím tlačítkem simulaci pozastavím a také znovu rozeběhneme. Předposlední dvě zas zrychlují průběh animace. Poslední tlačítko kompletně zastaví simulaci a zruší instance animačního a simulačního jádra.

Rychlost:	Zobrazuje násobek zrychlení animace, je to nejjednodušší a neefektivnější řešení. Protože každá animace se pohne o jinou délku kroku během jednoho animačního kroku, kterému odpovídá půl sekundy simulačního času. Hodnota jen znásobí tyto změny a odpadne složité přepočítávání délky kroku. Podrobněji se budeme této problematice věnovat později.
	Zobrazí dialog, kde se vypisuje aktuální čas simulace.
	Otevře okno, kde kompletní nabídka pro nastavení simulace a barev simulačního prostředí.
	Okno pro výpis animačních zprav nebo seznamu s vlaky se maximalizuje a tím se ostatní okna zakryji nebo se zobrazí v původní velikosti.
	Vypíše se seznam vlaků do okna výpisu.
	Smaže se obsah okna výpisu.
	Okno pro výpis aktuálního seznamu vlaků v systému se maximalizuje a tím se ostatní okna zakryji nebo se zobrazí v původní velikosti.
	Otevře se okno pro editaci infrastruktury.
	Uloží se aktuální infrastruktura nádražní stanice do souboru v značkovacím jazyku XML.

7.2 Nastavení simulace

Před spuštěním simulace je dobré si zkontrolovat vstupní údaje simulátoru. Všechny údaje jsou v dialogovém oknu Nastavení simulace (obr. 7.3), které nachází v menu pod pojmem Nastavení. Délka trvání simulace je určena od počátečného data a času do konečného data a času. Důležité je, aby nebyla záporná hodnota doby simulace. Dále se nastavuje cesta k souboru, kde je seznam vlaků, které projedou železniční stanicí za jeden den. Nabízí se tu volba, jestli se má simulace spustit s animací, nebo ne a jestli se mají vypisovat animační zprávy s aktuálním seznamem vlaků v systému. Nastavuje se tu i tabulka směrů, kde se určuje, z jaké vstupní koleje se lze dostat na nástupištní koleje a jakou rychlostí. Určuje se doba postavení cesty

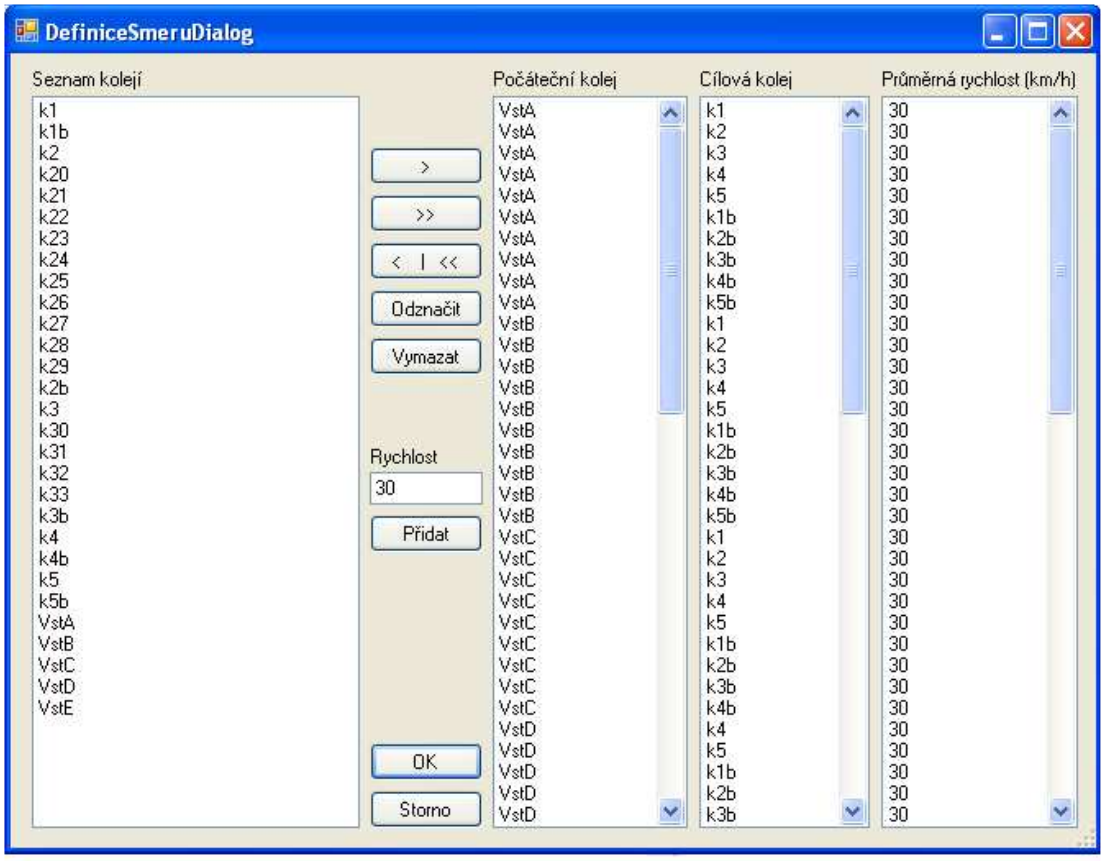
pro vlak. Velmi příjemně si tu uživatel může upravit barevné zobrazení, které mu bude vyhovovat. Uživatel si může zvolit volbu, jestli chce logovat průběh simulace a zvolit si cílový adresář, kde bude soubor uložen. Soubor si bude moc kdykoliv prohlížet a vytahat si z něho vlastní statistické údaje, které ho budou zajímat.



