

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ
V PRAZE**

FAKULTA DOPRAVNÍ



Tomáš Starý

**DATOVÝ POPIS KOLEJIŠTĚ DOPRAVNÍHO SÁLU
FD**

Bakalářská práce

2020

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta dopravní

děkan

Konviktská 20, 110 00 Praha 1



K620..... **Ústav dopravní telematiky**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Tomáš Starý

Kód studijního programu a studijní obor studenta:

B 3710 – ITS – Inteligentní dopravní systémy

Název tématu (česky): **Datový popis kolejiště Dopravního sálu FD**

Název tématu (anglicky): Railroad Data Description of the Railway Laboratory of
Faculty of Transportin

Zásady pro vypracování

Při zpracování bakalářské práce se řiďte následujícími pokyny:

- Analyzujte metodiku RailTopoModel a RailML pro datový popis železničního systému.
- Analyzujte datový popis železničního systému v prostředí ERSW SW ERTMS/ETCS Operational Simulator.
- Zhodnoťte využití těchto popisů železničního systému pro potřeby Dopravního sálu FD.
- Navrhněte a modelově ověřte datový popis železničního systému Dopravního sálu FD.



Rozsah grafických prací: dle pokynu vedoucího práce

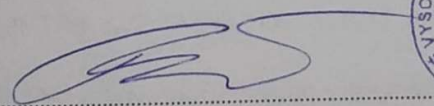
Rozsah průvodní zprávy: minimálně 35 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)

Seznam odborné literatury: Internetové zdroje k problematice RailTopoModel, RailML
ERSA SW - ERTMS/ETCS Operational
SimulatorDatabázové systémy

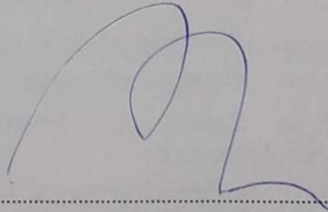
Vedoucí bakalářské práce: **doc. Ing. Martin Leso, Ph.D.**

Datum zadání bakalářské práce: **10. srpna 2019**
(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

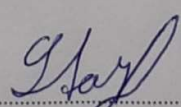
Datum odevzdání bakalářské práce: **10. srpna 2020**
a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia


Ing. Zuzana Bělinová, Ph.D.
vedoucí
Ústavu dopravní telematiky




doc. Ing. Pavel Hrubeš, Ph.D.
děkan fakulty

Potvrzuji převzetí zadání bakalářské práce.


Tomáš Starý
jméno a podpis studenta

V Praze dne10. srpna 2019

Poděkování

Tímto bych chtěl poděkovat doc. Ing. Martinu Lesovi, Ph.D. za jeho čas, konzultace a návrhy, které přispěly ke zkvalitnění této práce. Dále bych chtěl poděkovat Ing. Adamu Hlubučkovi za jeho odborné rady. Zároveň bych chtěl poděkovat Ing. Petru Kouteckému za poskytnuté schéma, které posloužilo jako základ pro praktickou část.

Veliké díky patří mojí rodině, která mi poskytovala veškerou podporu při práci a bez které by tato práce nemohla vzniknout.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Nemám závažný důvod proti užívání 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 10.8.2020

.....

podpis

Abstrakt

Práce se zabývá možnostmi datového popisu železniční infrastruktury. První a druhá kapitola obsahuje teorii k popisu železniční infrastruktury ve formátu RailTopoModel a railML. Třetí kapitola se zaměřuje na software ERSA a možnosti exportu ve formátu railML z tohoto programu. Poslední kapitola ukazuje možný směr popisu železniční infrastruktury. Část železniční infrastruktury je popsána v grafickém programu a soubor ve formátu railML je potom z tohoto grafického popisu vyexportován.

Klíčová slova

datový popis železniční infrastruktury, RailTopoModel, railML, ERSA, RaiLAiD

Abstract

This bachelor thesis deals with the possibilities of data-based description of the railway infrastructure. The first and second chapters contain the theory to describe the railway infrastructure in RailTopoModel and railML format. The third chapter focuses on ERSA software and the possibility of exporting in railML format from this program. The last chapter shows the possible direction of the railway infrastructure description. Part of the railway infrastructure is described in a graphical program and the file in railML format is then exported from this graphical description.

