

České vysoké učení technické v Praze  
Fakulta elektrotechnická



Diplomová práce

## **Simulátor řízení železniční dopravy**

*Jan Čihák*

Vedoucí práce: Ing. Šimon Nešvera

Studijní program: Elektrotechnika a informatika, dobíhající

Obor: Informatika a výpočetní technika

leden 2007



## Poděkování

Na tomto místě bych chtěl poděkovat všem, kteří se mnou měli a mají trpělivost. Především vedoucímu mé práce Ing. Šimonu Nešverovi.

Z externích spolupracovníků a konzultantů bych rád poděkoval Mgr. Evě Zelové za korekturu textu, Ing. Lukáši Sedláčkovi za konzultační činnost, JUDr. Martinu Piškulovi za vytvoření reálných stanic, Ivu Strašilovi za editor Stag, Milanu Zánovi za editor Builder, Lukáši Procházkovi za uživatelskou příručku. Dále všem, kteří přispěli k dokončení mé diplomové práce, zejména zaměstnancům ČD.

V neposlední řadě patří můj dík mé rodině a mé snoubence, kteří mě v mé práci stále podporují.



## **Prohlášení**

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 8.1. 2007

.....



## **Abstract**

The thesis deals with the implementation of a control simulator of railway transport, which can be used as teaching aid for future CD dispatchers, who can attempt common operations via up-to-date timetables. After detailed examination of present simulators, and consultation with train dispatchers from actual operations, the resulting project Staničář was developed. This project is characterized primarily by liberated stations in the format of XML, more dispatchers that can cooperate in the same time (network simulation), and also the possibility of linking up other modules of control.

## **Abstrakt**

Diplomová práce se zabývá implementací simulátoru řízení železniční dopravy, která může být využita jako učební pomůcka pro budoucí dispečery ČD. Dispečeri si tak mohou vyzkoušet běžný provoz dle aktuálního jízdního řádu. Po důkladném prozkoumání současných simulátorů a po konzultacích s výpravčími z reálného provozu vznikl projekt Staničář. Tento projekt se vyznačuje především uvolněním formátu stanice ve formě XML, kooperací více dispečerů (síťová simulace) a možností napojení jiných modulů řízení.





# Obsah

Seznam obrázků	xiii
<b>1 Úvod</b>	<b>1</b>
<b>2 Popis problému, specifikace cíle</b>	<b>2</b>
2.1 Popis problému	2
2.1.1 Zabezpečovací zařízení	2
2.1.1.1 Autoblok	3
2.1.1.2 Návěstní soustava	4
2.1.1.3 Manipulace s vlaky	6
2.1.2 Jednotné obslužné pracoviště	6
2.1.2.1 Definice	6
2.1.2.2 Práce dispečera	6
2.1.2.3 Důvody ke vzniku simulátoru	7
2.1.3 Požadavky na simulátor	7
2.2 Cíle	7
2.3 Struktura práce	8
2.4 Rešerše	8
2.4.1 Existující systémy	8
2.4.2 ESA simulátor	8
2.4.3 REMOTE simulátor	9
2.4.4 Ostatní simulátory	9
2.4.5 Shrnutí	10
<b>3 Analýza a návrh řešení</b>	<b>11</b>
3.1 Obecně	11
3.1.1 Úvod	11
3.1.2 Ovládací prvky	11
3.1.3 Jízdní cesty	11
3.1.3.1 Typy cest	11
3.1.3.2 Stavění	11
3.1.3.3 Zásobník povelů	12
3.1.3.4 Rušení cesty	13
3.1.4 Vlaky	13
3.1.4.1 Manipulace s vlakem	13
3.1.4.2 Jízda vlaku	14
3.1.5 Spolupráce dispečerů	14
3.1.6 Návěstní soustava	14
3.2 Volba prostředí	14
3.3 Návrh řešení	15
3.3.1 Generické jádro	15
3.3.2 Hledání cesty	15
3.3.3 View	15
<b>4 Realizace</b>	<b>16</b>

4.1	Model stanice . . . . .	16
4.1.1	Třída TPolicko . . . . .	17
4.1.1.1	Kolejový obvod . . . . .	17
4.1.2	Třída TNavestidlo . . . . .	17
4.1.2.1	Vstupní a výstupní body . . . . .	17
4.1.2.2	Náhodná doprava . . . . .	18
4.1.2.3	Návěstní soustava . . . . .	18
4.1.2.4	Autoblok . . . . .	18
4.1.3	Třída TSouhlas . . . . .	19
4.1.4	Třída TZasobník . . . . .	19
4.1.5	Třída TPrejezd . . . . .	20
4.2	Komunikace . . . . .	20
4.2.1	Strana serveru . . . . .	20
4.2.2	Strana klienta . . . . .	21
4.3	Model vlaku . . . . .	21
4.3.1	Popis struktury . . . . .	21
4.3.2	Jízdní řád . . . . .	22
4.4	Jízdní cesty . . . . .	23
4.4.1	Stavba cesty . . . . .	23
4.5	Pohyb vlaku . . . . .	25
4.5.1	Zastávky . . . . .	27
4.5.2	Otočení vlaku . . . . .	27
4.5.3	Přenos čísel vlaku . . . . .	27
4.5.4	Opuštění modelu stanice vlakem . . . . .	28
4.6	Modul JOP . . . . .	28
4.6.1	Simulační cyklus . . . . .	28
4.6.2	Manipulace s vlaky . . . . .	28
<b>5</b>	<b>Testování</b>	<b>28</b>
5.1	Builder . . . . .	29
5.2	Stag . . . . .	29
5.3	Stanice . . . . .	30
5.4	Simulace . . . . .	30
5.5	Mantis . . . . .	30
<b>6</b>	<b>Závěr</b>	<b>31</b>
6.1	Zhodnocení . . . . .	31
6.2	Možné pokračování . . . . .	31
<b>7</b>	<b>Literatura</b>	<b>33</b>
<b>A</b>	<b>Seznam použitých zkratk</b>	<b>35</b>
<b>B</b>	<b>UML diagramy</b>	<b>37</b>
<b>C</b>	<b>XML Schema</b>	<b>41</b>
<b>D</b>	<b>Uživatelská / instalační příručka</b>	<b>51</b>





## Seznam obrázků

2.1	Prvky kolejiště . . . . .	2
2.2	Autoblok . . . . .	4
2.3	Diagnostika návěstních světél . . . . .	5
2.4	ESA - Inverzní podoba reliéfu kolejiště . . . . .	9
3.1	Návrh komunikace modulů . . . . .	16
4.1	Informace o vlaku v simulaci . . . . .	29
B.1	Diagram tříd pro políčko mapy (1. část) - zobrazeny jsou jen některé metody	38
B.2	Diagram tříd pro políčko mapy (2. část) - zobrazeny jsou jen některé metody	39
E.1	Seznam přiloženého CD . . . . .	53



# 1 Úvod

Železniční doprava patří do života téměř každého z nás. Zahrnuje nákladní i osobní dopravu. Zdá se, že železniční doprava ustupuje, ale skrývá se v ní potenciál, kterého se zajisté v budoucnosti využije.

Možná že některé pasažéry napadají otázky typu: „Proč má ten vlak zpoždění?“ nebo „Proč jede jiný vlak dříve než druhý?“. Odpovědí na tyto dotazy se měl stát program, jenž by osvětlil základní dění na železnici. Tento program by měl sloužit jak zájemcům o problematiku řízení železniční dopravy, tak i dispečerům ČD a jiným zaměstnancům ČD. Zájem o tento projekt mě příjemně překvapil, protože se týkal hlavně z řad lidí, kteří jsou nebo byli s železnicí nějakým způsobem spojeni. Oslovili mě především výpravčí, dispečerů, a to i zahraniční, ale také učitelé středních škol. Vyhověl jsem tedy jejich požadavkům a začal vyvíjet simulátor řízení železniční dopravy.

První krok k dokonalému výsledku je nastudování problematiky. Je potřeba získat praktický pohled na náplň práce dispečera, řízení železnice a prodiskutovat její okruh se zaměstnanci v oboru. Využil jsem nabídek několika zaměstnanců ČD a navštívil několik nádraží v České republice. Návštěvy mi umožnily poznat nová zařízení a seznámit se se současnými simulátory, které budou využívány novými dispečery. Prostudoval jsem drážní předpisy a základní standard jednotného obslužného pracoviště (JOP).

Požadavky byly z různých oblastí zájmů, jak profesionální, tak laické. Snažil jsem se navrhnout systém, který zachová simulátoru co největší reálnost, ale i odpoví na většinu otázek laikům. Vznikl projekt „Staničář“, který má nezávislé řídicí jádro, ke kterému je možné připojit libovolné moduly řízení. Protože je složitost problému příliš obsáhlá, po domluvě s vedoucím diplomové práce bude zpracován pouze jeden modul ovládacího rozhraní, a to modul splňující standard JOP.

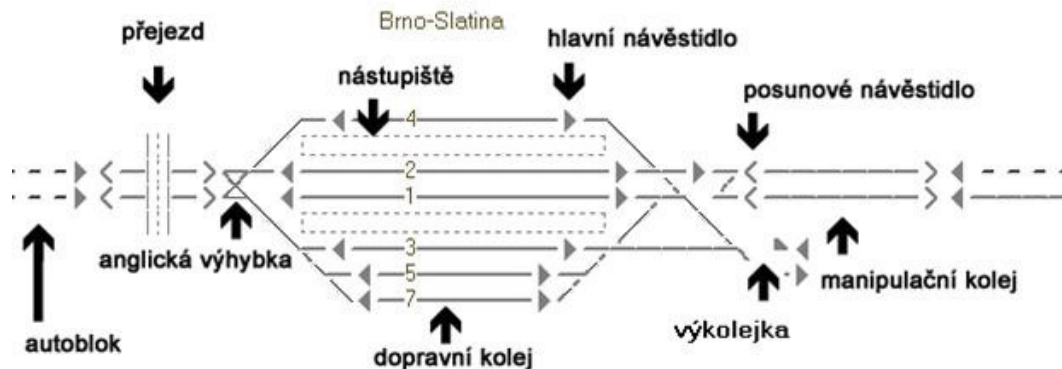
## 2 Popis problému, specifikace cíle

### 2.1 Popis problému

#### 2.1.1 Zabezpečovací zařízení

Každé zabezpečovací zařízení musí splňovat především kritérium bezpečnosti. U systému, kde může dojít k ohrožení lidského života, je kladen nejvyšší důraz na snížení rizik vzniku chyby. Proto se vyvíjí nástroje, které tato rizika snižují. Vždy však mějme na paměti, že v každé situaci hraje roli lidský faktor a že žádný systém se neobejde bez lidského faktoru; ať již se jedná o údržbu celého systému nebo o selhání softwaru nebo hardwaru. Především na selhání musí být řízení připraveno a musí existovat možnost, jak se bez kontrolní jednotky obejít.

Pro snadnější pochopení práce zabezpečovacího zařízení je dobré seznámit se se základními pojmy železnice. Na modelovém kolejišti vidíme tyto základní prvky:



Obrázek 2.1: Prvky kolejiště

Na modelovém kolejišti vidíme především koleje. Dle využití jednotlivé koleje mají upřesňující názvy. Manipulační kolej slouží k manipulaci s vlaky. Dopravní kolej je využívána pro nákladní a osobní dopravu. Autoblok je prvek se samočinnou řídicí činností. Pro možnost volby směru se používají výměny. Podle jejich konstrukčních vlastností mají různé názvy, např. anglická výhybka.

Kolejový obvod (KO) je souvislý úsek izolované koleje. Ten může být dlouhý jen několik metrů, ale i kilometrů. Důležité je, že celý obvod se chová jako jednotný celek. V případě, že se kdekoliv nachází vlak, celý úsek je obsazen.

Hlavní a posunové návěstidlo slouží k předání informace strojvedoucímu o stavu úseku za návěstidlem. V dnešní době jsou nejrozšířenější návěstidla světelná. Zjednodušeně je možné si představit, že mají stejnou funkci jako semafor na silniční křižovatce. Celý návěstní systém bude vysvětlen dále.

Přejezdové zařízení slouží jako ochranná křižovatka železniční a silniční dopravy. Nás zajímají především světelná přejezdová zařízení, která zajišťují bezpečnost pomocí

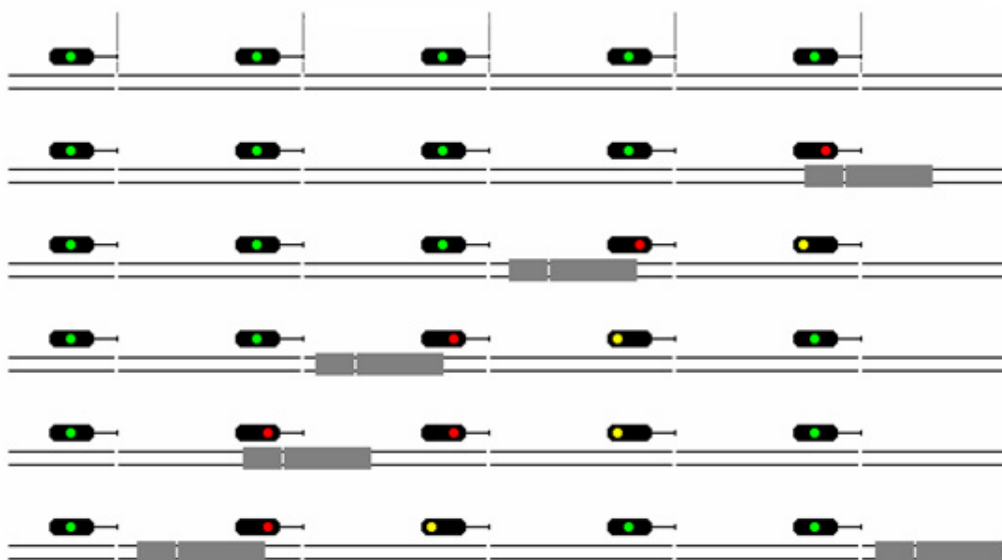


výstražných světel nebo závor. Blíží-li se k přejezdu vlak, je nutné signalizovat včas blížící se nebezpečí.

Prvek nástupiště umožňuje kontakt mezi osobní dopravou a cestujícími. Není možné provozovat osobní přepravu mimo nástupiště. Důležitost tohoto prvku je možné si představit při výlukách některé z dopravních kolejí a při přesedání cestujících. V případě, že dochází k přesunu osob z jednoho nástupiště na jiné, je nutné si uvědomit vzdálenost mezi nimi. Může tak docházet k nepříjemným zpožděním.

### 2.1.1.1 Autoblok

Autoblok je část zabezpečovacího zařízení, který umožňuje automatickou bezpečnou činnost. Na obrázku je vidět úsek skládající se z šesti kolejových obvodů a pěti návěstidel. Návěstidla mohou mít pouze následující tři stavy: VOLNO (svítí zelené prostřední světlo) — vlak může jet a následující návěstidlo má opět návěst VOLNO, VÝSTRAHA (svítí horní žluté světlo) — vlak může opatrnou jízdou jet a očekávat na následující návěsti STŮJ a STŮJ (svítí dolní červené světlo) — vlak musí zastavit.



Obrázek 2.2: Autoblok

Nyní si podrobněji popíšeme, co se děje na předcházejícím obrázku. Vlak jede zprava doleva, proto i kolejové obvody a návěstidla budou označována zprava doleva čísly od jedné do šesti. První řádek ukazuje volný úsek. Žádný KO není obsazen, nikde se nenachází vlak, všechna návěstidla ukazují VOLNO. Druhý řádek ukazuje situaci, kdy vlak vjel do druhého úseku a zároveň je ještě v prvním úseku. První návěstidlo změnilo svůj stav na STŮJ, protože úsek před návěstidlem je obsazen. V dalším řádku přešel celý vlak do třetího KO. Druhé návěstidlo nyní ukazuje návěst STŮJ. První návěstidlo návěst VÝSTRAHA, protože následující KO za návěstidlem je volný a na následujícím návěstidle svítí STŮJ. Ve čtvrtém řádku je vlak již na čtvrtém KO. První, druhé a třetí návěstidlo postupně ukazuje VOLNO, VÝSTRAHA a STŮJ. V dalších krocích se zo-

brazují světla dle popsaných pravidel výše. V posledním řádku vidíme, že do úseku vjel další vlak. V mezistaničním úseku se většinou na každý KO vejde celý vlak, proto může jet za sebou až šest vlaků.