Obr. 7.3 – Dialogové okno Nastavení simulace

7.2.1 Nastavení směrů

Pokud má uživatel už připravenou infrastrukturu železniční stanice a popsané koleje, musí ještě nastavit všechny směry a průměrnou rychlost na trati. Nastavuje u každé vstupní koleje do systému nástupištní kolej (cílová kolej), kam může vlak dojet. Pokud se vlak dostane ze vstupní koleje VstA na nástupištní koleje k1, k1b, k2 a k2b, tak bude muset nastavit osm směrů pro vstupní kolej, protože se musí zadat ještě opačné směry. Ovládání dialogu je intuitivní a snadné. První sloupci vybírá kolej a pomocí prvního tlačítka vloží kolej do sloupce Počáteční kolej a druhým zas do sloupce Cílová kolej.

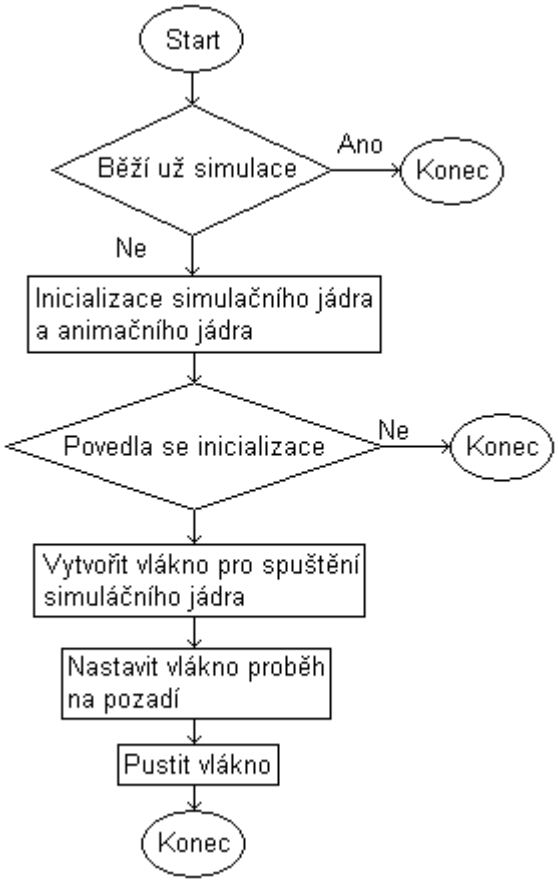


Obr. 7.4 – Dialogové okno pro nastavení směrů

7.3 Spuštění simulace

Než se simulace spustí, musí aplikace vykonat plno počátečních kroků. Nejdříve se zkontroluje existence infrastruktury a cesta k souboru s vlaky. Potom se inicializuje animační jádro s dvěma parametry, v prvním parametru se předává ukazatel na funkci, která vrací třídu Graphics pro danou komponentu na vykreslování animace. V druhém parametru se předává ukazatel na funkci, která vypisuje aktuální čas simulace. Potom se inicializuje diskrétní simulační jádro, v parametrech se hlavně předává ukazatel na metodu, která volá animační jádro, aby začalo vykonávat svojí činnost po dobu, která mu byla přidělena. Dále se předává rozhraní pro volbu koleje, pokud je kolej obsazena jiným vlakem. Simulačnímu jádru se ještě předává třída ParametrySpusteni, kde se nastaví veškeré parametry, jako počátek a konec simulace, infrastruktura nádražní stanice, seznam vlaků, čas pro postavení cesty atd. Nakonec je ještě animačnímu jádru předána instance diskrétního jádra, pro zjišťování potřebních informacích například vyhledání cesty pro vlak, zjištění aktuálního času

simulace a při překreslování zjišťovat rezervované a obsazené koleje. Pokud se inicializace zdařila, tak celá simulace je spuštěná v novém vláknu, aby hlavní okno aplikace nebylo zamrzlé a reagovalo na pokyny od uživatele.