Key words

data-based description of the railway infrastructure, RailTopoModel, railML, ERSA, RaiLAiD

Obsah

Seznam použitých zkratk	5
Úvod	6
1 RailTopoModel	7
1.1 Možnosti využití	7
1.2 Stav vývoje	7
1.3 První výsledky	8
1.4 Standard IRS 30100	8
1.4.1 Souvislý graf	8
1.4.2 Základní prvky v databázovém pohledu RTM	9
1.4.2.1 Schéma UML	9
1.4.3 Úrovně pohledu	10
1.4.3.1 Úroveň Micro	10
1.4.3.2 Úroveň Meso	11
1.4.3.3 Úroveň Macro	11
1.4.3.4 Úroveň Nano a další úrovně	11
1.4.4 Struktura RTM	12
1.4.4.1 Topologická struktura	12
1.4.4.2 Hierarchická struktura	13
1.4.5 Určování polohy	16
1.4.5.1 Geometrické souřadnice	16
1.4.5.2 Lineární referenční systém (LRS)	16
1.4.5.3 Vnitřní referenční systém (VRS)	17
1.4.6 Umisťování objektů v síti	19
1.4.6.1 Bodová lokace	19
1.4.6.2 Lineární lokace	20
1.4.6.3 Plošná lokace	21
2 RailML	22
2.1 Úvod	22
2.2 Schémata	23
2.3 Schéma IS	23
2.3.1 Kostra souboru railML	23
2.3.2 Common	24
2.3.3 Infrastructure	25
2.3.3.1 Elementy	26
2.3.3.2 Relace	26

2.3.3.3 Sítě.....	27
2.3.3.4 Prvky funkční infrastruktury	27
3 ERSA operační simulátor.....	28
3.1 ERTMS/ETCS	28
3.2 ERSA	28
3.2.1 Nástroj na vytváření infrastruktury.....	28
3.2.2 Nástroj na simulaci.....	29
3.2.3 Formát ukládání dat	30
3.3 Závěr	30
4 Návrh a implementace v DSFD.....	31
4.1 Možnosti využití různých popisů v DSFD.....	31
4.1.1 Rhinoceros.....	32
4.2 RailAiD.....	32
4.3 Popis stanice Senohraby a přilehlého úseku v grafickém rozhraní	34
4.4 Popis stanice Senohraby v railML.....	39
4.5 Závěr.....	39
Závěr	41
Zdroje	43
Obrázky	46
Tabulky	48
Přílohy	49

Seznam použitých zkratek

DSFD		Dopravní sál Fakulty dopravní
RTM	RailTopoModel	RailTopoModel
GNSS	Global Navigation Satellite System	Globální družicový polohový systém
GPS	Global Positioning System	Globální polohový systém
RINF	Registers of Infrastructure	Registr infrastruktury
ERTMS	The European Railway Traffic Management System	Evropský systém pro řízení železniční dopravy
ETCS	European Train Control System	Evropský vlakový zabezpečovací systém
ERIM	European Rail Infrastructure Masterplan	Evropský železniční infrastrukturní plán
UIC	International Union of Railways	Mezinárodní železniční unie
IRS	International Railway Solutions	Mezinárodní železniční řešení
UML	Unified Modeling Language	Univerzální modelovací jazyk
XML	Extensible Markup Language	Rozšiřitelný značkovací jazyk
WGS84	World Geodetic System 1984	Světový geodetický systém 1984
LRS		Lineární referenční systém
VRS		Vnitřní referenční systém
GSM-R	Global System for Mobile communications for Railways	Globální systém pro mobilní komunikaci na železnici
RBC	Radio Block Centre	Radiobloková centrála
SoM	Start of Mission	Začátek mise
MA	Movement Authority	Povolení k jízdě
HMI	Human Machine Interface	Rozhraní člověk-stroj
CSV	Comma-separated values	Hodnoty oddělené čárkami

Úvod

Infrastrukturu v DSFD je nutné v budoucnosti popsat tak, aby bylo možné tento popis využít v aplikacích, které potřebují přesná data o poloze prvků v kolejišti. Záměrem této práce je analyzovat a popsat metodiku RailTopoModel a railML pro popis reálného železničního systému. První část práce se zaměřuje na shrnutí a překlad oficiálních zdrojů k formátům RailTopoModel a railML. Je blíže prozkoumáno, proč tento model vzniká a kdo stojí za jeho vývojem. Jsou také popsány možnosti jeho využití. Následně se práce zaměří na otázku, jakým způsobem RailTopoModel popisuje infrastrukturu, tedy na jakých teoretických základech model stojí, jaké rozlišuje úrovně popisu, jak se liší jeho topologická a hierarchická struktura, s jakými souřadnicemi je možné v modelu pracovat a jak se poté v modelové síti umísťují objekty.