S autoblokem někdy bývá svázán prvek traťový souhlas, jedná se o tzv. poloautoblok. K autobloku je přidána část řízení, která ovlivňuje směr, po kterém se mohou vlaky po autobloku pohybovat. Souhlas je řízen dispečery na jednotlivých koncích úseku, kde je autoblok zaveden. Změnu směru, kterým je souhlas udělen, je možné provést jen v případě, že je celý úsek volný a není již na autoblok vyslán vlak.

### 2.1.1.2 Návěstní soustava

Představíme-li si všechna návěstidla ve stanici, dostáváme celek zvaný „návěstní soustava“. Návěstidlo plní dvě základní funkce: upozorňuje na omezení rychlosti vlaku za návěstidlem, dále předvěstí znak na příštím návěstidle.

Rychlost určuje spodní část světelných návěstidel. Omezení rychlosti je závislé na celém úseku až k následujícímu návěstidlu. Velkou roli hrají především výhybky, na kterých záleží, zda vlak jede rovným směrem nebo do odbočky. Dále ovlivňuje rychlost technické omezení výhybky. Výrobci zpravidla uvádějí jinou rychlost pro směr po hrotu a proti hrotu. Hrotem se rozumí konec pohyblivé části výměny. Poloha hrotu ovlivňuje průjezdný směr.

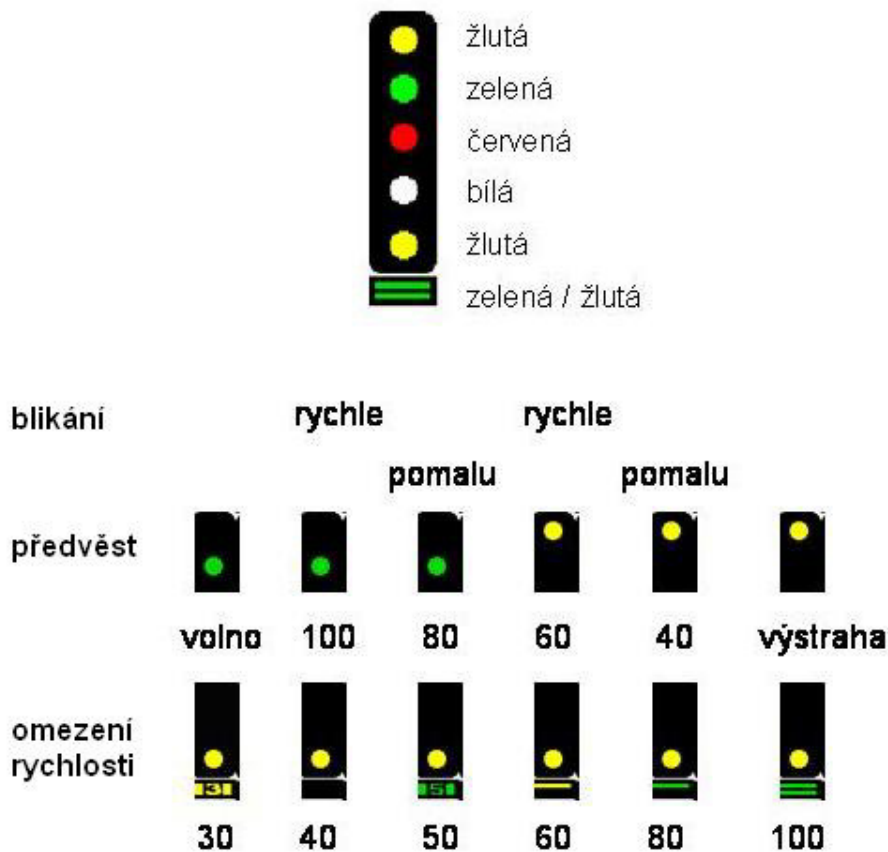
Předvěstní znak upozorňuje strojvedoucího, co má očekávat na příštím návěstidle. Neznamena to však, že tomu tak skutečně bude. Modernější hnací vozidla mají vyřešenu komunikaci s návěstidlem, a to tak, že dochází k přenosu znaku na řídicí desku hnacího vozidla. Jak funguje předvěstní znak jsme již viděli na autobloku. V obvodu stanice však bývá značně rozšířen. Přehled je znázorněn na obrázku 2.3.

Horní a dolní znak lze libovolně skládat. Můžeme tak získat návěst „40 a očekávej volno“, která znamená, že maximální rychlostí 40 km/h (dolní znak) a volno, další úsek bez omezení. V horní polovině svítí zelené světlo, dole svítí žluté světlo. Jiná možnost „100 a očekávej 40“ znamená, že vlak má jet rychlostí maximálně 100 km/h a od příštího návěstidla pouze 40 km/h. V horní polovině návěstidla rychle bliká zelené světlo, v dolní svítí žlutá barva a dva zelené pruhy.

### 2.1.1.3 Manipulace s vlaky

Dalším úkolem zabezpečovacího zařízení je zabezpečit a umožnit základní manipulaci s vlaky. Především je to jízda vlaku, která je uskutečňována po jízdnicích cestách. Jízdní cestu si můžeme představit jako úsek, který má počátek a konec u návěstidla, a dále jako posloupnost po sobě jdoucích kolejí. Navíc jízdní cesta zajišťuje správnou polohu výměn a znemožnění postavení kolizní cesty.

Manipulací se rozumí pohyb, rozřazení, přepřah, příprež či sunutí vlaku. Rozřazení vlaku znamená, že je možné změnit uspořádání vozů u jednotlivých vlaků. Přepřah je manipulace vlaku, kdy je odpojeno hnací vozidlo, poté dochází k jeho přesunu na druhou stranu vlaku a pak je opětovně připojeno. Jinou variantou může být, že je připojena jiná lokomo-



Obrázek 2.3: Diagnostika návěstních světel

tiva (například při záměně elektrické lokomotivy za dieselovou). Přípřež je hnací vozidlo, jehož úkolem je posílit tažnou sílu vlaku. Hnací vozidlo je tak připojeno k vlaku, který již jiné hnací vozidlo má. Sunutím vlaku se rozumí pohyb vlaku, kdy ve směru vlaku není na počátku hnací vozidlo. Jednoduše řečeno, vlak „couvá“.

## 2.1.2 Jednotné obslužné pracoviště

### 2.1.2.1 Definice

Aby bylo možné vyvinout řídicí systém, musí být definováno rozhraní mezi systémem a člověkem, ale také mezi systémem a zařízením, které systém ovládá. V praxi to tedy znamená, že dispečer zadá příkaz, který se odešle systému, ten se pokusí provést všechny úkony s tím spojené, např. přestavit výměnu, blikání signalizačního světla přejezdu, a dát odezvu zpět uživateli, zda se povedlo příkaz správně provést. Může se jednat o grafickou změnu symbolů na monitoru, zvukovou indikaci nebo textovou zprávu. Takových systémů lze vymyslet jistě mnoho. Pokud by však nebyla standardizována komunikace uživatele a systému, byly by v praxi nevyužitelné. České dráhy proto vydaly směrnici ZTP - JOP (Základní technické požadavky - Jednotné obslužné pracoviště).

„Jednotné obslužné pracoviště (JOP) slouží jako rozhraní mezi dopravním zaměstnancem

*a zabezpečovacím zařízením pro operativní úroveň u elektronických stavědel, hybridních stavědel a ústřední dálkového ovládní zabezpečovacího zařízení (DOZ). Pro ovládní pomocných stavědel, popř. jiných zařízení, která nezajišťují stavění zabezpečených jízdních cest, se nasazení tohoto typu ovládací úrovně nedoporučuje.“ [4].*

### 2.1.2.2 Práce dispečera

Dispečer má zodpovědnost za plynulost a bezpečnost provozu na železnici. K tomu mu do jisté míry pomáhá pomocný systém, který mu ulehčuje práci. Moderní řízení dopravy probíhá přes osobní počítač s monitorem. Na obrazovce má k dispozici model kolejíště řízené stanice nebo oblasti. Počet dispečerů, kteří stanici řídí, je závislý na velikosti nádraží. Velká nádraží mají několik dispečerů a každý dispečer se stará o jinou část stanice, např. nádraží Praha hl. nádraží řídí pouze 4 dispečeri ve 12 hodinových směnách [5], střední nádraží již zvládne jediný zaměstnanec. Pokud jsou stanice malé, připojují se ke větším a ty nejmenší, pouze průjezdné, nepotřebují řízení vůbec, cesty se staví automaticky.

### 2.1.2.3 Důvody ke vzniku simulátoru

Naučit se ovládat jakékoliv zařízení obnáší řadu školení a především nutnou praxi. Bohužel školící simulátory se k veřejnosti nedostanou, proto je třeba udělat si představu, co vše tato práce obnáší. Problémem je i získat potřebnou praxi. Přestože existují demoverze přímo od výrobců, neposkytují to hlavní — skutečný provoz. Je možné si vyzkoušet základní ovládní, ale vlaky nejezdí dle grafikonu vlakové dopravy (GVD). Domnívám se však, že to má stejný účinek jako trenažér u autoškol, který se postupně ruší. České dráhy (ČD) mají k dispozici školící modelové kolejíště, ale počet hodin je pro jednotlivce omezen. Člověk, který není z oboru, má minimální možnost si zařízení vyzkoušet.

### 2.1.3 Požadavky na simulátor

Simulátor musí vykonávat dvě základní činnosti. Zpracovávat povely od uživatele a simulovat komunikaci mezi systémem a fyzickými prvky zabezpečovacího zařízení. Přijímání povelů od uživatele bude založeno na standardu rozhraní JOP, čímž bude zaručena reálnost simulace. Nejen ovládní, ale i vzhled bude stejný jako při reálném provozu. Zabezpečovací hardwarové zařízení bude zcela chybět. Není možné vyslat signál výhybce, aby se přehodila, a čekat, dokud nepřijde signál zpět, že se povel správně povedl. Děje mezi softwarovým a hardwarovým systémem lze simulovat reakčními časy. Povel se neprovede hned, ale až po určité době, jakoby probíhala komunikace mezi systémem a zařízením.

Dalším úkolem simulátoru bude možnost nahradit manuálně vykonávané úkony s vlaky. Dispečer se nestará o rozpojování vlaků přes zabezpečovací zařízení. K manipulaci s vlaky dochází většinou přes vysílačku. Aby však bylo možné ukázat náročnost provozu, musí být do simulátoru zapracovány základní povely.

## 2.2 Cíle

Cílem diplomové práce je rozšířit obecné povědomí o tom, jak se řídí železniční doprava. Simulátor maximálně přiblížit reálnému provozu, ale i přidat pomůcky pro snazší pochopení jednotlivých řídicích procesů na železnici. Domnívám se, že síťová simulace poskytne uživatelům větší zájem do zkoumání řízení železniční dopravy, především proto, že bude možné si vyzkoušet spolupráci sousedních dopravních uzlů — skutečnou koordinaci dispečerů. Další přínosy diplomové práce jsou shrnuty v následujících bodech:

- Představit návěstní soustavu v závislostech na rychlosti výměn a následném návěstidle, rozsvícení předvěstního a návěstního znaku. Ukázat, že obsluha zařízení je zodpovědná za vše, co dělá, že je možné při neodborné manipulaci srazit vlaky, např. postavením nezabezpečené kolizní cesty. Ukázat, že není možné urychlit stavbu jízdní cesty, protože je závislá na uvolnění závěru předchozí cesty, přehození výměn či uzavření přejezdu.
- Poskytnout dispečerům pomůcku řízení železničního provozu za reálného provozu. Simulátor se bude hodit nejen novým zaměstnancům, ale i starým při přestavbě zabezpečovacího zařízení ve stanici. S rekonstrukcí koridoru přibývá případů, kdy je změněno zabezpečovací zařízení ve stanici. Dispečeri nebudou při získávání praxe omezeni časem a budou si moci vyzkoušet simulátor doma.
- Umožnit simulaci základní manipulace s vlaky, tj. odpojení a připojení lokomotivy. Proto je nutné mít k dispozici viditelné řazení vlaku.
- Další přínos je pro příznivce železnice, kteří budou mít možnost si vyrobit libovolnou stanici, třeba i fiktivní.
- Simulace programu měla prokázat, že změna lokality železničního nádraží v Brně je zbytečná. Program nakonec využit nebyl, protože rozhodnutí přesunu padlo bez jakéhokoliv vlivu odpůrců návrhu, včetně referenda obyvatel Brna. Věřím však, že podobných možností využití se naskytne více.

## 2.3 Struktura práce

Diplomová práce byla zpracována dle implementační práce uvedené na stránkách katedry počítačů [3]. Jednotlivé kapitoly jsou koncipovány tak, že na začátku každé kapitoly bude obecný úvod do problému, poté podrobnější informace. Problematika simulací řízení železniční dopravy je velmi rozsáhlá, proto se po dohodě s vedoucím diplomové práce zabývám pouze standardem JOP.

## 2.4 Rešerše

### 2.4.1 Existující systémy

V současnosti jsou v České republice nejrozšířenější dva systémy řízení železniční dopravy, a to ESA (Elektronické staniční zabezpečovací zařízení) od firmy AŽD Praha, s.r.o.[1] a systém REMOTE od firmy AK signal Brno, a.s.[2]. Oba tyto systémy splňují standardy JOP a nabízejí kompletní řízení skutečné železnice. Hardwarové i softwarové elektronické staniční zabezpečovací zařízení zajišťují bezpečnější přepravu na železnici. Oba dodavatelé mají k dispozici demoverzi, ve které si lze vyzkoušet základní ovládání.

### 2.4.2 ESA simulátor

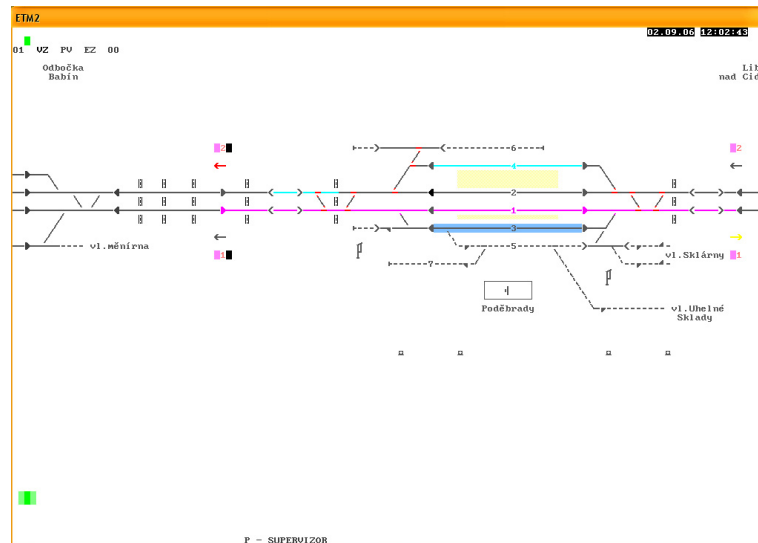
ESA zabezpečovací zařízení má velice zdařilou a propracovanou demoverzi, která poskytuje značnou část základních funkcí. Demoverze poskytuje věrné kopie skutečného reliéfu stanice, čímž umožňuje vyzkoušet si řízení stanice ještě před nasazením do reálného provozu. Existuje možnost stavění jak vlakových, tak posunových cest, a to zabezpečených i nezabezpečených. Bohužel při stavění křížových nezabezpečených cest u některých stanic celý systém spadl. Program poskytuje zobrazení stavů všech návěstidel, kolejových obvodů, přejezdů a ostatních venkovních prvků. Dále demoverze umožňuje zadávání běžných povelů, možnost výluky napěťové i kolejové. Ve srovnání s demoverzí umístěné na internetových stránkách společnosti se mi podařilo porovnat i program, který je poskytován přímo ČD. Program umožňoval navíc diagnostiku návěstidel, zobrazoval aktuálně rozsvícená světla v menu návěstidla. Dále navíc obsahoval mnohem více stanic. Nevýhodu systému vidím v manipulaci s vlaky. Ve skutečném provozu se dispečer o manipulaci nestará. Pokud provádí přeprah hnacího vozidla, přemístění lokomotivy z jednoho konce vlaku na druhý, nemusí se starat o odpojení a připojení vozů. Zde ale není možné tento úkon provést vůbec. Dalším nedostatkem je nereálnost provozu ve stanici. Vlak jede vždy, jakmile se postaví cesta. To, že vlaky nejezdí dle skutečného jízdního řádu, znemožňuje budoucímu dispečerovi snadnější nasazení do reálného provozu. Zpoždění, čekání na nástup a výstup cestujících, přípojné vlaky, to vše potřebuje mít budoucí dispečer dobře zvládnuté a zažité.