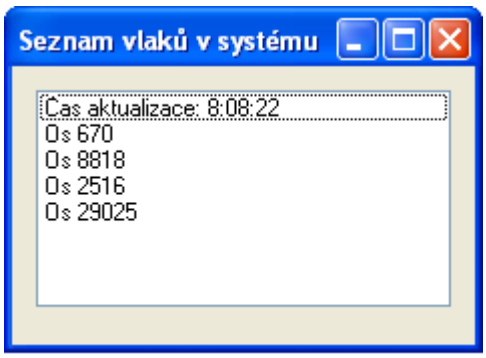


Obr 7.6 – Vývojový diagram spuštění simulace

7.4 Výpis aktuálního seznamu vlaků v systému

Uživatel má přehled o vlacích, které jsou zaregistrovány v systému. Tato informace je pro něho velmi důležitá, protože nemůže vidět všechny vlaky, buď má zobrazenou jen část železniční stanice, ale to by moc nevadilo. Horší problém je, že přijíždějící vlaky se zobrazí v animaci až, když mají postavenou cestu a mohou jet od vstupního návěstidla. Tím může nastat problém, že začne jeden vlak čekat, až se mu zarezervují všechny potřebné koleje na postavení cesty a začne blokovat další vlaky, a simulace by se mohla dostat do nekompatibilního stavu. Tento seznam se aktualizuje pokaždé před začátkem činnosti animačního jádra. Vypisují se i vlaky

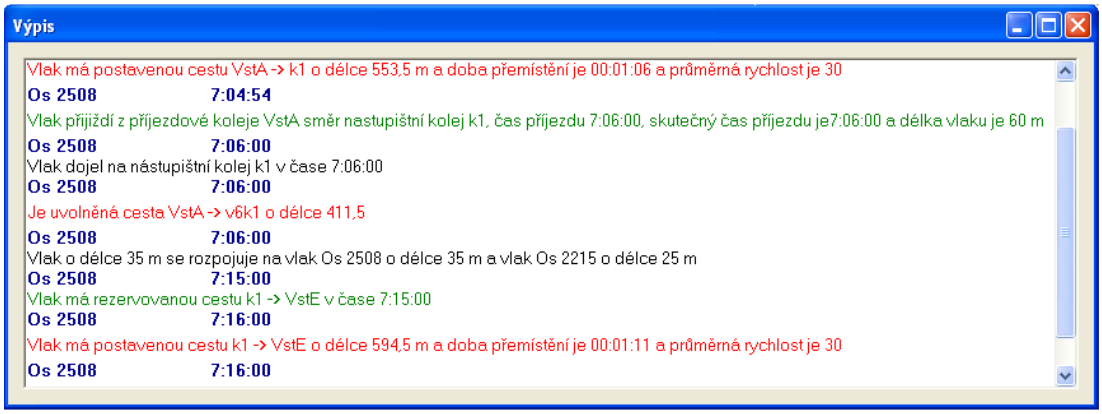
čekající na postavení cesty a tím uživatel vidí, jestli se mu v systému nehromadí neodbavené vlaky.



Obr. 7.7 – Zobrazení aktuálního seznamu vlaků v systému

7.5 Výpis animačních zpráv v aplikaci

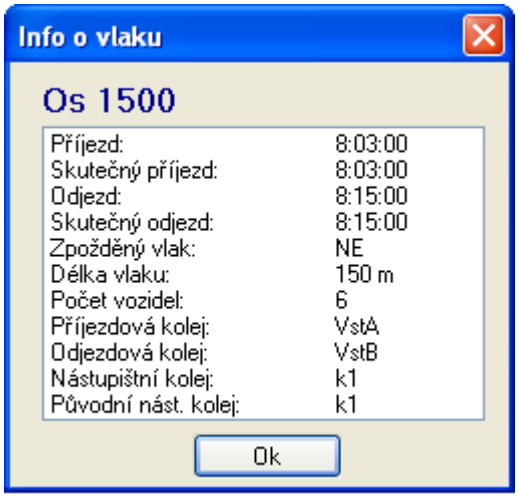
Během spuštěné simulace se vypisují informace z animačních zpráv v pořadí, tak jak je zpracovává animační jádro. Formát zprávy je jednoduchý a přehledný, nejdříve se vypíše druh a číslo vlaku, zatím aktuální simulační čas a na dalším řádku se vypíšou informace, které zpráva obsahuje. Ve zprávách o přijíždějícím a odjíždějícím vlaku se vypíše odkud, kam se přemístí vlak, aktuální čas, čas podle jízdního řádu a délku vlaku, pokud vlak je zpožděný, tato informace se vypisuje na úrovni výpisu druhu a čísla vlaku. U postavené cestě pro vlak se zobrazí ve výpisu informace o délce cesty, doba přemístěná po cestě, průměrná rychlost a odkud kam vede. U rezervované a uvolněné cesty to už není, tak podrobné. Informace o jednotlivých délkách při spojovaní nebo rozpojování souprav se vypisují pří zpracovaní animačních zpráv o spojení nebo rozpojení vlaku. Další důležitou zprávou je zpráva o přečíslování vlaku, kde se hlavně zobrazí informace o novém číslu vlaku. Tento výpis hodně pomáhá v orientaci průběhu simulace. Pokud je požadavek na ještě lepší orientaci ve vlacích, je možnost si simulaci pozastavit a projít jednotlivé vlaky v systému (o tomto tématu je uvedeno více v následující kapitole).