Druhá kapitola je zaměřena na to, jak se tento popis následně promítá do souborů ve formátu railML, je zodpovězena otázka, co je to railML, na jakém základě je vystavěno a kdy je možné ho použít. Dále je podrobněji prozkoumána část railML věnující se infrastrukturní části železničního systému, kterou je vhodné zpracovávat jako první, jelikož se na něj ostatní prvky vážou. Ať už jde o popis zabezpečovacího zařízení, stavění vlakových cest nebo přípravu jízdnicích řádů. Je popsáno, z jakých částí se soubor v railML skládá a jaké obsahují jednotlivé části informace.

Vzhledem k povaze dokumentů v railML není vhodné a efektivní vytvářet je přímo v textové podobě. Efektivnějším řešením je možnost popisovat infrastrukturu v grafickém editoru a poté ji do railML exportovat. To samozřejmě vyžaduje existenci takových programů, které export do railML umožňují. V rámci práce bude prozkoumána možnost použití softwaru ERSA Track Editor pro tento grafický popis a následný export. Dále bude prozkoumána možnost použití tohoto programu pro výuku ERTMS/ETCS.

V následné praktické části budou prozkoumány možnosti využití výše zmíněných datových popisů infrastruktury pro potřeby Dopravního sálu Fakulty dopravní. Budou prozkoumány programy, ve kterých je možné popis provádět a bude vybrán jeden, ve kterém proběhne návrh možného řešení části kolejiště. Zároveň dojde k popisu programu Rhinoceros, který je možné v budoucnu v DSFD využívat, jelikož je do něj popis kolejiště nyní vymodelovaný v AutoCADu jednoduše převoditelný. Praktická část této práce se zaměří na popis stanice Senohraby, která je součástí kolejiště DSFD, v grafickém rozhraní programu RailAiD a následně na její export do formátu railML. Stanice bude nakreslena v grafickém editoru podle možností vybraného programu, budou vymodelovány prvky, které nebudou součástí základní knihovny prvků a potom budou tyto prvky přidány do zkresleného schématu stanice. Dojde také k přiřazení vlastností těmto prvkům. Následně proběhne export těchto dat do souboru ve formátu railML. Tento soubor bude možné dále používat pro budoucí aplikace. Na závěr bude zhodnocena kvalita použitého programu a jeho možné další využití.

1 RailTopoModel

Výměna infrastrukturních dat uvnitř organizace a s obchodními partnery patří mezi každodenní aktivity železničních správců infrastruktury v celé Evropě. Tato data jsou využívána v téměř všech činnostech správců infrastruktury, od údržby a provozování dráhy až po řízení dopravy. Data jsou mezi správci infrastruktury předávána pomocí různých datových formátů. Tyto formáty jsou většinou zaměřeny jen na určitý úkol a nejsou mezinárodně standardizované. Neexistence jednotného datového formátu přináší správcům infrastruktury a dalším uživatelům dat vyšší náklady způsobené nemožností data ihned využít pro své aplikace. Tyto vícenásobné náklady jsou způsobené nutností vytvářet specifická rozhraní pro převod dat mezi těmito formáty.

RailTopoModel je logický model, který je vyvíjen za účelem standardizace popisu dat o železniční infrastruktuře. Společně s railML, které definuje schéma pro výměnu dat, má zjednodušit přenosy dat mezi různými systémy. Od toho se očekává snížení nákladů, a tedy větší konkurenceschopnost železniční dopravy v porovnání s ostatními druhy dopravy.

Vývojem RTM se zabývá pracovní skupina RailTopoModel Expert Group. Dodatečné informace, příklady, dokumentaci a encyklopedii RTM je možné nalézt přímo na webových stránkách projektu: <http://www.railtopomodel.org>. [1]

1.1 Možnosti využití

RailTopoModel je základní model, který definuje, jak by mohla být data modelována, popisována, zpracovávána, ukládána a zobrazována. Cílem je možnost spojovat, lokalizovat a zobrazovat topologické vztahy železniční sítě a objektů na ní umístěných, podporovat vícenásobný referenční systém zahrnující lineární reference, GNSS a různé typy souřadnic, popisovat vlastnosti železniční sítě, jejích součástí a na ni navázaných událostí v několika rozlišovacích hladinách, od podrobných po obecné. [1]

1.2 Stav vývoje

Výměna železničních infrastrukturních dat pro projekty jako RINF a ETCS stojí železniční společnosti mnoho času a peněz. Problémem je absence společného modelu a standardu. [2] Na základě těchto požadavků byla vypracována pracovní skupinou ERIM, spadající pod UIC, studie proveditelnosti. Výsledky této studie byly poprvé prezentovány na konferenci v Paříži v roce 2013. Studie prokázala, že je technicky možné vytvořit RTM a související formát výměny dat pomocí nástrojů aktivní vývojářské komunity, ale problémy mohou nastat při zajištění politické podpory a financování. [3]

Studie proveditelnosti dále doporučuje pokračovat ve vývoji standardizovaného formátu výměny dat založeném na dvou komponentech:

- Topologický datový model (RailTopoModel®)
- Formát výměny dat (railML® 3) [2]

1.3 První výsledky

Skupinou RTM Expert Modelling Group byl následně vyvinut RailTopoModel jako základní model pro výměnný formát. Je to systematický model popisující železniční infrastrukturu bez detailního zaměření na jednotlivá využití.