### 2.4.3 REMOTE simulátor

REMOTE je zabezpečovací zařízení od brněnské firmy AK signal Brno, a.s. Podařilo se mi získat pouze verzi, která neposkytovala mnoho funkcí. Na rozdíl od jiných simulátorů umožňoval tento simulátor přihlášení do systému v roli dispečera nebo v roli technika. Reliéf kolejiště nebyl skutečný, poskytoval pouze jedinou stanici. Nezabezpečené jízdní cesty nebylo možné stavět. V simulaci chyběla manipulace s vlaky. Nebyla zpracována jízda vlaků dle jízdního řádu nebo alespoň náhodná doprava. Demoverze slouží pouze jako představení systému. Toto zabezpečovací zařízení není tolik rozšířeno jako ESA.

### 2.4.4 Ostatní simulátory

Mezi ostatní simulátory patří velká řada simulátorů volně dostupných na internetu, které nejsou skutečnými zabezpečovacími zařízeními, např. GORDIKON [6]. Tyto simulátory však mají za cíl především pobavit a jsou proto do značné míry zjednodušovány. Především tyto simulátory nemají standardizované rozhraní JOP a kvůli přehlednosti



Obrázek 2.4: ESA - Inverzní podoba reliéfu kolejíště

se dopouští různých ústupků. Reliéf kolejíště bývá ve skutečnosti většinou přes několik obrazovek, maximálně tři obrazovky, a monitory mají větší rozlišení než nejrozšířenější mezi uživateli, a to 1024x768. Autoři simulátorů se snaží dostat model kolejíště na jedinou obrazovku. Důvodem tohoto ústupku je orientace uživatelů. Dalším neduhem je určité přizpůsobování si zobrazení některých prvků kolejíště a jejich ovládání.

Pokud si nestandardizované návyky vezmou za své budoucí dispečeri, jejich školitelé je musí přeučovat a i povědomí o řízení železniční dopravy se zkrusluje.

Výhodou těchto simulátorů je, že umožňují manipulaci s vlaky, odvěšení a přivěšení lokomotivy jsou již samozřejmostí. Setkáváme se dokonce i s možností rozřazovat si vlaky dle vlastního přání.

#### 2.4.5 Shrnutí

Ukazuje se, že dostupné simulátory mají úzce specializované cíle. Výrobci se logicky zaměřují pouze na představení vlastního zabezpečovacího zařízení, a to nebývá vždy volně dostupné. Ostatní simulátory mají za cíl spíše pobavit nebo se odchylují od JOP. Každý simulátor má své výhody a nevýhody, ale stále chybí něco, co by možností rozšiřovalo. Proto vznikl projekt Staničář, kterým se zabývám v této diplomové práci.

## 3 Analýza a návrh řešení

### 3.1 Obecně

#### 3.1.1 Úvod

Dispečer řídí vlakovou dopravu přes model kolejiště. Dispečer staví vlakové cesty a stará se o bezpečnost a plynulost odbavování vlaků. Simulátor musí splňovat základní požadavky dle předpisů JOP specifikované níže, a to jak v ovládní, tak v zobrazení. Navíc je nutné zpracovat základní úkony s vlaky, např. přepraha, přípreže, rozrazení apod.

#### 3.1.2 Ovládací prvky

Kolejiště se skládá z několika základních prvků, jako jsou koleje, výhybky, přejezdy, návěstidla, souhlasy a stanoviště obsluhy. Koleje pak mohou tvořit celek kolejového obvodu, tj. úsek, který vykazuje v každém jeho bodě stejnou vlastnost. Pokud je jen jeho část koleje obsazená, celý úsek je obsazen. To platí i o jízdnicích a posunových cestách či závěru. Autoblok je speciální úsek, ve kterém není třeba stavět jízdnicí cesty. Staví se mezi jednotlivými dopravnými. Návěstní soustava funguje automaticky a nemůže dojít k rozsvícení dovolující návěsti, pokud je předchozí úsek obsazen.

#### 3.1.3 Jízdnicí cesty

Pro pohyb vlaků po kolejišti se používají zabezpečené i nezabezpečené jízdnicí cesty. Zabezpečená jízdnicí cesta je bezpečná a před jejím postavením se hlídají všechny podmínky. Dochází ke kontrole, zda neexistuje kolizní cesta, nejsou obsazené úseky, zda výměny jsou přestaveny do správné polohy apod. Na rozdíl od nezabezpečené cesty, kdy lze poslat vlak na trať a výměny lze přehazovat těsně před začátkem vlaku. Tento úkon je velmi nebezpečný a může tak dojít k vážným dopravním nehodám, i když je rychlost vlaku v těchto případech omezena na 30 km/hod. Mohou však nastat případy, kdy by mohl vjet vlak i do jízdnicí cesty vlaku, který jede maximální povolenou rychlostí. Dispečer se stará o plynulý průjezd vlaků stavěním nekolizních cest. K dispozici může mít zásobník cest, do kterého se mohou ukládat kolizní cesty. Postavit lze i cestu, která se má automaticky stavět po projetí vlaku. Tato volba se hodí především při dálkovém ovládní malé stanice, kde vlaky jen projíždějí po dvojkolejné trati.

##### 3.1.3.1 Typy cest

Máme dva základní typy cest, a to vlakové a posunové. Vlakové cesty jsou vyznačené zeleně a slouží pro očíslované vlaky. Posunové cesty jsou značené bílou barvou a slouží pro přepraha lokomotiv a manipulaci s posunovými vlaky. Posunovou cestu je možné postavit i na obsazenou kolej.

##### 3.1.3.2 Stavění

Cesta je určena vždy počátečním a koncovým bodem, případně posloupností variantních bodů. Variantní body slouží pro případy, kdy chceme, aby se cesta vyhnula určitému úseku. Začátek cesty se volí vždy od návěstidla: hlavního nebo seřadovacího. Od hlavního



návěstidla mohou být stavěny jak vlakové, tak posunové cesty. Od seřadovacího návěstidla se staví pouze posunové cesty. U vlakových cest existují navíc možnosti volby pro automatické stavění cesty, kdy po projetí vlaku dojde k opětovnému postavení stejné cesty. Návěstidlo má ve stavu automatického stavění speciální tvar. Po splnění podmínek pro stavění cesty dojde k přestavení výměn, k uzavření přejezdů, k závěru cesty a rozsvícení návěsti na VOLNO. Podmínka uzavření přejezdu je nutná jen v případech, kdy je vlak v příliš blízké vzdálenosti od přejezdu. V tomto případě nesmí dojít k rozsvícení dovolující návěsti před uzavřením přejezdu. Při stavění mimo řízenou oblast je nutné mít udělen „souhlas“ od výpravčího v jiné oblasti. K tomu slouží stejnojmenný prvek „souhlas“. Bez udělení „souhlasu“ není možné do druhé oblasti zabezpečenou cestu postavit. Výjimkou je stavění cesty na přivolávací návěst, kde mohou nastat dvě možnosti: buď se staví nezabezpečená cesta stejně jako vlaková či posunová cesta (dojde tak při stavění k nouzovému závěru výměn a je možné stavět i na obsazenou kolej), nebo přes již postavenou jízdní cestu, druhou možností je volit konec cesty na stanovišti obsluhy, v tom případě dojde pouze k rozsvícení přivolávací návěsti. Vlak jede dle aktuálně přehozených výměn a nehlídá se, zda může dojít ke kolizi či rozřezu výhybky apod. Tato volba musí být vždy potvrzena potvrzovací sekvencí kláves „ASDF“.

### 3.1.3.3 Zásobník povelů

Při stavění jízdních cest je k ulehčení práce používán zásobník povelů. Protože doba stavění cesty trvá několik vteřin, záleží na mnoha okolnostech. Doba přehození výměn je závislá i na počasí. Záleží na tom, zda jsou výměny přehazovány postupně nebo všechny najednou apod. V případě kolizních cest není možné postavit cestu vůbec. Nové stavění cesty by bylo blokováno po celou dobu. Vlastnost zásobníku je ovlivněna třemi stavy:

- Přednostní volba

Zadávané cesty nejdou přes zásobník, jsou vykonávány přímo, a pokud není možné cestu postavit, nepostaví se a povel je zahozen. Činnost zásobníku je v tomto stavu pozastavena, a pokud v něm jsou již některé cesty navoleny, nedochází k jejich stavění.

- Volba ze zásobníku

Tento stav je používán nejčastěji. Všechny povely pro stavění cesty jdou přes zásobník a jsou řazeny na konec fronty. Zásobník je založen na funkci „first in first out“ (FIFO). Zásobník se pak stará automaticky o možnost postavení první cesty. Do zásobníku se přesouvají všechny cesty, které se staví volbou počátečního a koncového bodu na koleji. Do zásobníku se tak nezařadí povel pro stavění nezabezpečené cesty při volbě PN a volbou koncového bodu na stanovišti obsluhy. Tato volba je vždy vykonána hned.

- Editace zásobníku

V tomto režimu je možné měnit pořadí vykonávání povelů a mazat již navolené povely. Činnost během editace je pozastavena, není ani možné zadávat nové povely.

### 3.1.3.4 Rušení cesty

Protože může dojít k chybné volbě při stavění cesty, je třeba mít možnost cestu zrušit. Cestu, na které ještě není vlak, lze zrušit povelom RC. Doba, za kterou je cesta zrušena, je závislá na úseku před návěstidlem, od kterého je cesta rušena. Pokud se k návěstidlu blíží vlak, nemusel by stihnout včas zabrzdít. Proto vždy dochází nejprve k přestavění návěstidla do polohy STŮJ. V případě, že je možné cestu zrušit hned a úsek před návěstidlem je volný, je doba rušení stanovena na pět sekund. Blíží-li se již k návěstidlu vlak, záleží na typu rušené cesty. U vlakové cesty tato doba činí tři minuty a u posunové jednu minutu. Cesta, po které jede vlak, je rušena postupně podle uvolňování kolejových obvodů a uvolnění závěru cesty. Při použití přivolávací návěsti lze zrušit nouzový závěr. Tento úkon vyžaduje potvrzovací sekvenci.

Někdy je třeba vlak pouze pozdržet, proto zhasneme povolující znak příkazem STŮJ. Jakmile chceme jízdu povolit, je možné opětovně rozsvítit dovolující návěst pomocí povelu „DN“.

### 3.1.4 Vlaky

Simulátor nahrazuje skutečné chování vlaků, a proto musí umožňovat základní úkony s vlaky. Vlak se skládá z vozů, a to vagónů a lokomotiv činných či nečinných. Vlak musí být možno rozpojit. Některé vlaky se mohou rozdělit na dva. Jindy může být rozšířen počet vozů u vlaku. U nákladu mohou být odbaveny vozy každý jinam. Pro přehlednost je vhodné grafické zobrazení řazení vlaku. V simulaci se vlaky dělí na posuny a číslované vlaky. Pro posuny stačí, aby se pohybovaly v rámci jedné stanice. Slouží především pro přeprahu a příjezd vlaků z odstavných nádraží. Číslované vlaky jsou buď osobní, nebo nákladní. U osobních vlaků je třeba splnit jízdní dobu dle jízdního řádu a zastavit na všech zastávkách. Dodržovat čas na výstup a nástup cestujících je spíše otázkou dispečera. U nákladních vlaků je úkolem dopravit náklad správným směrem.

#### 3.1.4.1 Manipulace s vlakem

Přepřahem se rozumí odvěšení lokomotivy. Tak vznikne posunový vlak, kterému se postaví posunové cesty na druhou stranu vlaku, a hnací vozidlo se opět připřáhne. Úkon se používá při potřebě změnit směr vlaku.

Vlak je sunutý v případě, kdy má hnací vozidlo na konci vlaku a nemá hnací vozidlo jako první ve směru vlaku. „Couvání“ vlaku se využívá v některých případech i u číslovaných vlaků, nejen u posunů. V tomto případě může být použit postrk. Není nutné pro potřeby simulace řešit nezavěšený postrk.

Přípřeží se rozumí připojení dalšího činného hnacího vozidla k vlaku. Toho se využívá například u těžkých nákladů.

Přečíslování vlaku je úkon, kdy se mění číslo vlaku na jiné. Číslo bývá odvozeno od směru vlaku. U vlaků z jízdního řádu je třeba nabízet možnosti, na které vlak má být přečíslován.

### 3.1.4.2 Jízda vlaku

Pohyb vlaku se pozná dle změny obsazenosti jednotlivých kolejových obvodů. Je dobré si uvědomit, že elektrické hnací vozidlo nemůže jet po neelektrifikované trati. Zároveň je však nutné mít nouzové řešení pro odstranění zaseknuté soupravy. Dynamiku vlaku není nutné zpracovat do detailů. Stačí rychlost omezit dle traťové rychlosti, rychlosti stanovené návěstidlem a nejnižší maximální rychlosti všech vozů, činných a nečinných lokomotiv. Dispečeri se učí řídit železniční dopravu na modelových kolejistkách, kde je dynamika vlaků zanedbávána. Je vhodné naprogramovat funkce pro změny směru vlaku, jeho zastavení a rozjetí. Například při posunu ve skutečnosti vlak nezajíždí vždy až ke šturcu, ale zastaví hned za výhybkami. Tyto instrukce jsou však podávány v praxi ústně.

### 3.1.5 Spolupráce dispečerů

Staniční oblast vždy sousedí s jinou staniční oblastí. Může samozřejmě sousedit i s více než jednou. Z toho vyplývá, že je nutné brát v úvahu automatického výpravčího v sousedních stanicích. Automatický výpravčí mimo jiné zasílá žádost o souhlas a uděluje souhlas k vjezdu vlaku. V rámci jedné oblasti je možné předat řízení na místní provoz. Pro účely simulace stačí řídit vždy celou stanicí oddělenou autoblokem. Simulátor dostane reálnější vzhled, protože si budou moci vyzkoušet řízení a komunikaci více lidí zároveň.

### 3.1.6 Návěstní soustava

Pravidlem železničních řídicích systémů je zásada, že je jízda vždy zakázána, když není povolena. Návěstní znaky mají následující význam: červená pro STŮJ, zelená pro VOLNO, žlutá pro POMALU, modrá a bílá pro posun. Návěstidlo se dělí na horní a dolní část. Horní část ukazuje předvěstní znak, oznamuje strojvedoucímu, co má zpravidla očekávat na příštím návěstidle. To neznamená, že tomu tak skutečně bude, protože ve výjimečných situacích může dojít ke změně. Například může dojít ke zrušení cesty. V dolní polovině je zobrazen návěstní znak. Omezuje rychlost vlaku za návěstidlem. Existuje ještě několik výjimek. Návěst STŮJ nezobrazuje předvěstní znak následujícího návěstidla. Dalším problémem je, pokud je nutné vyslat vlak na trať bez zabezpečení systémem. V tom případě zodpovídá za vyslání vlaku výpravčí poté, co se přesvědčí, že nehrozí žádné nebezpečí. K tomu slouží přivolávací návěst, přerušované bílé světlo za současného rozsvícení návěsti STŮJ.

Omezení rychlosti na návěsti se spočítá jako minimum z traťové rychlosti. Zajímavým prvkem je výhybka. Záleží, jakým směrem vlak na výhybku najíždí. Pokud jede rovně, záleží jestli jede po hrotu nebo proti hrotu. Horší situace po technické stránce je pro směr proti hrotu. Je nutné zajistit, aby jazyk přiléhal ke kolejnici. Další vliv na omezení rychlosti má poloměr výhybky při odbočování. V simulaci stačí zavést přenos návěstních světél v rámci oblasti.