Obr. 7.8 – Zobrazení výpisu animačních zpráv během simulace

7.6 Zobrazení informací o vlaku

V průběhu simulace se pohybují v systému různé vlakové soupravy. Během spuštěné simulace je těžké zjišťovat informace o daném vlaku v systému, protože vlak je v pohybu, při přečíslování, rozpojovaní a spojování se mění instance daného vlaku. Ale pokud je animace pozastavena a klikneme na daný vlak, zobrazí se dialogové okno s informacemi o vlaku (obr. 7.9). Vypisuje se je příjezd a odjezd vlaku, dále skutečné údaje o příjezdu a odjezdu, které odpovídají jízdnímu řádu. Pokud je příjezd nebo odjezd déle než skutečný, tak se vlak označí jako zpožděný. Dále je tu informace o délce vlaku a počtu vozidel včetně lokomotivy a údaje o příjezdové, odjezdové koleji a nástupištní koleji.



Obr. 7.9 – Informační dialog o vlaku

7.7 Výpis vstupního souboru s vlaky

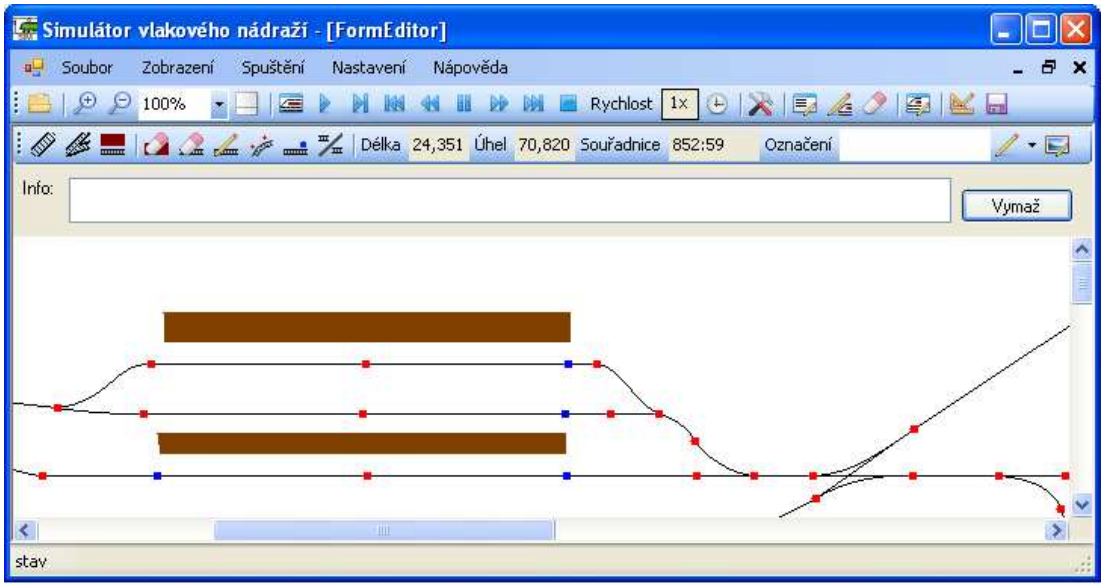
Předem si uživatel může zkontrolovat seznam vlaků, který obsahuje vstupní soubor pro simulaci. Soubor by měl obsahovat všechny vlaky, které projedou stanicí za celý týden. U jednotlivých vlaků lze nastavit jízdu jen v určitých dnech, pokud se tyto omezení nenastaví, počítá se s tím, že jezdí každý den. Ve výpisu se zobrazují všechny informace, jak o času příjezdu, tak i odjezdu. Vypisuje se příjezdová kolej a odjezdová a nástupištní kolej. Vypisují se i dny, pokud má vlak omezení, rozpojování a spojování souprav atd.