Tento datový model má podporovat všechny potřeby železničních podniků. Zahrnuje tyto koncepty:

- Topologie: logická reprezentace železniční sítě jako grafu
- Objekty a události: bodové (např. balíza), lineární (např. nástupiště), plošné (např. tarifní zóna)
- Cesty: možné pohyby vozidel v rámci trati a stanic
- Referenční systém: geografické souřadnice, lineární reference, poloha vzhledem ke staničení
- Víceúrovňová agregace: tratě, koridory, ... (RTM je navržen, aby měl stejnou strukturu na každé úrovni)
- Traťová geometrie

RTM bude základem pro railML 3, novou verzi výměnného datového formátu. [2]

1.4 Standard IRS 30100

Standard IRS 30100 RailTopoModel je určen k použití ve všech situacích, které pracují s návrhem, konstrukcí, provozem a údržbou železniční sítě. Je tak základem pro rychlé, jednoznačné a bezchybné ukládání dat a datový výměnný formát.

RailTopoModel zpracovává nutné pojmy ve formě diagramu UML 2.0. Tato kapitola popisuje teoretické základy modelování v RTM. Jsou použity a shrnuty informace pocházející ze standardu IRS 30100 a z projektové RTM Wiki, která je spravována přímo skupinou ERIM. Databázovou podobu modelu je možné najít přímo ve standardu. [4]

1.4.1 Souvislý graf

Souvislý graf je základní matematický pojem. Graf je dvojice $G = (V, E)$, kde $V = \{v_1, \dots, v_n\}$ je seznam vrcholů a uzlů a $E = \{e_1, \dots, e_m\}$ je seznam hran a oblouků s tím, že každý prvek z E je dvojice prvků z V (např. $e_1 = (v_1, v_2)$).

V RTM je využíván jednoduchý graf, tedy graf bez násobných hran a smyček. Souvislost grafu je poté definována jako stav, kdy mezi každou dvojicí vrcholů grafu existuje sled. [5]



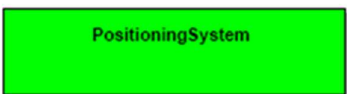

„Sled v grafu je posloupnost vrcholů taková, že mezi každými dvěma po sobě jdoucími je hrana.“ [6]

V souvislém grafu se tedy využívají hrany tak, že definují vztahy souvislosti mezi dvěma vrcholy (uzly). V RTM je souvislý graf využit pro modelování topologie sítě. Topologie je pouze logický pojem a je nezávislá na fyzických prvcích, které ji vytvářejí. V teorii grafů jsou všechny prvky uzly a hrany představují vztahy uzlů. [5]

1.4.2 Základní prvky v databázovém pohledu RTM

RailTopoModel je popsán v jazyce UML. Je dále seskupen do balíčků (packages). Každý balík je sbírkou úzce spjatých modelovacích pojmů. [7]

1.4.2.1 Schéma UML

Package	Colour code	Main element(s)
Base	Grey 	Network, LevelNetwork
Topology	Yellow 	NetElement, Relation, CompositionNetElement, IntrinsicPositioningSystem
Positioning Systems	Green 	PositioningSystem, IntrinsicCoordinate
Net Entities	Light blue 	LocatedNetEntity, EntityLocation

obr. 1: Rozdělení podle barev balíčků ve schématu UML [8]

Jednotlivé balíčky modelu na obr. 1 jsou vybarveny různými barvami. Význam barev je následující:

Základní (Base): Šedá

Balíček „Základní“ obsahuje základní definici dat a sítě.

Příklad: Jméno sítě [7]

Topologie (Topology): Žlutá

Balíček „Topologie“ obsahuje topologické prvky, které tvoří síť. Topologie je zaváděna jako graf s uzly a hranami. Uzly jsou vyjádřeny jako třídy "NetElement" a hrany jako třídy "Relation".
Poznámka: Obě třídy existují na všech úrovních pohledu.

Příklad: výhybky, tratě, zarážedla [7]

Odkazování (Referencing): Zelená

Balíček „Odkazování“ obsahuje seznam metod pro určování polohy prvků sítě nebo polohy na nich navázaných prvků.