## 3.2 Volba prostředí

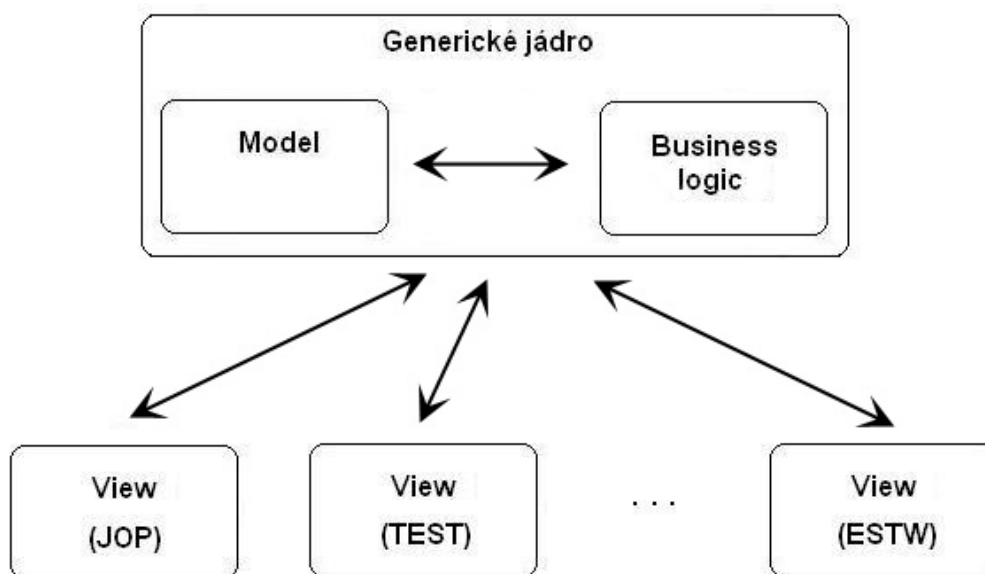
Jednoznačnou volbou pro uchování dat je formát XML (eXtensible Markup Language). Je snadno rozšiřitelný a jednoduše čitelný a přehledný. Na rozdíl od datového souboru,

kde by se s každou změnou musela měnit gramatika.

Snahou je, aby systém fungoval pokud možno nezávisle na platformě. V úvahu přichází především JAVA a Microsoft .NET FrameWork. Upřednostnil jsem platformu Microsoft .NET FrameWork ve verzi 2.0, sice ještě není plně podporována všemi systémy, ale dle mého názoru je jen otázkou budoucnosti, kdy tomu tak bude. Například projekt MONO umožňuje spouštět .NET aplikace na systémech UNIX.

### 3.3 Návrh řešení

Pro splnění podmínek zadání navrhují často používanou architekturu generické jádro komunikující s view. Generické jádro se skládá z controleru a modelu. Controller se bude starat o business logiku programu, model je pak model stanice a její struktury. View mohou být jednotlivé plug-in pro řízení. Navržen je pouze jeden vizualizační modul dle standardu JOP. Pro daný problém se generické jádro bude chovat jako celek a komunikovat s view.



Obrázek 3.1: Návrh komunikace modulů

#### 3.3.1 Generické jádro

Model stanice si můžeme představit jako neorientovaný graf. Uzly slouží jako jednotlivé prvky kolejíště, hrany udávají strukturu propojení jednotlivých prvků. Orientace hran bude nutná při vyhledávání cesty, především u prvku výhybka, aby nemohlo dojít k postavení nesmyslné cesty. Jednotlivé prvky se pak starají o svůj aktuální stav, jako jsou závěr, obsazení koleje, vlaková či posunová cesta apod.

Je nutné řídit stavění cest. Mohl by ho zabezpečit centrální zásobník cest, který by se staral o rozdělování příkazů. Kromě zadávaných cest je nutné obsluhovat i automaticky stavěné cesty.

Vlak se skládá z vozů. Pokud má na libovolném konci hnací vozidlo, může být v pohybu. Jeho rychlost je dána minimální rychlostí ze všech maximálních rychlostí všech vozů. Dále definujme zastávky. Zastávka je vždy souvislá a skládá se minimálně z jednoho uzlu grafu. Vlak obsahuje seznam časů odjezdu z jednotlivých zastávek.

### 3.3.2 Hledání cesty

Vlaková cesta se staví k následujícímu prvnímu návěstidlu. Můžeme si ji představit jako cestu v orientovaném grafu, která začíná a končí u návěstidla. Délka cesty není při vyhledávání důležitá. Orientace grafu je nutná pro odstranění cyklů v grafu. Zaručili jsme, že při stavění cesty bude vždy k dispozici acyklický orientovaný graf, není nutné používat pro vyhledávání Dijkstrův algoritmus pro vyhledání nejkratší cesty v kladně ohodnoceném grafu, protože problém se zjednoduší na jednoduché vyhledávání do šířky.

### 3.3.3 View

Jednotlivá view slouží jako ovládací plug-in jednotlivých zařízení. Definují jak vizualizaci, tak ovládání simulátoru. Po domluvě s vedoucím diplomové práce se ve svém zadání zaměřím pouze na návrh splňující podmínky JOP. Zároveň předpokládám bezporuchový systém.

## 4 Realizace

Řešení modulů bylo realizováno dle návrhu řešení. Nejdůležitější je generické jádro, logika systému dává simulátoru jeho sílu prosadit se v konkurenčním prostředí.

### 4.1 Model stanice

Struktura stanice se dá snadno popsat jako dvourozměrné pole objektů, které mohou, ale nemusí být vzájemně provázány. Jednotlivá políčka pole lze rozdělit na ta, co mají kolej a může na nich stát vlak, a na políčka, která slouží pro obsluhu, ale vlak na nich stát nemůže, a políčka, která slouží pouze pro vizualizaci. Zaměřím se pouze na popis nejdůležitějších prvků.

K uchování informací je použit XML soubor, který má výstižně určenou strukturu. Jednotlivé sekce jsou rozděleny na definici mapy, jízdního řádu a obecná nastavení. U mapy je nutné uchovat informace o jednotlivých políčkách, jako jsou jeho jednotlivé atributy a jeho umístění v dvourozměrném poli pomocí souřadnic „x“ a „y“. Výjimečná specifika jsou popsána u konkrétních případů.

K uložení celého modelu je použito výhod serializace objektů. Protože veškeré informace o sobě uchovává objekt TMapa, stačí tuto třídu nejprve serializovat a informace uložit. Instanci třídy lze jednoduše obnovit deserializací. Množství informací, které se při se-

rializaci ukládá, je zbytečně velké. Vlastní formát by znatelně snížil velikost uloženého souboru. V dnešní době, kdy i osobní počítače mají velké diskové kapacity, je to dle mého názoru zbytečné. V případě potřeby komprimace dat existuje snadná možnost, jak ušetřit kapacitu disku. Data jsou ukládána v podobě streamu do souboru. Použijeme-li místo obyčejného streamu tzv. ZIP-stream, dojde tak jednoduše ke komprimaci dat.

#### 4.1.1 Třída TPolicko

Třída TPolicko je základní třída, která definuje standardní rozhraní prvků s okolím. Třída definuje základní vlastnosti, které jsou společné všem třídám, jako je typ, tvar, stav, obsazenost, závěr apod. A stejně tak metody, nalezení nejbližších sousedů prvku, zjištění obsazenosti koleje apod.

##### 4.1.1.1 Kolejový obvod

Kolejový obvod si lze představit jako posloupnost uzlů grafu, které spolu sousedí a mají stejné číslo. Tím se rozpoznává například obsazenost kolejového úseku, neboť, je-li třeba jen jediný uzel obsazen, celý úsek musí být ve stavu obsazeno.

#### 4.1.2 Třída TNavestidlo

Návěstidlo je nejdůležitější prvek v simulaci. Rozhoduje o pohybu vlaku, staví se od něho jízdní a posunové cesty, zabezpečené i nezabezpečené. Má řadu zvláštností, které si přiblížíme.

##### 4.1.2.1 Vstupní a výstupní body

Podíváme-li se na propojená políčka jako na graf, je nutné definovat vstupní a výstupní políčka. Tato políčka slouží jako sousední body s virtuální sousední stanicí. Definují návěstidla vstupní, výstupní a vstupněvýstupní. Některá návěstidla mohou sloužit pouze pro jeden směr.

Některými vstupy do oblasti mohou vjíždět vlaky, druhými mohou vjíždět posuny. Je pravda, že lze posun vypravit i mezi dopravnami, ale tento fakt je v simulaci zanedbán. V praxi se stává velice málo, že by byl vyslán posun do mezistaničního úseku, např. při dopravní nehodě. Jinými případy jsou vlečky a vjezd do depa, který je simulován právě posunovými vstupy a výstupy.

##### 4.1.2.2 Náhodná doprava

Kromě pravidelné dopravy, která je definována v jízdním řádu, a to jak osobní, tak nákladní, je umožněno definovat náhodnou dopravu. V simulaci se dá nastavit frekvence generování náhodné dopravy. Jednotlivé vstupy mají definován koeficient atributem „mult“, tzn. jak často má být vybrán jako bod vjezdu. Čím menší je koeficient, tím méně je pravděpodobné, že bude vstup vybrán. Dále se definují typy vlaků, které mají vjíždět do obvodu. Takto definované vlaky mohou navíc mít i definovanou soupravu. Jednoznačná identifikace soupravy je dána atributem „id“. Zadá-li se stejný typ vlaku s jinou soupravou, dostaneme náhodné řazení vlaku. Posledním nastavitelným údajem je směr, kam z daného vstupu může být vlak směřován, element je nazván „to“. Samozřejmě má

také atribut „mult“, z kterého se vypočítává pravděpodobnost výběru výstupního bodu. Vstupní a výstupní body musí být jednoznačně identifikovatelné, aby bylo jasné, odkud a kam má vlak jet. Následuje ukázka definice návěstidla:

```
<policko ... typ="VSTUPVYSTUP" text="C2" mult="1" celejmeno="Chalupki" ...>
  <type id="prazdne_Falls">Vn</type>
  <type id="uhli_Falls">Pn</type>
  <to mult="1">DE1</to>
  <to mult="1">OH1</to>
</policko>
```

Takto je nadefinováno návěstidlo s jednoznačným označením C2. Frekvence četnosti „mult“ je jedna, což je malá pravděpodobnost pro výběr tohoto prvku. Návěstidlo může být využito pro vstupní i výstupní bod. Mohou sem vjíždět a vyjíždět dva typy vlaků — vyrovnávkový vlak (Vn) a průběžný nákladní vlak (Pn). Každý vlak má přiřazenou soupravu atributem „id“ (PRAZDNE\_FALLS a UHLI\_FALLS). V případě, že je návěstidlo vybráno při náhodném výběru vstupního bodu, uplatní se elementy „to“. V tomto příkladu dojde se stejnou pravděpodobností k výběru výstupního bodu DE1 nebo OH1. Stejně jako je v ukázce nadefinováno návěstidlo C2, musí existovat v tom samém XML souboru definice pro návěstidla DE1 a OH1.

#### 4.1.2.3 Návěstní soustava

U návěstidla se udržuje informace o aktuálním stavu. Základní stav je návěst STŮJ. Existují ale další možnosti: od návěstidla je postavená posunová cesta, vlaková cesta či nezabezpečená cesta nebo jen rozsvícená přivolávací návěst. Pouze při postavené vlakové cestě je nutné zjišťovat, jaká návěst je zobrazená na následujícím návěstidle a zjišťovat minimální omezení rychlosti kolejí k následujícímu návěstidlu. Tyto informace jsou dostačující k tomu, aby je bylo možné předat controleru ke zpracování a view k zobrazení.

#### 4.1.2.4 Autoblok

Simulace návěstidel na autobloku je vyřešena jako samostatný problém. Aby nemohlo dojít k tomu, že by se na koleji s návěstidlem vyskytly dva vlaky najednou, je prvku návěstidlo vždy definována délka nula. Autoblok je simulován jako obyčejná kolej s označením autobloku, čehož se využívá při stavění automatických cest. Na takto označené koleji je zakázáno rozpojování vlaků.

#### 4.1.3 Třída TSouhlas

Souhlas je prvek bez koleje a slouží k určení směru (vjezd či odjezd), kterým se mohou vlaky pohybovat po mezistaničním úseku. Prvky je nutné mít vždy dva a jsou přímo napojeny na návěstidla na koncích úseku. Tak se může zajistit kontrola, zda není úsek obsazen, jelikož jen v té době je možné měnit stav udělení souhlasu. Pro úseky v rámci jedné řízené oblasti je souhlas udělován automaticky dle stavění jízdních cest. Pokud jsme na rozhraní se sousední stanicí, musí být oblast označena jako hraniční a

žádost pro odjezd je udělována automaticky po náhodné době. Stejně jako ve skutečnosti ji uděluje dispečer sousední oblasti. Bezpečnost se simuluje tak, že souhlas může být přiřazen maximálně jednomu návěstidlu. To znamená, že souhlas nemusí být vůbec udělen. Tato situace nastane především při inicializaci systému.

#### 4.1.4 Třída TZasobnik

Třída TZasobnik zpracovává příkazy pro stavění jízdních cest. Zásobník má tři základní módy: přednostní volba, volba do zásobníku a editace zásobníku. Každá oblast může mít vlastní zásobník, i jedna stanice může být rozdělena do více oblastí. Protože je zásobník využíván i pro stavění cest, které nezadá uživatel, musí mít stanice vždy zásobník pro oblast nula. Povel se ukládá jako struktura odkud (počáteční návěstidlo), kam (koncová kolej) a o jaký typ cesty se jedná.

- Přednostní volba — PV

V tomto stavu je zásobník vyřazen z činnosti. Co se týče uživatelských příkazů, příkazy pro stavění cesty jsou zpracovávány ihned a v případě nemožnosti provedení volby, je příkaz zahozen. Programově je však zásobník neustále v činnosti, protože umožňuje stavění automatických cest. Automaticky stavěné cesty jsou přidávány do zásobníku během simulace, např. při vjíždění vlaku z virtuální oblasti do autobloku.

- Volba do zásobníku – VZ

Zásobník je založen na frontě FIFO „first in first out“, tj. povel který je vložen do fronty jako první, z ní i jako první vychází. Dokud se zpracovává položka na samém vrcholu, není možné pokračovat ve vykonávání povelů dalších položek. Každý simulační cyklus je vyvolána metoda, která se pokusí postavit cesty v zásobníku. V zásobníku jsou cesty zadané uživatelem nebo automaticky. Metoda zkouší postavit všechny automatické cesty a z uživatelem navolených pouze první v pořadí. Před pokusem o postavení nezabezpečené cesty je navíc nejprve nutno volbu potvrdit.

- Editace zásobníku – EZ

Editace zásobníku označuje stav, při kterém je možné měnit pořadí dispečerem zadaných povelů. Po celou dobu je blokováno stavění cest. Položku můžeme smazat. Není přípustné zadávat nové povely a celková obsluha stanice musí být blokována. O tyto kontroly se však již stará modul view.

#### 4.1.5 Třída TPrejezd

Třída TPrejezd zajišťuje správný stav přejezdového zařízení. Instance musí kontrolovat blížící se vlaky. Třída nabývá tří základních programových stavů: **klidový stav** je stav, při kterém závory jsou nahoře a bliká bílé světlo. Jakmile je vlak ve vzdálenosti, která se v simulátoru vypočítává od rychlosti koleje přejezdu, uplatní se stav předzvánění. **Předzváněcí doba** je čas, kdy již blikají červená světla. Je možné ji také nastavit. Nastavením předcházejících dvou hodnot lze docílit, aby byly závory dole bezpečně



dlouho před vjetím vlaku na přejezd. Posledním stavem je **uzavření závor** pro bezpečné projetí vlaku. Ihned po projetí vlaku a uvolnění klopného obvodu přejezdu dochází k otevření přejezdu. Vlak je sice ještě v blízkosti přejezdu, ale jede směrem od závor, proto se předzváněcí doba neuplatní. Kdyby však jely dva vlaky bezprostředně za sebou, podmínka uzavření závor je stále splněna a závory zůstávají uzavřeny. Speciální případ nastává tehdy, kdy je zaveden tzv. dopravní klid. V takovém případě se předchozí algoritmus neuplatní, přejezdové zařízení má symbol stále jako zavřené.