Obr. 7.10 – Výpis vlaků ze vstupního souboru

7.8 Editace infrastruktury

Aplikace umožňuje graficky editovat infrastrukturu železniční stanice. Snadno lze přidat nebo odebrat koleje a tím si ověřit nějaké předpoklady chování želez-

niční stanice. Je možno si dopředu simulovat přestavbu kolejiště, jestli se navrhované řešení vyplatí nebo ne. Protože takovéto přestavby jsou cenově náročné a je vhodné vědět dopředu, jestli se investice vyplatí. V editoru lze také přidat nebo odebrat jednotlivá nástupiště. Vykreslení kolejí v editoru se příliš neliší, jen jsou navíc označený body spojů kolejí a body zastavení (obr. 7.11). Červené body označují spoje jednotlivých kolejí a modré spoje ukazují body zastavení, kde zastavuje vlak u nástupiště. Aktuální pozice myši se zobrazuje v kolonce souřadnice, kde počátek souřadnicového sytému je v levém horním rohu editoru. Pokud se táhne kolej nebo hrana nástupiště, tak se zobrazuje svírající úhel mezi hranou a vodorovnou osou a také se zobrazuje délka úsečky. Další možností je stisknout pravé tlačítko myši a údaj o úhlu a délce se zobrazí. Lze tak měřit údaje ještě před kreslením. Do kolonky označení se píše název kreslicího objektu. Kolonka info slouží jako malá nápověda, jak pracovat s editorem. Vypisuje pokyny, které se mají vykonat a informace o průběhu kresleného objektu.



Obr. 7.11 – Editační zobrazení infrastruktury

7.8.1 Popis editačních prvků

Tab. 7.12 – Popis editačních prvků

	Kreslení koleje. Předem se může vyplnit kolonka pro označení. Kolej se kreslí tahem od počátku ke konci. Počátek se volí stiskem levého tlačítka na myši a konec uvolněním. Pokud počátek nebo konec je zvolen už v existujícím spoji, tak se tento spoj přiřadí ke koleji, aby spojené koleje měli totožný spoj.
	Výhybka se kreslí podobně jako kolej, taháte jednotlivé koleje, které tvoří výhybku. Jednotlivé koleje lze pojmenovat. Po vytažení všech kolejí je potřeba stisknout pravý tlačítko myši aby se výhybka uložila do infrastruktury a pokud je vyplněný označení, tak se uloží i označení výhybky.
	Nástupiště se kreslí taháním hran, až poslední hranu natáhnete do počátku první hrany, tak se nástupiště vybarví. Potom vyplníte onačení a pravým tlačítkem myši určíte levý horní bod na nástupišti, odkud bude začínat text popisu a nástupiště se uloží do infrastruktury.
	Vymaže se nástupiště z infrastruktury, tak že se klikne myší na dané nástupiště.
	Vymaže se kolej, tím že označíte tažením od počátku ke konci koleje, jako při kreslení koleje. Pokud je kolej součástí výhybky, tak se vymaže celá výhybka.
	Vybere se kolej pro editovaný následujících třech editačních prvků. Kolej se vybere jako při mazání.
	Ke koleji se přidají další dva body, tažením, jako při kreslení koleje. Tyto dva body s bodem počátku a konce tvoří kubickou Bézierovu křivku.
	Bod zastavení se udává hlavně na kolejích u nástupiště, aby vlak zastavil na konci nástupiště. Bod zastavení se určuje pro daný směr pohybu na koleji. Tahem úsečky od spoje koleje ve směru pohybu do pozice na koleji, kde má být bod zastavení. Tímto se bod uloží ke koleji.

	Ke koleji se přidá související kolej. Kolej se přidá stejným způsobem, jako při výběru koleje pro editaci. Tato vlastnost se využívá u kolejí, které jsou u nástupiště. Většinou podél nástupiště se koleje značí pro levou a pravou část zvlášť (např. kolej 1 a kolej 1b).
Označení: 	Toto tlačítko se nachází za kolonkou označení. Slouží pro přejmenovaní koleje, výhybky nebo nástupiště. Po kliknutí se rozjede nabídka, kde se vybere kolej, výhybka, nebo nástupiště. Kolej se určí výběrem jako pro editaci koleje, u výhybky se vybere jedna její kolej. Nástupiště se vybere kliknutím na něho a tím se určí levý horní roh počátku textu.
	Vloží se obrázek ze souboru. Tento obrázek může sloužit jako podklad pro kreslení infrastruktury, pokud je ve správném měřítku nebo jestli je znám velikost nějakého úseku, tak lze změnit procentuální zobrazení, obrázek se velikostně nezmění, ale koleje už kreslíte v jiném rozměru. Při 100% zobrazení odpovídá jeden pixel jednomu metru ve skutečnosti.
Rozměry plátna:	V kolonce se přepíšou hodnoty ve formátu „šířka:výška“ a údaje se potvrdí tlačítkem Enter. Pokud se infrastruktura nevejde na plátno, tak se může rozšířit a rozšíří se směrem doprava a dolu.
Posun Infrastruktur: 	Do kolonky před tlačítkem se napíše hodnota o kolik pixelů se má celá infrastruktura posunout a potom se zvolí směr v nabídce, která vyjede pod tlačítkem.