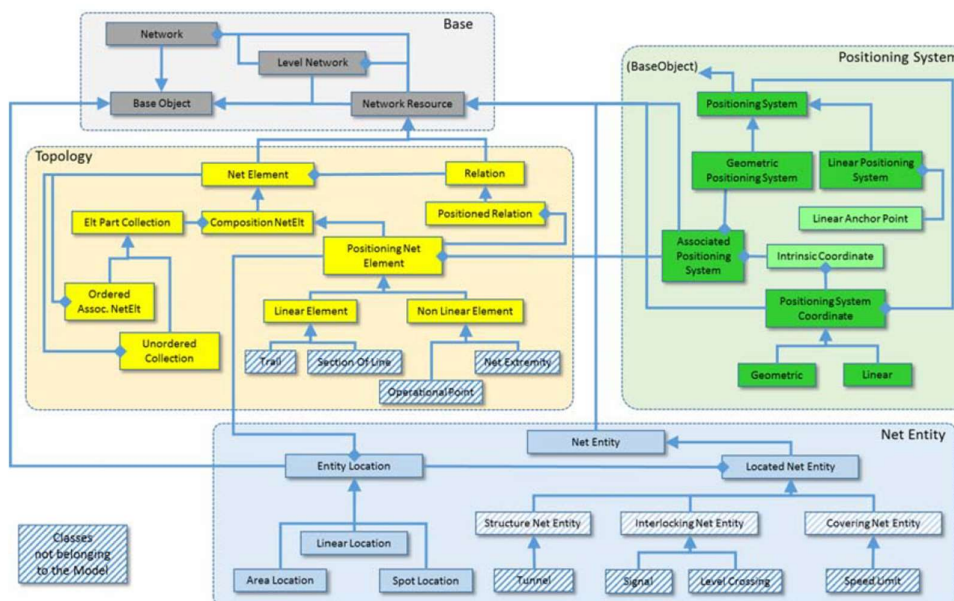
Příklad: Geometrické souřadnice (x, y, z), lineární souřadnice (poloha vzhledem k lineárnímu prvku) [7]

Síťové objekty (Net Entities): Modrá

Balíček „Síťové objekty“ obsahuje další objekty, které jsou ve vztahu se sítí, ale nepatří do topologie. Tedy jsou spojené, nebo přidružené k síti a mají rozměr. Mohou být bodové, liniové nebo plošné.

Příklad: návěstidla, budovy, mosty, ale i rychlostní nebo sklonový profil tratě [7]

Celková databázová struktura je na obr. 2. Zde je rozdělena do skupin podle barev, tak jak to bylo popsáno v předchozím odstavci.



obr. 2: Databázová struktura RTM [4]

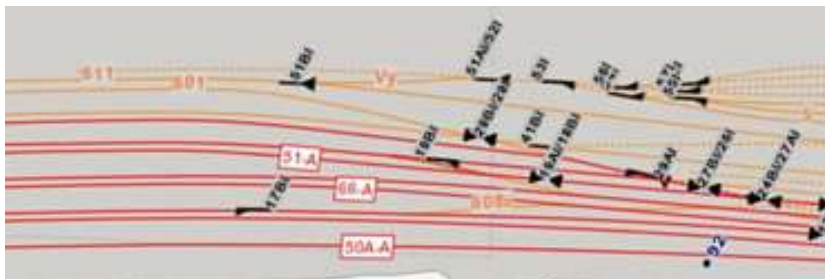
1.4.3 Úrovně pohledu

Dalším úkolem modelu je poskytnout popis sítě na několika úrovních podrobnosti. Tyto rozdílné hladiny jsou spojeny agregačními pravidly.

Na základě zkušeností byly vytvořeny tři hlavní úrovně podrobnosti, tyto ale nejsou povinné a je možné vytvořit si vlastní. [4]

1.4.3.1 Úroveň Micro

Na úrovni Micro se síť svým popisem blíží skutečné infrastruktuře (obr. 3). Nelineárními prvky jsou výhybky a lineárními prvky jsou koleje. Počítá se s použitím těchto dat například při údržbě nebo při použití zabezpečovače ETCS. [4]



1.4.3.2 Úroveň Meso

Úroveň Meso se zabývá popisem tratí mezi operačními body (obr. 4). Nelineárními prvky jsou operační body (železniční stanice, depa, výhybny) a lineárními prvky jsou koleje mezi operačními body. Tuto úroveň lze použít pro plánování přidělování kapacity dráhy. [4]



obr. 4: Popis infrastruktury v úrovni Meso [4]