## 4.2 Komunikace

Komunikace mezi více počítači je realizována pomocí TCP/IP protokolu. Jeden počítač je určen uživatelem za server a stará se o komunikaci se všemi klienty a přenos informací mezi klienty. K takové komunikaci je využita třída „server“. Celý model stanice je tím rozdělen do oblastí, které mohou být samostatně řízené. Odděleny musí být autoblokem z důvodu předávání informací o vlaku na přelomu oblastí.

Přenášená data si můžeme představit jako serializované objekty v XML podobě. Pro snížení toku dat se nepřenášejí všechny informace, ale jen nutné stavy objektů. Dalšího snížení toku dat se docílilo uchováním informace o změnách jednotlivých prvků. Jsou přenášeny jen ty objekty, které změnil od poslední komunikace stav. Na opačné straně jsou změněny patřičné stavy, nikdy není přenášet celý objekt. Není ani nutné přenášet celé jízdní cesty nebo vlaky, protože simulace dané oblasti probíhá vždy jen na počítači, který daný objekt řídí. Ostatní vidí jen to, co se v jiných oblastech děje, ale nemohou zasahovat do řízení.

Přínosem protokolu TCP/IP je jistota, že se data neztratí, dorazí ve správném pořadí a nejsou duplikována. Není však zaručeno, že dojdou veškerá data najednou. Proto je na každé straně mechanismus opravy doručených dat, který zaručí, že jsou k dispozici veškerá data.

### 4.2.1 Strana serveru

Při aktivaci této třídy dojde k poslouchání na určitém komunikačním portu. Když se klient připojí, dojde k vytvoření instance třídy typu „ObsluhaKlienta“ (dále jen OK), která uchovává informaci o otevřeném kanálu a komunikuje s klientem. Kolik je připojených klientů, tolik instancí této třídy existuje. Každý klient musí žádat o přiřazení k určité oblasti. Oblast musí být jednoznačná a po přidělení ji nesmí server dál umožňovat obsluhovat.

Třída OK má dvě fronty: vstupní a výstupní. Do vstupní fronty ukládá vše, co přichází ze strany klienta. Fronta je nutná z důvodu, že nemusí dorazit všechna data najednou. Ve výstupní frontě jsou data, která je nutné odeslat klientovi. Neustále je opakován cyklus, který zjišťuje, zda nedorazila nějaká data. Když ano, jsou načtena a uložena do příchozí fronty. Dále jsou zkontrolována data k odeslání, a pokud nějaká data jsou, mohou být odeslána. Poslední částí cyklu je krátký časový interval uspaní vlákna. Je neúčelné, aby se stále kontrolovala data, protože simulační cyklus je po jedné sekundě.

Dojde-li ke změně stavu některého prvku na serveru, dojde k zařazení této informace

do všech front OK. Jednotlivé OK se postarají o zaslání dat svému klientovi. K jiné situaci dochází, přijdou-li data od klienta. Přijatá data mohou být určena buď jen serveru, serveru a ostatním klientům nebo klientovi. První situace je nejjednodušší, server data zpracuje, a komunikace je tak ukončena. Při druhé možnosti dojde k zpracování a k zaslání té samé informace všem klientům kromě toho, od kterého informace přišla. U třetí možnosti jsou data pouze přeposlána. S kým daná komunikace probíhá, je dáno příslušným číslem oblasti obsažené v datech.

#### 4.2.2 Strana klienta

Pro připojení se k severu je nutné znát jeho IP adresu a port. Teprve tehdy může dojít k propojení a ke komunikaci. Klient nejprve musí zažádat o přidělení jedné určité oblasti. Pokud mu je přidělena, může řídit pouze přidělenou oblast. Komunikace probíhá stejně jako u ObsluhyKlienta, tj. existují dvě fronty pro příjem a odesílání. Všechna data jsou zasílána přímo na server a ten je přerozděluje. Klient nepotřebuje dopředu znát, komu jsou data předána. Poté klient přijímá změny a požadavky ze serveru.

### 4.3 Model vlaku

Problém s vlaky se dá rozdělit do více podproblémů, čímž se nám celý problém zjednoduší. Z analýzy vyplývá, že je třeba definovat řazení vlaku, zastávky, datumové omezení. V simulaci navíc ještě přibude cesta vlaku, která určuje, kde se právě vlak nachází.

#### 4.3.1 Popis struktury

Vlak jako celek poskytuje obecné vlastnosti, které jsou zpravidla odvozené od struktury vozů. Celá souprava se může pohybovat v případě, že je alespoň na jednom konci činné hnací vozidlo. Navíc ale musí být splněna podmínka napájení. Elektrická lokomotiva nemůže jet jako činné hnací vozidlo po koleji, kde není natažené trolejové vedení.

Jednotlivé vozy jsou nadefinovány zvlášť. Souprava se skládá již jen z odkazů na určité typy vozů. Každý vůz má definován cílový bod, proto je nutné některé soupravy rozřazovat. Pro pestřejší provoz je možné využívat různé mutace stejného obrázku daného vozu. Tím je pokryta prakticky veškerá škála možností a v praxi se ukázalo, že je postačující.

Zastávky jsou určeny dvojicí jméno a číslo koleje. Vlak musí mít u jednotlivých zastávek určen čas odjezdu z dané stanice. Počet stanic není nijak omezen. Vlak má pouze ukazatel na vrchol zásobníku, který se po zastavení ve stanici posouvá o jednu stanicí dále. Za zmínku stojí atribut, zda má vlak čekat na výstup cestujících. Tak je možné donutit vlak projet určitou stanicí, ale není v ní nutné zastavit.

Vlak má také přiřazenu nějakou cestu. Pokud tomu tak není, vlak není v oblasti a čeká na vjezd. Na mezistaničních úsecích jsou proto generovány cesty automaticky, i když je uživatel nevidí. Automaticky stavěné cesty jsou přiřazeny vlaku při vjetí do řízené oblasti.

### 4.3.2 Jízdní řád

Aby bylo možné vytvářet reálné prostředí, byla přidána možnost vytvářet skutečný jízdní řád. Již byla popsána možnost řazení a zastavování vlaku. Jízdní řád navíc umožňuje definování datumového omezení či návaznost vlaků. Vlak může do oblasti buď vjíždět (z depa nebo sousední virtuální oblasti), nebo vznikat z jiného vlaku (tzv. „přečíslování vlaku“).

Existuje seznam všech vlaků, jejichž čísla se mohou i opakovat. K opakování čísla může dojít v případě, že určitý vlak má v různé dny jiné řazení. Není přípustné, aby ve stejný den jely dva vlaky se stejným číslem. Musí být vždy určeno, odkud a kam má vlak jet. U končících vlaků se definuje, na který má být přečíslován. Vznikající vlaky mohou být nabízeny jako potenciální kandidáti pro přečíslování jiného vlaku. Od kandidáta jsou zděděny nové zastávky a časy odjezdů. Není striktně řečeno, že je nutné zachovávat řazení vozů; tato informace je však k dispozici.

Definice datumového omezení umožňuje stejné možnosti jako je tomu v jízdním řádu, především omezení na dny v týdnu, určité konkrétní datum, ale i období. Lze definovat jak zakázané, tak povolené omezení.

Celou problematiku si popíšeme blíže. Základní strukturu si můžeme představit následovně: v uzlu „dates“ jsou nadefinována různá datumová omezení, která je pak následovně možné přiřadit jednotlivým vlakům v jízdním řádu. Jako identifikátor slouží atribut „name“. V ukázkovém případě mají hodnoty 11 a 12.

```
<dates>
  <date name="11">
    <run at="X"/>
  </date>
  <date name="12">
    <run at="1"/>
    <run at="2"/>
    <run at="5"/>
    <stop at="12.4."/>
    <stop at="5.7." till="6.8."/>
    <stop at="28.9."/>
    <stop at="28.10."/>
  </date>
</dates>
```

V prvním případě by vlak jel pouze pondělí až pátek (X značí všední dny). Element „run“ udává přípustné datумы, kdy je možné vlak generovat. Státní svátky nejsou v programu přímo zakompilovány. Jízdu v pracovní dny lze však simulovat, přidají-li se elementy s omezením „stop“. Omezení jsou vidět v druhém případě, kdy jsou vyloučeny dny 12.4., 28.9. a 28.10., ale dokonce i období od 5.7. do 6.8. Existuje tedy možnost definovat konkrétní den nebo období vymezením počátečního a konečného dne platnosti.

Třída Date uchovává údaje o jednotlivých omezeních. K tomuto účelu třída využívá

dynamická pole, která uchovávají jednotlivé informace zvlášť. K dispozici jsou seznamy opakujících se dnů, jednotlivých dnů či období. Tuto variantu jsem zvolil pro zpřehlednění celého kódu. Je pak na první pohled jasné, která omezení jsou prioritní.

Nejdůležitější metodou třídy `Date` je metoda pro určení, zda může v konkrétní simulační den vlak jet. Nejprve jsou kontrolovány zakázané podmínky, pak teprve podmínky dovolující. Povolující datumové omezení tak nemá prioritu nad zakázaným omezením.

Při vytváření jízdního řádu se načtou všechny vlaky. Každý vlak může mít přiřazeno datumové omezení. V případě, že nemá vlak přiřazeno žádné omezení, vlak jede vždy a nedochází k žádné kontrole, v opačném případě dojde k vyvolání dříve popsané metody. Vlaky jsou nadefinovány v rámci jednoho simulačního dne. Jsou-li proto splněny podmínky pro jízdu vlaku, je vlaku nastaven atribut pro generování vlaku. O půlnoci pak dochází k přepočítávání celého seznamu znovu.

K zařazení vlaku do simulace dochází na vyžádání modulu. Vlak je vytvářen v určitém předstihu před pravidelným vložením vlaku, a to vždy o minutu a nějaký náhodný čas. Důvodem je udělení souhlasu vlaku pro vjezd do řízené oblasti. Nejprve musí být přijat vlak, pak musí být udělen souhlas sousední virtuální oblasti. Způsob se však může lišit modul od modulu.

Datová struktura jízdního řádu umožňuje definovat více jízdních řádů, například pro různé roky nebo různé fikce. Simulace však musí mít vždy určeno, s kterým konkrétním GVD má pracovat. Pokud se tak nestane, bere v potaz pouze první definovaný jízdní řád.

## 4.4 Jízdní cesty

Již máme nadefinovanou strukturu stanice a vlaku. Nyní si popíšeme, jak spolu vzájemně spolupracují a jak je řešen pohyb vlaků po kolejích. Vlaky se pohybují po jízdních cestách, proto existuje seznam těchto cest, které jsou přidělovány jednotlivým vlakům. Vlak nemůže jet, aniž by měl přidělenou cestu. Speciální případ na přivolávací návěst je řešen zvlášť.

### 4.4.1 Stavba cesty

Ze zadání plyne, že existuje několik typů jízdních cest. Dle typu jízdní cesty je určeno, jak se má cesta stavět, jízdní cestu například není možno postavit na obsazenou kolej, ale posunovou či nezabezpečenou ano. Vyhledávání cesty je postaveno na prohledávání do šířky. Základní algoritmus je nutno jen nepatrně pozměnit.

Především je nutné zorientovat graf tak, aby nemohlo dojít k postavení cesty, která by neměla smysl. K nesmyslné cestě může dojít u výměn, pokud se na ni dostaneme ze směru „po hrotu“, tj. pokud se v hledání dostaneme na prvek ze směru rovně, nesmí se dále pokračovat směrem odbočujícím. Vlak se nemůže otočit na výhybce o úhel větší jak o 180 stupňů. Je tedy nutné zkontrolovat směr, ze kterého se k prvku dorazilo. V případě, že jsme „proti hrotu“, můžeme dále pokračovat pouze směrem „po hrotu“. V opačném případě záleží na tom, zda je povoleno měnit polohu výměny. To je možné

rozlišit vlastností „automaticky“. Když můžeme polohu měnit, můžeme pak pokračovat dvěma směry, jinak pouze směrem, kterým je výměna aktuálně přehozená. Stav, kdy není měněn směr vlaku, je označován jako „rovně“, v opačném případě jako „přehozeně“.

Prvotní nastavení uzlů. Uzly mohou nabývat následujících stavů FRESH, OPEN a CLOSED. FRESH značí uzel, který lze využít pro hledání cesty a který ještě nebyl navštíven. Ve stavu OPEN je prvek, který byl zařazen na konec vyhledávací fronty. Tento prvek již nemůže být použit při hledání cesty. Stav CLOSED označuje buď již zpracovaný uzel, nebo uzel, který je nepoužitelný. Tato úprava původního algoritmu je důvodem prvotní inicializace. Na počátku nejsou všechny uzly kromě počátečního označeny jako FRESH, ale záleží na typu cesty. V případě, že se jedná o vlakovou cestu, jsou nastaveny všechny obsazené koleje, koleje v nerozlišeném závěru a koleje s jízdní či posunovou cestou, na uzavřené. Není problém poslat posun na obsazenou kolej, cesty se ale nesmí křížit. Cesta je proto postavena jen na počátek obsazené koleje a jízda k návěstidlu probíhá dle algoritmu popsaného u jízdy vlaku. Poslední stavěnou cestou je nezabezpečená cesta, která pouze provádí závěr výměn. Cestu lze postavit i na obsazenou kolej. V tomto případě jsou prvky se staniční kolejí, tj. kolejí s číslem větším nebo rovným nule, označeny jako otevřené.

Protože prvek návěstidlo má konkrétní orientaci, je zřejmé, kterým směrem je nutno začít cestu hledat. Pak již je celý graf orientován. Z nastavení uzlů vyplývá, že nemůže dojít k uvážnutí algoritmu v cyklu. Proces hledání je ukončen ve chvíli, je-li nalezen koncový bod anebo není-li již ve frontě žádný uzel ke zpracování. Vlaková i posunová cesta, pokud je kolej prázdná, nekončí uprostřed koleje, ale je vždy vyhledáno návěstidlo na konci koleje. U vlakové cesty jsou ignorována seřadovací návěstidla. U posunové cesty na obsazenou kolej se bere v úvahu jako cílový bod první obsazený prvek staniční koleje. Tímto způsobem se zablokuje celý kolejový obvod pro ostatní vlaky.

Cesta se musí skládat vždy minimálně z posloupnosti dvou disjunktních uzlů grafu. Poslední uzel je využit jen jako ukazatel, slouží jako senzor (co je očekáváno). Senzoru je využíváno při pohybu vlaku. Jestliže je cesta dle výše popsaného algoritmu nalezena, je uložena do seznamu všech cest. V případě, že nalezena není, záleží na daném modulu, stavech zásobníku apod.

Kdy dochází ke stavění cesty, záleží na stavu zásobníku dané oblasti. Buď je vyvolána metoda stavění cesty přímo, nebo je stavba cesty volána z jednotlivých zásobníků. Vždy dochází k pokusu postavit všechny automaticky generované cesty. Ručně zadaná cesta má nižší prioritu a je postavena jen první (ta na vrcholu zásobníku). Automatické cesty jsou cesty například na vjezdu do mezistaničního úseku na hraniční oblasti. Jedná se o simulaci stavění cesty virtuálním výpravčím, vstupní návěstidlo je označeno příznakem, že má generovat automaticky cestu. Ta je vygenerována jen jednou, jakmile je vlaku udělen souhlas pro vjezd. Dalším případem je cesta v režimu „AB“, kdy po projetí vlaku touto cestou musí být cesta znovu postavena.