7.9 Informativní hláška

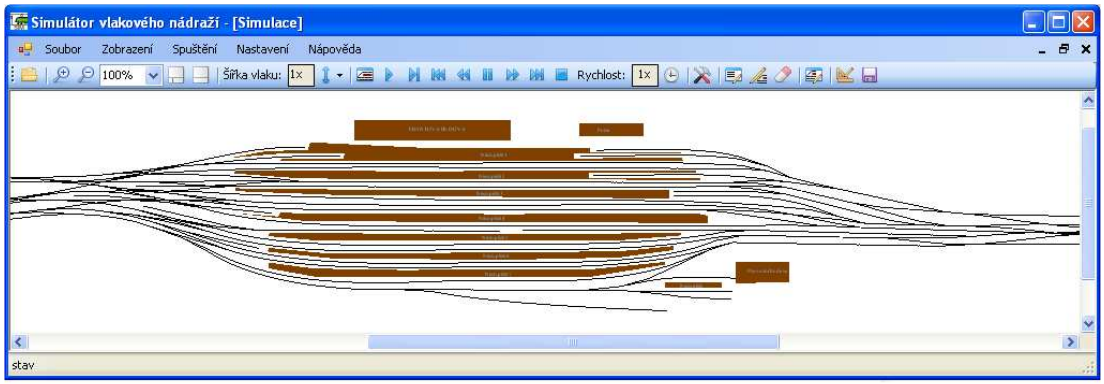
Informativní dialog (obr. 7.13), byl zaveden pro kontrolu, jestli odpovídá animační stav simulačnímu stavu. Pokaždé kdy simulační jádro ukončí pohyb vlaku, tak je informováno o tom animační jádro. Tato situace nastává velmi ojediněle, většinou vlaků chybí malá část úseku, která je menší než jeho délka kroku. Příčinou je, že při počítání dochází k velmi malému zaokrouhlování při počítaní délky kroku, protože se rychlost předává v km/h, která se přepočítává na m/s, další vliv je délka úseku

a vlaku. Vznikne tím, že animační jádro musí už vrátit řízení a zpětným vrácením dostane informaci, že vlak už dojel. V dialogu se vypíše, kterého vlaku se to dotýká a kolik metrů mu chybělo. Na testovacím kolejišti se tato informace vůbec neobjevovala. Při testování na hlavním nádraží Prahy se objevila jen párkrát při odjezdu vlaku ze systému, když se simulovalo 24 hodin. Překonávají se tu větší vzdálenosti a délky vlaků. Pokud se simuluje 24 hodin, což je jeden celý den, kdy se provede tohle testování minimálně 700krát a hláška vyskočila asi 5krát, tak se informace může zanedbat. Uživatel má jistotu, že stav animačního jádra odpovídá stavu simulačního jádra.

Obr. 7.13 – Informativní hláška o nedojetí vlaku včas

8 Vyhodnocení aplikace

Simulační nástroj se otestoval na železniční stanici Praha hlavní nádraží. Jízdní řád vlaků a informace o jednotlivých vlacích je přepsán z grafikonu a tabulek, které poskytl dispečink stanice. Ještě bylo k dispozici schéma a neúplný výkres železniční stanice. Výkres se použil jako podklad do editoru aplikace, kde se podle něho natahaly koleje a výhybky. Výsledná infrastruktura je vidět na následujícím obrázku (obr. 8.1). Příjezdové a odjezdové koleje končí u příjezdových a odjezdových návěstidel, které taky tvoří ohraničení sytému.



Obr. 8.1 – Zobrazení Hlavního nádraží v Praze

8.1 Nastavení rychlosti jízdy vlaku do stanice

Je potřeba nastavit jednotlivé směry v infrastruktuře, jedem směr představuje jednu vstupní koleji a nástupištní kolej, na kterou může vlak přijet. Aktuální nastavení lze vidět v následující tabulce. Tabulka představuje jednoduchý výpis, že v aplikaci se tyto údaje nastavují složitě, protože se umožňuje nastavit rychlost pro určitý příjezd a odjezd z dané nástupištní koleje zvlášť, více v předešlé kapitole 7.2.1 Nastavení směrů.

8.1.1 První testovaní aplikace na žst. Praha hlavní nádraží

Při začátcích testování bylo nastaveno na všech směrech průměrná rychlost 30 km/h a doba postavení cesty trvala 1 minutu. Už jen při krátkém běhu simulace docházelo k zahlcení vlaků, které žádaly o přidělení koleje, protože jejich kolej byla obsazena. Problém byl hlavně ze směru Vyšehrad a Vršovice, kde vlaky musí urazit poměrně dlouhou trasu, než dojely k nástupišti, délka trasy se pohybuje okolo 800 metrů. Jejich pomalou jízdou blokovaly výhybky pro ostatní vlaky a tím se zvyšovalo zpoždění vlaků a obsazení kolejí.