1.4.3.3 Úroveň Macro

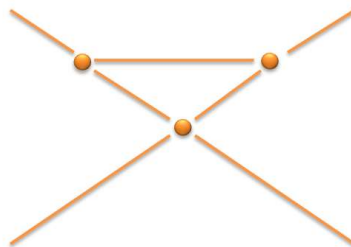
Macro úroveň (obr. 5) popisuje síť na regionální, nebo národní úrovni. Nelineárními prvky jsou hlavní operační body (nejdůležitější železniční stanice) a lineárními prvky jsou celé tratě bez rozlišení kolejí. Úroveň popisu Macro lze využít při konstrukci jízdních řádů. [4]



obr. 5: Popis infrastruktury v úrovni Macro [4]

1.4.3.4 Úroveň Nano a další úrovně

Na úrovni Nano (obr. 6) mohou být popsány prvky sítě v největším detailu. Jedná se například o popis výhybek. Na rozdíl od Micro úrovně je výhybka v Nano úrovni popsána podrobněji. Úroveň Micro a další úrovně lze získat z dat Nano úrovně pomocí RTM agregace. [4]



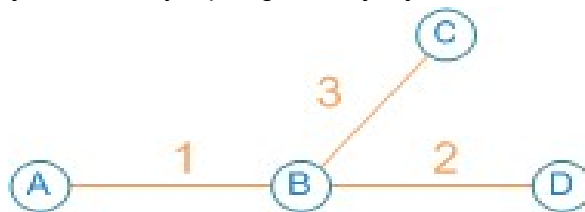
obr. 6: Popis infrastruktury v úrovni Nano (jednoduchá výhybka) [9]

Popis dat na těchto úrovních není povinný a lze si vytvořit i vlastní úrovně, které mohou lépe odpovídat aktuálním potřebám. [10]

1.4.4 Struktura RTM

1.4.4.1 Topologická struktura

Topologická struktura sítě popisuje vztah základních stavebních prvků této sítě na dané úrovni popisu. Třída "NetElement" (dále jen element) znázorňuje všechny stavební prvky topologie. Z důvodu nižšího nároku na výpočetní výkon jsou fyzické uzly sítě odstraněny a zůstávají pouze hrany. Vlastnosti odstraněných fyzických uzlů jsou základem pro vztahy mezi elementy. Na obr. 7 je znázorněna klasická reprezentace železniční sítě se čtyřmi uzly a třemi hranami. V RTM se z hran stávají tři elementy topologie sítě, jak je vidět na obr. 8. [11]

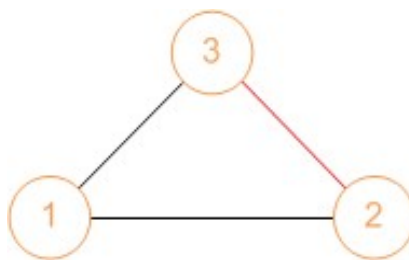


obr. 7: Klasická reprezentace železniční sítě [12]



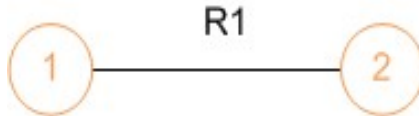
obr. 8: Elementy vzniklé z původní sítě [12]

Při tomto pohledu se ale ztrácí topologická provázanost jednotlivých elementů. Proto jsou přidány prvky třídy "Relation" (dále jen relace). Tyto prvky zajistí topologické vztahy mezi elementy. Každá relace spojuje dva elementy sítě (obr. 9). [11]



obr. 9: Elementy původní sítě s vzájemnými relacemi [12]

Uzly odstraněné na začátku budou při pozdějším popisu reprezentovány entitami. Pro správný popis vztahu lineárního elementu a relace je nutné znát, k jakému konci lineárního elementu relace náleží. Každá relace obsahuje alespoň dva atributy, a to polohu na prvním elementu (značí se "A") a polohu na druhém elementu (značí se "B"), které obsahují polohu s využitím vnitřních souřadnic. "0" je označení pro polohu na začátku elementu a "1" pro polohu na konci elementu. [11]

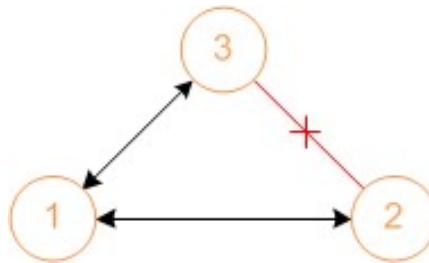


obr. 10: Popis relace R1 [12]

Relace R1 (obr. 10) poskytuje následující informace:

- kruh s označením “1” je lineární element A
- relace R1 má na daném elementu A hodnotu atributu “poloha na elementu A” rovnou “1” (je na konci elementu)
- kruh s označením “2” je lineární element B
- relace R1 má na daném elementu B hodnotu atributu “poloha na elementu B” rovnou “0” (je na začátku elementu)

Nyní je třeba přidat informace o možnostech jízdy vozidla mezi elementy. S využitím počátečního schématu na obr. 7 a základních znalostí funkce výhybek je možné popsat možné směry jízdy přes uzel B. Je zřejmé, že je možný pohyb z bodu 1 do bodů 2 a 3 a obráceně, ale není možný pohyb z bodu 2 do bodu 3 a obráceně. Výsledná síť i se směry možného poježdění je zobrazena na obr. 11. [11]



obr. 11: Výsledná síť s vyobrazením nemožnosti přímé cesty mezi elementy 2 a 3 [12]

1.4.4.2 Hierarchická struktura

RTM obsahuje prostředky pro popis hierarchické struktury sítě, která je tvořena elementy patřící k různým úrovním pohledu. Je možné rozlišovat mezi uspořádaným souborem a neuspořádaným souborem elementů, a tak nastavovat přesné pořadí elementů v detailnější úrovni pohledu, pokud je nutné toto pořadí znát. Lze také popisovat skladbu elementů na jednotlivých úrovních, tedy skutečnost, že jednotlivé elementy na vyšších rozlišovacích úrovních se většinou skládají z více elementů z nižších rozlišovacích úrovní. Pro přechod mezi úrovněmi pohledu lze využít agregaci a disagregaci.

Agregace je postup, při kterém je z nižší úrovně pohledu vytvářena vyšší úroveň pohledu. Následující příklad ukazuje postup při agregaci. Výsledkem agregace jsou v tomto případě operační body OP (významné body sítě, např. dopravní) a koleje spojující tyto body. Popis tratě v úrovni pohledu Micro znázorňuje výhybky a koleje tak, jak je vidět na obr. 12. [13]



obr. 12: Původní síť [14]

S použitím principů RTM je síť rozložena na elementy (obr. 13). Každý bod je elementem sítě. [13]



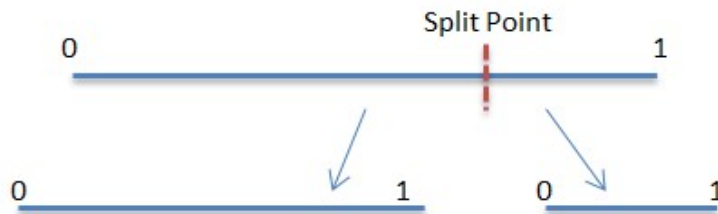
obr. 13: Elementy původní sítě [14]

OP mají známé hranice, které zasahují i do tratí mezi nimi. Jedná se např. o návěstidla. Koleje mezi OP musí být rozděleny na části spadající pod OP a části pod ně nespádající (obr. 14). [13]



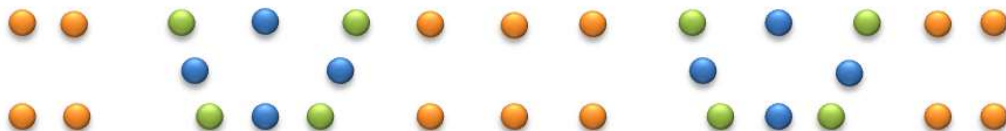
obr. 14: Rozdělení lineárních elementů [14]

V RTM je možné elementy rozdělit. V takovém případě vznikne z původního elementu několik částí. Na obr. 15 je vidět, jak se jeden element dělí na dva elementy. "Split Point" určuje místo, kde vzniká koncový bod prvního elementu, tedy jeho vnitřní souřadnice 1 a počáteční bod druhého elementu, tedy jeho vnitřní souřadnice 0. [13]



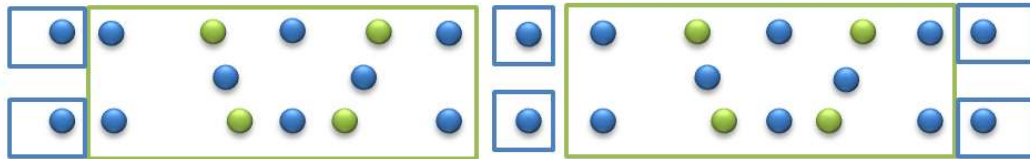
obr. 15: Princip dělení lineárních elementů [14]

Výsledná síť vypadá podle obr. 16. Každý oranžový bod je novým elementem, který vznikl rozdělením původního lineárního elementu. [13]



obr. 16: Rozdělení elementů na části [14]