Výjimkou při stavění cesty jsou nezabezpečené cesty na přivolávací návěst. Nezabezpečená cesta je v simulaci chápána trochu jinak. Vlak nemá nezabezpečenou cestu vůbec přiřazenou. Důvodem nepřičtení cesty může být situace, že nezabezpečená cesta může

být postavena přes jinou cestu a může s ní být manipulováno i po postavení. Nezabezpečená cesta totiž pouze snižuje riziko nebezpečí, a to tak, že správně přehodí výměny a upevní je nouzovým závěrem. Jedná se o rizikovou funkci právě proto, že tato funkce umožňuje postavit nezabezpečenou cestu do jízdni cesty jiného vlaku, ale také zrušit nouzový závěr na výměně a přehodit výhybku. Rušení nouzové cesty je vyřešeno tak, že je posléze jako ostatní cesty přidána do seznamu jízdni cesty, odkud může být zrušena.

Rušení cesty může být ruční nebo automatické a je značně specifické dle jednotlivých modulů. Cesta je vždy vyjmuta ze seznamu cest a u jednotlivých uzlů je nastaveno uvolnění prvku, a tak může být tento prvek použit při stavbě nové cesty. V našem případě je umožněno cestu zrušit ručně nebo automaticky. Ručně může být zrušena pouze cesta, která není přiřazena žádnému vlaku. Nezabezpečené cesty jsou odstraněny automaticky, jakmile jsou všechny nouzové závěry po celé její délce ručně zrušeny. Nezabezpečená cesta může být však zrušena i ručně. Různé časové závislosti a intervaly pro rušení jsou splněny dle normy JOP.

## 4.5 Pohyb vlaku

Realizaci pohybu vlaku můžeme rozdělit do dvou podproblémů: pohyb po jízdni cestách a pohyb bez jízdni cest. Jak již bylo řečeno, základní teze byla, že vlaky jezdí vždy po jízdni cestách, protože cesty se nemohou za normálních okolností křížit. Jízdni cesty jsou tedy disjunktní. V simulaci je proto stavěna i pro pohyb bez jízdni cest skrytá cesta, tak jsou splněny předchozí popisy algoritmů, především disjunktnost uzlů. Pokud je vlak v simulaci, musí mít vždy přiřazenou cestu. Jízdni cesta může být buď vlaku přiřazená, nebo nepřiřazená, nemůže však být přiřazená více vlakům najednou. Posunovým vlakům jsou přiřazovány posunové cesty, naopak očíslovaným vlakům vlakové cesty. Po opuštění posledního prvku celého KO je cesta automaticky rušena.

Nejprve je nutné vlaku přiřadit cestu. Vlak má pomocí vstupního a výstupního bodu definováno „odkud“ a „kam“ má jet. Nyní si ukážeme, že se nedopustíme omezení obecnosti, pokud popíšeme možnost, že jsou vstupní a výstupní body vždy definovány. Vstupní bod „odkud“ je přiřazen pouze u příjíždějícího vlaku z některé z virtuálních oblastí. Vlak dále může vzniknout z jiného vlaku nebo být přečíslován. Vznikající vlak znamená, že se predefinují již existujícímu vlaku zastávky, cílový uzel a číslo. Existující vlak však buď mohl přijet z některého ze vstupních bodů, vzniknout nebo být přečíslován. Při pouhém přečíslování vlaku se mění pouze číslo, což v našem případě nemá vliv na vstupní bod. Po konečném kroku se vždy dopravujeme k výsledku, že vstupní bod byl definován. Stejným způsobem bychom si mohli ověřit, že i výstupní bod bude nakonec jednou definován. Můžeme uvažovat o možnost, kdy přijede na stanici hnací vozidlo, které tam již zůstane. Takový vlak však již má přiřazenou cestu.

Máme tedy vznikající vlak, který již má udělen souhlas. Ukazatelem cesty vlaku je nyní vstupní bod. Existující, ještě nepřiřazená cesta z daného bodu, je přidělena prvnímu vlaku z daného místa, který má udělen souhlas. Přiřazením cesty prvnímu vlaku se zajišťuje, že dva vlaky nemohou mít přidělenou stejnou cestu. Po tomto prvotním přiřazení cesty se již nemůže stát, že by se vlak pohyboval v simulaci a neměl cestu definovanou.

Nyní si charakterizujeme pohyb po jízdni cestě. Popis platí jak pro vlakovou, tak analogicky pro posunovou cestu. V každém cyklu časovače „view“ je vyvolán jeden krok simulace. Jednotlivé prvky cesty mají definovanou délku. Délka vlaku se rozkládá mezi podposloupnost uzlů přiřazené cesty. Jeden takt cyklu je považován za jednu reálnou sekundu. Každý uzel má definováno, kolik je již z délky koleje obsazeno. U nové cesty je tedy tato hodnota nulová. V každém cyklu se dle aktuální rychlosti spočítá, kolik metrů má vlak ujet. Z této vzdálenosti je postupně plněna hodnota obsazenosti koleje jednotlivých uzlů cesty, dokud není nově ujetá vzdálenost nulová. Tímto způsobem se nám postupně zaplní celá cesta. Nyní je nutné uvolňovat od počátku uzly tak, aby součet obsazenosti koleje všech prvků (kromě posledního, to je jen senzor) cesty nebyl delší, než je délka vlaku. V případě, že by při odebrání uzlu mělo dojít k tomu, že by již tato délka byla menší než délka soupravy, uzel je ponechán. Dle mého názoru toto zanedbání však ničemu nevádí, v praxi to jen znamená, že na jednom poli nemohou být dva vlaky.

Jakmile vlak dojede na konec cesty, na kterém je návěstidlo, uplatní se senzor. Konec cesty je rozpoznán, když hodnoty „délka“ a „obsazenost“ u předposledního políčka cesty jsou stejné. Aktuálně ujetá vzdálenost tak zůstane nerozprostřena mezi nové uzly. Nyní je hledána cesta v seznamu již postavených cest, která ještě není přiřazena žádnému vlaku a její počátek je na samém konci aktuální cesty vlaku. Z této situace vyplývá funkce senzoru, protože cesty musí být disjunktní a také cesty nesmí být postaveny z obsazeného pole. Senzor tedy slouží jako spojka při navazování dvou cest. Při nalezení volné cesty jsou jednotlivé uzly nové cesty (kromě prvního) přidány k aktuální cestě. Dojde tedy ke sjednocení dvou cest a v seznamu cest se ze dvou stane jen jedna. Od této chvíle není již možné tuto cestu zrušit, protože je přiřazena vlaku. Novým senzorem se stal opět poslední prvek a pohyb pokračuje dle stejného algoritmu.

Vlakové cesty jsou vždy stavěny od návěstidla. Potíže začínají, když na konci cesty není návěstidlo. K tomu může dojít z různých důvodů: vlak se pohybuje po autobloku na mezistaničním úseku, vlak je otočen na staniční koleji, při manipulaci s vlakem a při jízdě po nezabezpečené cestě. Jak je vidět, škála případů, kdy senzor neukazuje na návěstidlo, je široká. Proto jsou u jednotlivých kolejových prvků definovány dva základní směry: „tam“ a „zpět“. Jsou to ukazatelé na okolní prvky, kudy se může vlak pohybovat. Nastavení těchto ukazatelů je nutné především u výměn, kde jsou přípustné tři směry. Opět se uplatní senzor, který zjišťuje, zda políčko, na které ukazuje, může využít pro stavění cesty, tj. zda-li je volné, není na něm žádná cesta apod. V případě, že je volné toto políčko, může být využito pro další pohyb vlaku. Toto políčko je poslední, a proto je nutné přidat k cestě nový senzor. Použijeme proto definované ukazatele „tam“ a „zpět“. Jeden z nich by měl ukazovat na prvek, na kterém stojí vlak, a druhý je využit jako nový senzor (přidáním k aktuální cestě na konec). Může však nastat i možnost, že tomu tak není. Např. jedeme-li u výhybky „po hrotu“ a zároveň ze směru, na který není výměna přehozena. Protože existují i výhybky, kde je tato možnost povolena, simulátor tuto chybu ignoruje a zvolí správný směr pro pokračování. V praxi se této vlastnosti výhybky využívá při křížení vlaků bez nutnosti přehazování výměny.

### 4.5.1 Zastávky

Zastávka je řešena jako souvislý úsek prvků koleje se stejným jménem. Jakmile vlak po cestě dorazí na první prvek se stejnou kolejí, aktivuje se do režimu zastavení. Opět je využít poslední prvek cesty. Když se senzor dostane na první políčko, které již není zastávkou, vlak je zastaven a čeká do svého odjezdu. V případě, že se čeká na výstup a nástup cestujících, musí být zaručeno, že vlak stojí v zastávce minimálně po tuto dobu. Proto je za čas odjezdu považováno maximum z času odjezdu a součtu času zastavení s dobou čekání. Nezáleží na číslu koleje stanice, protože to je pouze doporučené, nikoliv povinné využít.

### 4.5.2 Otočení vlaku

Při otočení vlaku mohou nastat dvě varianty. První je snadná, vlak mění pouze směr. To znamená, že řazení vlaku je zachováno a dochází k sunutí vlaku. V té době je rychlost vlaku omezena na 30 km/h. Při této změně je jízdní cesta pouze zkrácena (o jedno pole) a otočena. Za nový konec musí být opět přidán senzor. Kdyby tomu tak nebylo, senzor by ukazoval na první prvek původní cesty. Ten by byl ale určitě obsazen a vlak by se po otočení nikdy nerozjel. Druhá možnost je složitější a skládá se z několika úkonů. Jedná se o tzv. přepřah. Nejprve je nutné odpojit hnací vozidlo. Obecně můžeme odpojit od vlaku libovolný počet vozů. Při tomto úkonu je zkopírován vlak, kterému jsou následovně upraveny vlastnosti. Především je to řazení vozu, kde průnik řazení vozů nového a původního vlaku musí být prázdný a sjednocení musí dát původní soupravu, přičemž nový vlak musí mít odpojené vozy. Podobně je nutné zajistit, aby se zkopírované cesty nekřížily. Algoritmus je následující: nově vzniklému posunovému vlaku je krácena cesta od začátku tak dlouho, dokud délka cesty neodpovídá délce hnacího vozidla. Tato cesta musí mít vždy nejméně dva prvky, což je zaručeno. U původního vlaku je postupováno obdobně, přičemž poslední prvek cesty ukazuje na počátek nové cesty. Pro nový vlak musí být stavěny postupně posunové cesty tak, aby se dostal posun na druhou stranu prvotního vlaku. Po zastavení vlaku je možné oba vlaky spojit. Zde zaniká posunový vlak a zůstává pouze primární vlak, ke kterému jsou přidány vozy z posunové soupravy. Zbývá již jen sjednotit cestu. Je nutné si uvědomit, že při sjednocování cesty i vozů mohou být soupravy k sobě různě natočeny. Tyto čtyři možnosti jsou správně ošetřeny.

### 4.5.3 Přenos čísel vlaku

Pokud je vlak již na mapě, musí mít přiřazenou jízdní cestu. Podíváme-li se na stejnou věc z pohledu cest, cesta může mít přiřazen vlak. Pro přehlednost je u JOP systému využíváno zobrazení čísla vlaku na určeném místě. U již přiřazených cest vlaku není problém číslo vlaku zjistit, horší situace je v případě navazujících cest. Je zde však využito podobného algoritmu jako při zjišťování zobrazené návěsti, jen opačně. Jakmile má cesta přiřazen vlak a na konci této cesty existuje další cesta, je toto přiřazení převzato. Neznamená to však, že tato cesta musí být již přiřazena některému z vlaků. Informace o vlaku je pouze jako očekávaná. Nikdy nemůže dojít k tomu, že by jedné cestě bylo takto přiřazeno vícero vlaků právě kvůli tomu, že cesty nemohou být kolizní.



#### 4.5.4 Opuštění modelu stanice vlakem

Posledním případem manipulace s vlakem je jeho opuštění stanice. V simulaci je to vyřešeno pomocí výstupních bodů. Výstupní body jsou vždy návěstidla speciálně označena jako „výstupní“. Jakmile vlak dorazí na konec jeho cesty, kontroluje se, zda nedojel na výstupní bod. Pokud ano, je krácena jeho délka o vzdálenost, o kterou oblast přejel. Tak dochází ke zkrácení cesty. Jakmile je délka vlaku nulová, může být vlak zrušen.

### 4.6 Modul JOP

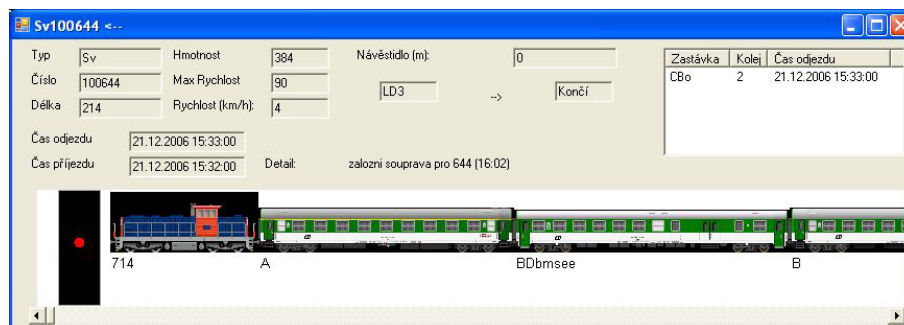
Modul JOP je implementací řízení a zobrazení dle předpisů JOP [4]. Jednotlivé procesy a ovládání jsou dány právě tímto standardem. Výjimky jsou popsány v uživatelské příručce v programu. Realizace modulu je provedena jako samostatná knihovna, která se zavádí do programu.

#### 4.6.1 Simulační cyklus

Ve smyčce běží dva základní cykly: reálný čas, který se stará o správné zobrazení světelných návěstidel či zapnutí a vypnutí zvuku a simulační čas, který může být měněn, především zrychlován. Zajímavější je právě ten druhý, který obstarává jeden simulační cyklus provozu. Pokud je vyvolán jednou za sekundu, je rychlost simulace reálná. Vždy jsou provedeny jednotlivé metody generického jádra. Simulační cyklus obstarává vznik a pohyb vlaků, stavění a rušení cest, přiřazení cest k vlakům a nakonec předání informací se serverem či klientem.

#### 4.6.2 Manipulace s vlaky

Na ovládání této manipulace neexistuje žádný předpis. V simulaci jsou proto přidány do menu vlaku možnosti pro připojení a odpojení vlaku. Navíc pro vizualizaci je využito speciální okno, které poskytuje důležité informace pro uživatele. Najdeme zde všechny vlakové zastávky, údaje o soupravě, řazení, očekávaná návěst apod. Výsledek vidíme na obrázku níže:



Obrázek 4.1: Informace o vlaku v simulaci

## 5 Testování

Aby bylo možné celý systém otestovat, bylo nutné vytvořit testovací stanice. Testování je časově velmi náročný problém, stejně je náročná tvorba dokonalých stanic. Byly využity nabídky zájemců o simulátor, aby se spoluzúčastnili projektu s tím, že vytvoří editory pro tvorbu stanic. Vznikl tak projekt Builder a Stag. Dalším krokem k otestování programu byla především snaha rozšířit simulátor mezi cílovou skupinu. Zde byla využita spolupráce jak výpravčích, tak i zájemců o simulátor.

### 5.1 Builder

Builder znamená program umožňující výrobu a editaci modelu stanice. Nejprve vznikl velice primitivní nástroj využívající knihoven generického jádra pro zobrazení stanice. Po předání projektu Milanu Záhovi došlo ke zdokonalení editoru a rozšíření jeho možností. V editoru bylo vylepšeno uživatelské rozhraní, byly přidány různé pomůcky pro zjednodušení práce a získání přehlednosti u stanice. Přidáním filtrů (pro zobrazení kolejí) bylo umožněno například rozlišení řízených oblastí, úrovní detailů zobrazitelných textů apod. Novinkou je také kopírování vybraných úseků stanice nebo nastavování vlastností více prvkům najednou. Mezi výhody editoru patří nastavení písma a výběr stylů písma. Upravená stanice je uložena do souboru ve formátu XML, kdy je využito funkce generického jádra.

### 5.2 Stag

Stag je editor umožňující tvorbu jízdního řádu, řazení vlakových souprav a nastavení náhodné dopravy. Autorem editoru je Ivo Strašil, který spolupracuje na projektu od samého začátku. Program načte soubor vygenerovaný editorem stanic. Je možné přidat jména vstupních a výstupních bodů, názvy jednotlivých zastávek, připojit datumové poznámky a nastavit náhodnou vlakovou dopravu. Kromě ručního vytváření jednotlivých vlaků program umožňuje i import z velmi rozšířeného programu jízdních řádů IDOS. Lze dokonce i vygenerovat nákrešné jízdní řády, jako to mají skuteční dispečeri. Tak se simulátor stává o něco více reálnější.