Zvýšením průměrné rychlosti ze všech směrů na 50 km/h a zkrácením doby postavení cesty na 45 sekund se docílilo velkého zlepšení. Simulace mohla bez problémů běžet celý den, jen v několika případech se musela přidělit jiná kolej z důvodu obsazení koleje plánované podle jízdního řádu.

8.1.2 Rychlost při skutečném testování

Abychom věrohodně otestovali chování aplikace je potřeba vycházet trochu z reality a přiblížit se rychlosti, která odpovídá skutečnosti. Po rekonstrukci severního zhlaví žst. Praha hlavní nádraží byla zvýšená rychlost z 35 km/h na 50 km/h (Zdroj: [13]) ve všech směrech. Průměrné rychlosti byly probrány ještě s technologem. Technolog doporučil průměrnou rychlost v okolí nástupišť asi 35 km/h a ve větších vzdálenostech od nástupišť 55 km/h. Těchto hodnot jsem se držel při určování průměrné rychlosti z daného směru. Protože je různá délka kolejí z každého směru, byl počítán vážený průměr ze dvou určených rychlostí a váhou byla délka úseku, který se projížděl danou rychlostí. Vypočítané hodnoty jsou v následující tabulce. Doba na postavení cesty byla natavena na 45 sekund.

Tab. 8.2 – Tabulka rychlostí

Koleje	Směr příjezdu a odjezdu	Průměrná rychlost daného směru
201, 202	Vyšehrad	50,1 km/h
101, 102, 103, 105	Vršovice	50,3 km/h
301, 302	Holešovice, Vysočany	48,2 km/h
601, 602	Libeň	48,4 km/h

8.2 Výsledky

Simulace běžela 24 hodin simulačního času. Začátek spuštění bylo v jednu hodinu ráno až do druhého dne do jedné hodiny ráno, byl zvolen v kalendáři pracovní den. Během této doby projede nádražím asi 650 vlaků včetně přečíslování, z toho 46 vlaků přijelo nebo odjelo zpožděně a 5 vlakům byla přidělena jiná kolej u nástupiště, protože kolej podle jízdního řádu byla obsazena. Zpožděný se počítá, pokud pravidelný příjezd se liší od skutečného více jak 30 sekund. Zpoždění vznikalo tím, že vlak čekal na postavení cesty. Počet 46 vlaků k poměru 650 je vyhovující výsledek a můžeme říct, že se vlaky na vstupu do systému nehromadily.

8.3 Hodnocení aplikace

Aplikace bez vážných problémů běžela 24 hodin simulačního času, přitom nepřestala reagovat na uživatelské pokyny. Rozeběhnutá simulace šla kdykoliv pozastavit, zrychlit i zpomalit. Skutečného času simulace byla spuštěna asi 20 minut. Během simulace byly puštěny všechny výpisy i log od simulačního jádra do souboru. Tímto hodnotím aplikace za stabilní a kompletně funkční.

9 Závěr

Hlavním cílem diplomové práce byl vývoj simulačního nástroje železniční stanice založený na agentově orientovaném simulačním jádru. Aplikace obsahuje a umožňuje:

- animační okno (uživatel vidí vykreslené kolejiště a jednotlivá nástupiště a po kolejišti jezdí jednotlivé vlaky, tento vizuální pohled je nejpřehlednější),
- okno pro vypisování animačních zpráv a seznamu vlaků (zároveň lze vidět i textový výpis, kde se vypisují informace z animační zprávy, třeba délka vlaku, délka cesty, číslo vlaku atd. Nebo lze vypsat seznam vlaků, které tvoří vstupní proud do simulačního jádra),
- okno pro výpis aktuálního seznamu vlaků v systému (zobrazuje přehled vlaků v systému, že všechny vlaky nemusí být vidět a čekají na postavení cesty při příjezdu, tak lze vidět, jestli se systém nepřehlcuje),
- zobrazuje simulační čas v samostatném dialogu (dialog má vlastnost, že vždy navrchu ostatní oken, tak je vždy vidět a muže si ho uživatel přesunout na jakékoliv místo v obrazovce),

- horní lištu s nabídkou (nabízí přibližování a oddalovaná zobrazené infrastruktury, rozšiřovaní vlaku, ovládaní průběhu simulace, srovnávaní oken atd. Celá aplikace se jednoduše intuitivně ovládá),

- nastavení simulace (uživatel si nastaví pomocí formuláře všechny vstupní parametry simulace a barevné zobrazení, formulář obsahuje příjemné ovládací prvky jako kalendář, paletu barev, dialogové okno pro určení souboru a adresáře atd.),

- editor infrastruktury (uživatel si může překreslit, nakreslit celou infrastrukturu nebo doplnit koleje, či výhybky, tím může testovat různé změny v kolejišti)

Podle mého názoru jsem hlavní cíl diplomové práce splnil. Aplikace je kompletně napsaná a funkční. Splňuje základní požadavky, které se očekávají od simulačního nástroje. Obsahuje grafické zobrazení simulace, které není jednoduché implementovat. Jednoduché ovládání, což je pro uživatele příjemné. Aplikace je napsána s využitím objektově orientovaného programování, což umožňuje jednoduchou implementaci dalších funkcí.

Diplomová práce byla pro mě velice přínosná, prohloubil jsem si programátorské dovednosti. Hlavně v programování formulářové aplikace a zobrazení grafiky ve 2D. Ohledně aplikace jsem musel prostudovat práci s vlákny a prohloubil jsem si znalosti v železniční dopravě.

10 Seznam použitých zdrojů

[1] KAVIČKA, Antonín; KLIMA, Valent; ADAMKO, Norbert. *Agentovo orientovaná simulácia dopravných uzlov*. Žilina : EDIS-vydavatelstvo ŽU, 2005. 206 s. ISBN 80-8070-477-5.

[2] HUSÁKOVÁ, Martina. *Agentově orientované modelování : Znalostní technologie III* [online]. [s.l.], [2007]. 12 s. Materiál pro podporu studia. Univerzita Hradec Králové. Dostupné z WWW: <http://lide.uhk.cz/fim/ucitel/fshusam2/lekarnicky/zt3/zt3_dokumenty/AgentModelSimul.pdf>.

[3] Simulace. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 19. 5. 2008, last modified on 2. 2. 2010 [cit. 2010-05-31]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Simulace>.

[4] Bézierova křivka. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2008, last modified on 16. 9. 2009 [cit. 2010-07-15]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/B%C3%A9zierova_k%C5%99ivka>.

[5] ALEXANDR, Lubomír. *Výuka počítačové grafiky cestou WWW*. Brno, 2000, [cit. 2010-07-20]. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a informatiky. Dostupné z WWW: <http://lubovo.misto.cz/_MAIL_/curves/obsah.html>.

[6] BLÁHOVEC, Václav. *Výpočetní jádro pro agentově orientovanou architekturu simulačních modelů*. Pardubice, 2010. 74 s. Diplomová práce. Univerzita Pardubice, Fakulta elektrotechniky a informatiky.

[7] KAVIČKA, Antonín; BAŽANT, Michael. Návrh infrastruktury železničních uzlů s podporou počítačové simulace : (část 1). *AUTOMA : časopis pro automatizační techniku*. 2007, 5, s. 2-3. Dostupný také z WWW: <http://www.odbornecasopisy.cz/index.php?id_document=34373>.

[8] KAVIČKA, Antonín; BAŽANT, Michael. Návrh infrastruktury železničních uzlů s podporou počítačové simulace : (část 2). *AUTOMA : časopis pro automatizační techniku*. 2007, 6, s. 2-4. Dostupný také z WWW: <http://www.odbornecasopisy.cz/index.php?id_document=34407>.

[9] KAVIČKA, Antonín; JÁNOŠÍKOVÁ, Ľudmila. Modelovanie koľajiska a výpočet najkratšej jazdnej cesty. *Komunikácie : Vedecké listy Žilinskej univerzity*. 1999, 2, s. 9-21. ISSN 1335-4205.

[10] KAVIČKA, Antonín. *Pokročilé techniky modelování a simulace: Diskrétní simulace*, Učební text Fakulty elektrotechniky a informatiky Univerzity Pardubice, Univerzita Pardubice, Pardubice 2009.

[11] VESELÝ, Petr. *Počítačová grafika 2D: Animace pohybu na dopravní síti,* Učební text Fakulty elektrotechniky a informatiky Univerzity Pardubice, Univerzita Pardubice, Pardubice 2009.

[12] KOTTNAUER, Jakub. *Vlákna v C#* [online]. 2008 [cit. 2010-08-01]. Překlad „Threading in C#“ od Josepha Albahari. Dostupné z WWW: <http://www.albahari.com/threading/threading_czech.pdf>.

[13] MITLÖHNER, Jan. Severní zhlaví – Praha Hlavní nádraží. *Silnice železnice* [online]. 26.1.2009, [cit. 2010-08-01]. Dostupný z WWW: <http://www.silnice-zeleznice.cz/clanek/severni-zhlavi-praha-hlavni-nadrazi/>. ISSN 1803-8441.

[14] *SŽDC : Správa železniční dopravní cesty* [online]. 2010-06-13 [cit. 2010-08-04]. Knižní jízdní řády. Dostupné z WWW: <http://www.szdc.cz/provozovani-drahy/knizni-jizdni-rady-130610.html>.