### 5.3 Stanice

Vyrobít kvalitní stanice bylo dalším nesnadným úkolem. Reálnosti stanic bylo dosaženo výbornou spoluprací zaměstnanců ČD a zájemců o program. Testy byly prováděny především na modelech stanic Bohumín a České Budějovice.

### 5.4 Simulace

Během testování provozu byly zkušeny základní dopravní situace. Projíždějící spoje bez zastavování, se zastávkami, s přepřahem stejné lokomotivy, přípřež apod. Provoz byl testován v různé dny a pro různé jízdní řády. Byla ověřena funkčnost zásobníků v jeho třech režimech, i když bylo vícero zásobníku v modelu. Testována byla i síťová kooperace, kde se ukázalo, že tok dat není příliš vysoký.

## 5.5 Mantis

Největším přínosem pro otestování programu bylo doporučení od vedoucího diplomové práce zpřístupnit testování všem zájemcům na internetu. Byl využit systém Mantis [7], který umožňuje správu problémů a nápadů k simulátoru. Jednoduché nastavení procesů od potvrzení přes schválení požadavku po uzavření celého případu dává možnost jak uživatelům, tak vývojařům mít přehled o aktuálních problémech. Díky tomuto systému byly jistě otestovány všechny standardní situace, protože se do testování zapojili jak dispečeri či zaměstnanci ČD, tak i zájemci o železniční simulátory. Domnívám se, že konečný pokles nápadů i připomínek je důsledkem funkčního stavu, neboť s některými testery jsem neustále v kontaktu.

## 6 Závěr

### 6.1 Zhodnocení

Vytyčených cílů diplomové práce jsem úspěšně docílil. V této diplomové práci jsem vytvořil univerzální jádro pro řízení železniční dopravy, ke kterému se dají připojit různé řídicí moduly. Nejmodernější používaný systém Českých drah, který splňuje standard JOP, byl zpracován v základních požadavcích. Velkým přínosem je simulace reálného provozu ve stanici, kdy vlaky jezdí dle skutečného jízdního řádu (GVD). Tak si dispečer může vyzkoušet ještě před nástupem do nového pracoviště skutečný provoz. Značným přínosem je také síťové řízení, ve kterém lze převzít určitou oblast na místní ovládání. Tato možnost momentálně není dostupná v žádném jiném simulátoru, který umí simulovat standard JOP.

Celý projekt včetně podpůrných editorů Builder a Stag a stanic od uživatelů dokazuje, že vytvořit reálné modelové kolejiště i provoz bylo umožněno. Věrohodnost zobrazení i řízení dokazuje spokojenost z řad uživatelů. Vždy je však možné něco vylepšit. Z neodborné veřejnosti jsou kladeny požadavky pro výstup po simulaci (nějaké bodové hodnocení). Bodové hodnocení je však spíše prvek herního charakteru a tímto směrem nebyl program veden.

### 6.2 Možné pokračování

Směr budoucnosti je nastaven bez velkých omezení. Jedním z plánovaných modulů se má stát řízení modelového kolejiště. Tento modul nemusí být nikterak náročný, protože může využívat stávajícího modulu pro řízení JOP a s novým modulem se připojit na běžící aplikaci. Pak bude nutné pouze přenést stavy jednotlivých prvků k modelu kolejiště. Přenesením stavů však vyřešíme pouze polovinu problému. Podíváme-li se na stejnou úlohu z druhé strany, bude nutné přenést povely z fyzického zařízení zpět do simulačního jádra jako mechanickou odezvu. V úvahu přichází vytvořit moduly starších železničních zařízení, která se dosud používají. Tam již bude docházet k mnohým k různým omezením, protože některé stanice mají výjimky oproti standardním předpisům. Takovým modulem se může stát například zařízení typu TEST. Protože ne každý má prostory a finanční možnosti k postavení modelového kolejiště, bylo by jistě dobré mít možnost simulovat provoz v 3D zobrazení. Náročnost takového projektu by narostla do jiných rozměrů a unikátnost a přínos celého projektu by se značně zvětšila. Přidáním možnosti vyzkoušet si i práci strojvedoucího, by se celý systém začal stávat univerzální pomůckou.

Dalším rozvojem simulátoru by mohl být automatický dispečer, který by řídil provoz dle výpočtů a byl by závislý na aktuální dopravní situaci. Tento problém je dle mého názoru nesnadný, neboť Současné předpisy takový přístup nepovolují. V zahraničí mají vlaky vyšší kategorie například definovanou jízdni cestu, která se staví automaticky v určitém čase před příjezdem vlaku. Tím se snižuje možnost vzniku zpoždění vlaku. U nás takovým místem je tříznakový nebo čtyřznakový autoblok, ve kterém se nemohou křížit jízdni cesty. Kdyby se pak zadalo do modelu celé modelové kolejiště státu, mohl by se udržovat provoz automaticky s tím, že by existovala možnost převzít jistou oblast na ruční ovládání.

Cest ke zlepšení simulátoru je hodně. Je však třeba si uvědomit, že náklady na rozšíření projektu se vrátit nemohou a i konkurence má své vytyčené cíle. Přínosem této diplomové práce je kromě praktického využití i rozšiřování povědomí o simulátorech řízení železniční dopravy.



## 7 Literatura

- [1] AŽd praha s.r.o.  
<http://www.azd.cz/produkty-a-sluzby/systemy-pro-kolejovou-dopravu/stanicni-zabezpecovaci-zarizeni/esa-11-elektronicke-stanicni-zabezpecovaci-zarizeni/>.
- [2] Ak signal brno a.s. <http://www.aksignal.cz/zab-remote98.html>.
- [3] M. V. a Pavel Tvrdik. Dokument popisující zadávání, zpracování a hodnocení diplomových prací na katedře počítačů fel Čvut, 2006.  
<https://info336.felk.cvut.cz/clanek.php?id=400&cele=1>.
- [4] České dráhy. Základní technické požadavky - jednotné oblužné pracoviště, 2000.
- [5] R. Jančar. Jak se řídí vlaky na pražském hlavním nádraží.  
fotoreportáž - idnes.cz. [http://technet.idnes.cz/tec\\_technika.asp?c=A060726\\_155153\\_tec\\_checktech\\_NYV](http://technet.idnes.cz/tec_technika.asp?c=A060726_155153_tec_checktech_NYV).
- [6] J. Konrád. Gordikon - simulace řízení vlakové dopravy. <http://softikon.wz.cz/>.
- [7] Mantis Bug tracking system <http://www.mantisbt.org/>.





## **A Seznam použitých zkratek**

**CD** Compact Disk

**ČD** České Dráhy

**DN** Dovolující Návěst

**DP** Diplomová Práce

**DOZ** Dálkově Ovládané zabezpečovací Zařízení

**ESTW** Elektronische STellWerke

**EZ** Editace Zásobníka

**FIFO** First In First Out

**GVD** Grafikon Vlakové Dopravy

**IP** Internet Protocol

**JOP** Jednotné Obslužné Pracoviště

**KO** Klopný Obvod

**OK** ObsluhaKlienta

**PDF** Portable Document Format

**PN** Přivolávací Návěst

**PV** Přednostní Volba

**RC** Rušení Cesty

**TCP** Transmission Control Protocol

**TEST** Typové Elektrické Stavědlo releového Typu

**UML** Unified Modeling Language

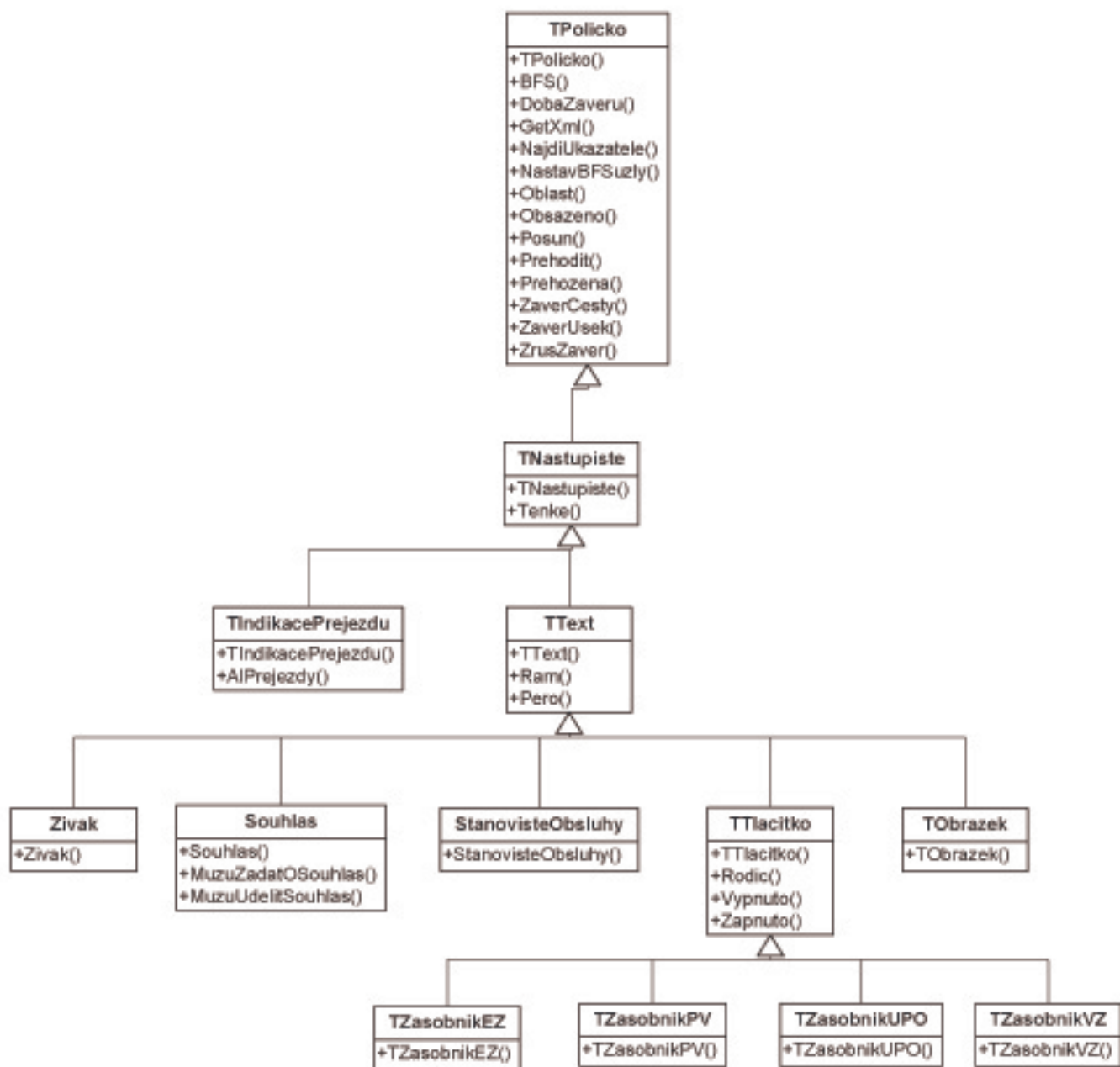
**VZ** Volba do Zásobníka

**XML** eXtensible Markup Language

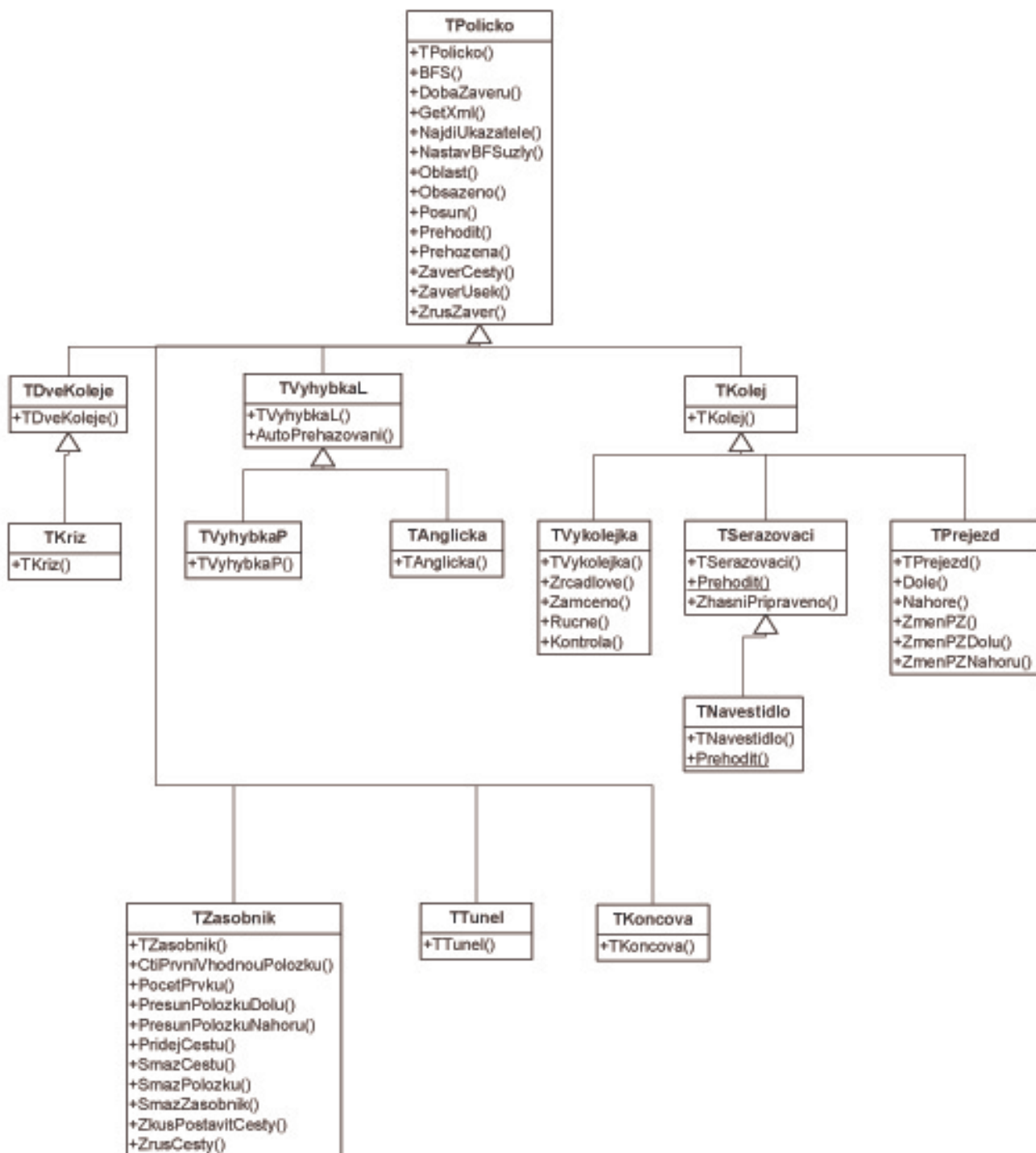
**ZTP** Základní Technické Požadavky



## B UML diagramy



Obrázek B.1: Diagram tříd pro políčko mapy (1. část) - zobrazeny jsou jen některé metody



Obrázek B.2: Diagram tříd pro políčko mapy (2. část) - zobrazeny jsou jen některé metody

## C XML Schema

```

<?xml version="1.0" encoding="windows-1250" ?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:annotation>
    <xs:documentation xml:lang="cs-cz">
      Schéma definuje model stanice, jízdní řád vlaků (včetně řazení) a globální
      nastavení.
    </xs:documentation>
  </xs:annotation>
  <xs:element name="root">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="policka">
          <xs:annotation>
            <xs:documentation xml:lang="cs-cz">
              Element POLICKA definuje mapu jako dvojrozměrné pole.
            </xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" name="policko">
                <xs:annotation>
                  <xs:documentation xml:lang="cs-cz">
                    Element POLICKO definuje jedno políčko mapy. Toto políčko je
                    určeno souřadnicemi X a Y. Atribut TVAR udává o jaký prvek
                    se jedná (např. NAVESTIDLO, KOLEJ, TEXT apod.).
                  </xs:documentation>
                </xs:annotation>
                <xs:complexType>
                  <xs:sequence minOccurs="0">
                    <xs:element minOccurs="0" maxOccurs="unbounded" name="type">
                      <xs:complexType>
                        <xs:simpleContent>
                          <xs:extension base="xs:string">
                            <xs:attribute name="id" type="xs:string"
                              use="required" />
                          </xs:extension>
                        </xs:simpleContent>
                      </xs:complexType>
                    </xs:element>
                    <xs:element minOccurs="0" maxOccurs="unbounded" name="to">
                      <xs:complexType>
                        <xs:simpleContent>
                          <xs:extension base="xs:string">
                            <xs:attribute name="mult" type="xs:unsignedByte"

```

```

        use="required" />
    </xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element minOccurs="0" maxOccurs="unbounded" name="pos">
    <xs:complexType>
        <xs:attribute name="x" type="xs:unsignedInt"
            use="required" />
        <xs:attribute name="y" type="xs:unsignedInt"
            use="required" />
    </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="x" type="xs:unsignedInt" use="required" />
<xs:attribute name="y" type="xs:unsignedInt" use="required" />
<xs:attribute name="tvar" type="xs:unsignedByte"
    use="optional" />
<xs:attribute name="stav" type="xs:string" use="optional" />
<xs:attribute name="delka" type="xs:unsignedInt"
    use="optional" />
<xs:attribute name="cislo" type="xs:int" use="optional" />
<xs:attribute name="oblast" type="xs:byte" use="optional" />
<xs:attribute name="napeti" type="xs:unsignedByte"
    use="optional" />
<xs:attribute name="rOsobni" type="xs:unsignedInt"
    use="optional" />
<xs:attribute name="rNakladni" type="xs:unsignedInt"
    use="optional" />
<xs:attribute name="typ" type="xs:string" use="optional" />
<xs:attribute name="text" type="xs:string" use="optional" />
<xs:attribute name="fName" type="xs:string" use="optional" />
<xs:attribute name="fSize" type="xs:string" use="optional" />
<xs:attribute name="fBold" type="xs:string" use="optional" />
<xs:attribute name="fItalic" type="xs:string"
    use="optional" />
<xs:attribute name="fUnderline" type="xs:string"
    use="optional" />
<xs:attribute name="fBarva" type="xs:int" use="optional" />
<xs:attribute name="smer" type="xs:unsignedShort"
    use="optional" />
<xs:attribute name="cesta" type="xs:string" use="optional" />
<xs:attribute name="rDo0dbocky" type="xs:unsignedInt"
    use="optional" />
<xs:attribute name="rZ0dbocky" type="xs:unsignedInt"
    use="optional" />
<xs:attribute name="rZRovne" type="xs:unsignedInt"

```

```

        use="optional" />
        <xs:attribute name="rDoRovne" type="xs:unsignedInt"
        use="optional" />
        <xs:attribute name="x1" type="xs:unsignedInt"
        use="optional" />
        <xs:attribute name="y1" type="xs:unsignedInt"
        use="optional" />
        <xs:attribute name="mult" type="xs:unsignedByte"
        use="optional" />
        <xs:attribute name="cas" type="xs:time" use="optional" />
        <xs:attribute name="yText" type="xs:unsignedInt"
        use="optional" />
        <xs:attribute name="xText" type="xs:unsignedInt"
        use="optional" />
        <xs:attribute name="auto" type="xs:string"
        use="optional" />
        <xs:attribute name="sklon" type="xs:byte"
        use="optional" />
        <xs:attribute name="zrcadlove" type="xs:string"
        use="optional" />
        <xs:attribute name="celejmeno" type="xs:string"
        use="optional" />
        <xs:attribute name="smerprij" type="xs:string"
        use="optional" />
        <xs:attribute name="smerodj" type="xs:string"
        use="optional" />
    </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="pocX" type="xs:unsignedInt" use="required" />
<xs:attribute name="pocY" type="xs:unsignedInt" use="required" />
<xs:attribute name="autor" type="xs:string" use="required" />
<xs:attribute name="email" type="xs:string" use="required" />
<xs:attribute name="popis" type="xs:string" use="required" />
<xs:attribute name="jmeno" type="xs:string" use="required" />
</xs:complexType>
</xs:element>
<xs:element name="pozice">
    <xs:annotation>
        <xs:documentation xml:lang="cs-cz">
            Element POZICE slouží pro možnost nastavení polohy obrazovky.
            Souřadnice X a Y udávají levé horní políčko na mapě.
            Číslo je využito jako identifikace.
        </xs:documentation>
    </xs:annotation>
</xs:complexType>
<xs:sequence>

```

```

<xs:element maxOccurs="unbounded" name="souradnice">
  <xs:complexType>
    <xs:attribute name="x" type="xs:unsignedInt" use="required" />
    <xs:attribute name="y" type="xs:unsignedInt" use="required" />
    <xs:attribute name="cislo" type="xs:int" use="required" />
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="station">
  <xs:annotation>
    <xs:documentation xml:lang="cs-cz">
      Element STATION přiřazuje zkratce stanice její celé jméno.
      Zkratka stanice musí být unikátní.
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="stanice">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" name="nadrazi">
              <xs:complexType>
                <xs:attribute name="zkratka" type="xs:string"
                  use="required" />
                <xs:attribute name="celejmeno" type="xs:string"
                  use="required" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element maxOccurs="unbounded" name="gvd">
  <xs:annotation>
    <xs:documentation xml:lang="cs-cz">
      Element GVD definuje grafikon vlakové dopravy. A to jak osobní,
      tak nákladní dopravu. Zároveň umožňuje nastavit omezení dnů, kdy
      má vlak jet a umožňuje řazení vlaků. GVD lze definovat pro různá
      časová období.
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>

```



```

<xs:element name="trains">
  <xs:annotation>
    <xs:documentation xml:lang="cs-cz">
      Element TRAINS obsahuje vlaky jak osobní tak nákladní
      dopravy. Vlak může mít zastávkový jízdní řád, řazení
      soupravy a přiřazeno jedno omezení DATE atributem KDYJEDE.
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" name="train">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="zastavky">
              <xs:annotation>
                <xs:documentation xml:lang="cs-cz">
                  Element ZASTAVKY obsahuje zastávky vlaku.
                  Zastávka je jednoznačně určena názvem stanice a
                  číslem koleje.
                </xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:sequence minOccurs="0">
                  <xs:element maxOccurs="unbounded" name="stavi">
                    <xs:complexType>
                      <xs:attribute name="st" type="xs:string"
                        use="required" />
                      <xs:attribute name="kol"
                        type="xs:unsignedInt" use="required" />
                      <xs:attribute name="cas" type="xs:time"
                        use="required" />
                      <xs:attribute name="kpov"
                        type="xs:unsignedByte" use="required" />
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="razeni">
              <xs:annotation>
                <xs:documentation xml:lang="cs-cz">
                  Element RAZENI určuje řazení vlaku. Vlaku jsou
                  přiřazeny vozy z elementu VOZY. Informace o vozu
                  jsou rozšířeny o směr, poznámku a id obrázku.
                </xs:documentation>
              </xs:annotation>
              <xs:complexType>

```

```

<xs:sequence>
  <xs:element maxOccurs="unbounded" name="vuz">
    <xs:complexType>
      <xs:attribute name="typ" type="xs:string"
        use="required" />
      <xs:attribute name="smer" type="xs:string"
        use="optional" />
      <xs:attribute name="obrid"
        type="xs:unsignedByte" use="optional" />
      <xs:attribute name="prevr"
        type="xs:unsignedByte" use="optional" />
      <xs:attribute name="pozn" type="xs:string"
        use="optional" />
    </xs:complexType>
  </xs:element>
</xs:sequence>
<xs:attribute name="delka"
  type="xs:unsignedShort" use="required" />
<xs:attribute name="hmotnost"
  type="xs:unsignedShort" use="required" />
<xs:attribute name="vykon"
  type="xs:unsignedShort" use="required" />
<xs:attribute name="maxv"
  type="xs:unsignedShort" use="required" />
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="cislo" type="xs:unsignedInt"
  use="required" />
<xs:attribute name="typ" type="xs:string"
  use="required" />
<xs:attribute name="jmeno" type="xs:string"
  use="required" />
<xs:attribute name="jednotka" type="xs:unsignedByte"
  use="required" />
<xs:attribute name="zastavuje" type="xs:unsignedInt"
  use="required" />
<xs:attribute name="vznika" type="xs:string"
  use="required" />
<xs:attribute name="smerprijezdu" type="xs:string"
  use="required" />
<xs:attribute name="casprijezdu" type="xs:string"
  use="required" />
<xs:attribute name="oznamenismeruprijezdu"
  type="xs:string" use="required" />
<xs:attribute name="konci" type="xs:string"
  use="required" />

```

```

    <xs:attribute name="smer" type="xs:string"
    use="required" />
    <xs:attribute name="casodjezdu" type="xs:string"
    use="required" />
    <xs:attribute name="naposun" type="xs:unsignedByte"
    use="required" />
    <xs:attribute name="konciposunem" type="xs:unsignedByte"
    use="required" />
    <xs:attribute name="cekatnavystup"
    type="xs:unsignedByte" use="required" />
    <xs:attribute name="oznamenismeruodjezdu"
    type="xs:string" use="required" />
    <xs:attribute name="kdyjede" type="xs:string"
    use="required" />
    <xs:attribute name="poznamka" type="xs:string"
    use="required" />
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="dates">
  <xs:annotation>
    <xs:documentation xml:lang="cs-cz">
      Element DATES slouží jako datumové omezení stejně jako
      v jízdním řádu. Pomocí elementů RUN a STOP jsou určeny dny,
      kdy vlak může nebo nemůže jet.
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" name="date">
        <xs:complexType>
          <xs:sequence>
            <xs:choice maxOccurs="unbounded">
              <xs:element maxOccurs="unbounded" name="run">
                <xs:complexType>
                  <xs:attribute name="at" type="xs:string"
                  use="required" />
                  <xs:attribute name="till" type="xs:string" />
                </xs:complexType>
              </xs:element>
              <xs:element maxOccurs="unbounded" name="stop">
                <xs:complexType>
                  <xs:attribute name="at" type="xs:string"
                  use="required" />
                  <xs:attribute name="till" type="xs:string" />
                </xs:complexType>
              </xs:element>
            </xs:choice>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

        </xs:complexType>
    </xs:element>
</xs:choice>
</xs:sequence>
<xs:attribute name="name" type="xs:string"
    use="required" />
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="vozy">
    <xs:annotation>
        <xs:documentation xml:lang="cs-cz">
            Element VOZY definuje podrobné informace o jednotlivých
            vozech.
        </xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" name="vuz">
                <xs:complexType>
                    <xs:attribute name="id" type="xs:string"
                        use="required" />
                    <xs:attribute name="typ" type="xs:string"
                        use="optional" />
                    <xs:attribute name="podtrida" type="xs:string"
                        use="optional" />
                    <xs:attribute name="popis" type="xs:string"
                        use="optional" />
                    <xs:attribute name="hmotnost" type="xs:unsignedShort"
                        use="optional" />
                    <xs:attribute name="delka" type="xs:string"
                        use="optional" />
                    <xs:attribute name="vykon" type="xs:unsignedShort"
                        use="optional" />
                    <xs:attribute name="sila" type="xs:unsignedShort"
                        use="optional" />
                    <xs:attribute name="max_rych_hnaci" type="xs:unsignedShort"
                        use="optional" />
                    <xs:attribute name="max_rych" type="xs:unsignedShort"
                        use="optional" />
                    <xs:attribute name="elektro" type="xs:unsignedByte"
                        use="optional" />
                    <xs:attribute name="vyrobce" type="xs:string"
                        use="optional" />
                    <xs:attribute name="autor" type="xs:string"

```

```

        use="optional" />
        <xs:attribute name="img" type="xs:short"
        use="optional" />
        <xs:attribute name="imgex" type="xs:unsignedByte"
        use="optional" />
        <xs:attribute name="typ_nakladu" type="xs:string"
        use="optional" />
        <xs:attribute name="naklad" type="xs:unsignedShort"
        use="optional" />
        <xs:attribute name="osob" type="xs:unsignedShort"
        use="optional" />
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="nahsoupravy">
    <xs:annotation>
        <xs:documentation xml:lang="cs-cz">
            Element NAHSOUPRAVY definuje řazení souprav, které je možné
            využít při generování náhodných vlaků. Atribut id je klíč.
        </xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" name="nahvlak">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="razeni">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element maxOccurs="unbounded" name="vuz">
                                        <xs:complexType>
                                            <xs:attribute name="typ" type="xs:string"
                                            use="required" />
                                        </xs:complexType>
                                    </xs:element>
                                </xs:sequence>
                            </xs:complexType>
                            <xs:attribute name="delka" type="xs:unsignedShort"
                            use="required" />
                            <xs:attribute name="hmotnost"
                            type="xs:unsignedShort" use="required" />
                            <xs:attribute name="vykon" type="xs:unsignedShort"
                            use="required" />
                            <xs:attribute name="maxv" type="xs:unsignedShort"
                            use="required" />
                        </xs:complexType>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>

```

```
        </xs:element>
      </xs:sequence>
      <xs:attribute name="id" type="xs:string" use="required" />
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="goodstrain">
  <xs:annotation>
    <xs:documentation xml:lang="cs-cz">
      Element GOODSTRAIN slouží pro nastavení frekvence náhodné
      nákladní dopravy.
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:attribute name="init_frequency" type="xs:unsignedByte"
      use="required" />
  </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="section" type="xs:string" use="required" />
<xs:attribute name="sect_desc" type="xs:string" use="required" />
<xs:attribute name="gvd_ver" type="xs:string" use="required" />
<xs:attribute name="gvd_editor" type="xs:string" use="required" />
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

## D Uživatelská / instalační příručka

Uživatelská příručka je velmi rozsáhlá, a proto je součástí instalačního balíčku. Autorem uživatelské příručky je Lukáš Procházka.

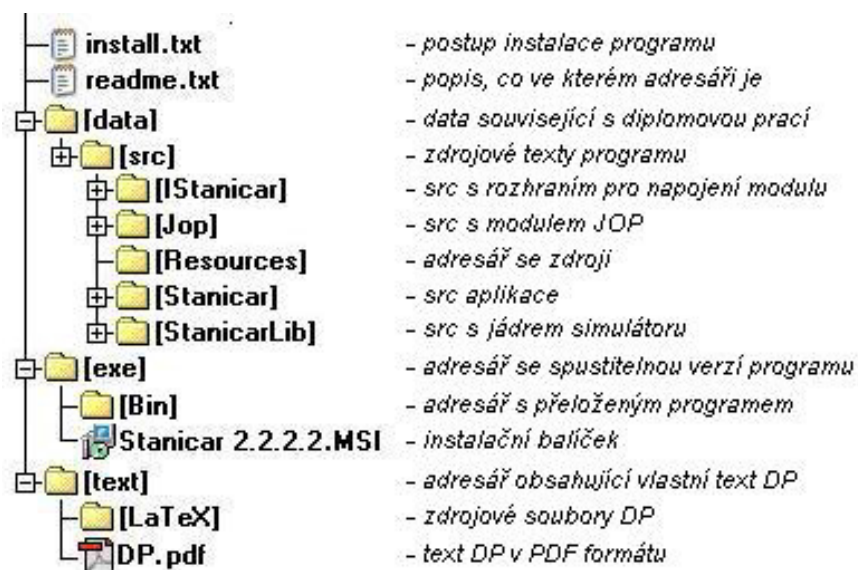
Postup při instalaci:

- 1. Mít jakýkoliv systém Windows, který podporuje platformu Microsoft .NET.
- 2. Nainstalovat .NET framework 2.0.
- 3. Spustit instalační balíček „Stanicar 2.2.2.2.MSI“ a řídit se instalačními instrukcemi.





## E Obsah příloženého CD



Obrázek E.1: Seznam příloženého CD