Všechny elementy jsou následně agregovány do nové sítě, podle obr. 17. Jsou tedy vybrány body, které spolu souvisejí a poté jsou tyto body sloučeny do nového elementu na vyšší rozlišovací úrovni. [13]



obr. 17: Postup spojení elementů do funkčních celků [14]

Vznikne následující schéma (obr. 18), ve kterém jsou modré body lineární elementy a zelené body nelineární elementy. Výsledná síť je potom na obr. 19. [13]



obr. 18: Spojení elementů do funkčních celků [14]



obr. 19: Výsledná síť [14]

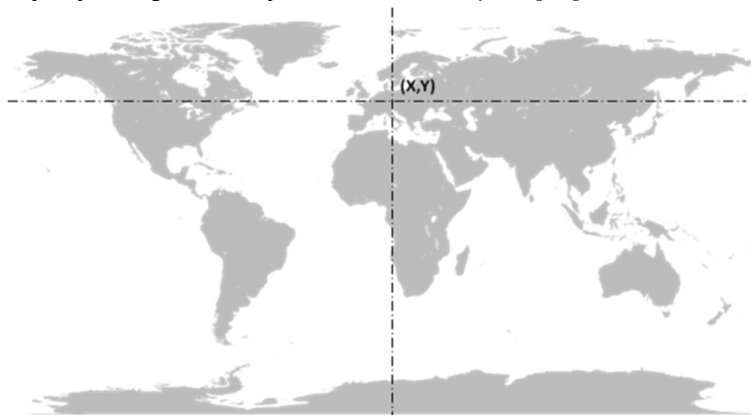
Při disagregaci dochází k obrácenému postupu. Disagregace ale vyžaduje doplňkové informace, protože při agregaci byly některé informace ztraceny. Příkladem takové informace může být schéma výhybek nebo stanice. [13]

1.4.5 Určování polohy

1.4.5.1 Geometrické souřadnice

Polohu bodů v síti je možné určovat pomocí geometrických a geodetických souřadnicových systémů. Nejznámějším příkladem geodetického souřadnicového systému je WGS84, který je také používán systémy Galileo a GPS.

Prvky sítě mohou být zaměřovány za pomoci souřadnicového systému X, Y, Z, tedy zeměpisné šířky, délky a nadmořské výšky. Zjednodušeně lze využívat jen schématických souřadnic X a Y (obr. 20), a to tam, kde není potřeba znát nadmořskou výšku prvku, ale jen pozici prvku. Druhou možností je využití geodetických souřadnic λ , ϕ , h. [15]

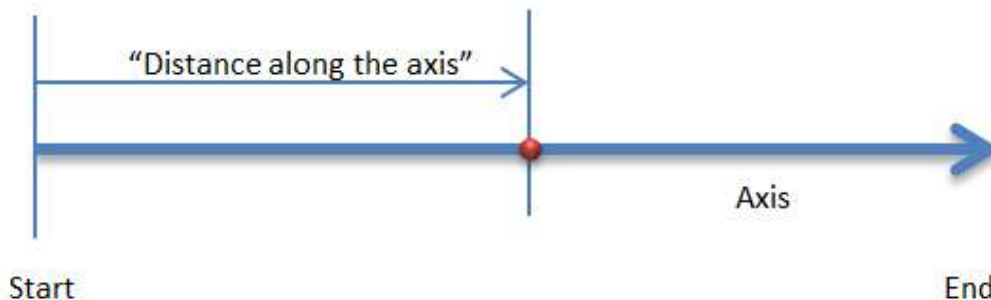


obr. 20: Schematické souřadnice bodu [16]

1.4.5.2 Lineární referenční systém (LRS)

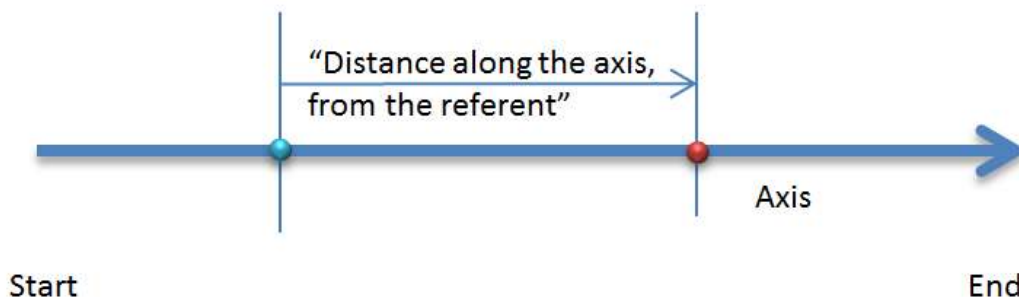
V každodenní praxi je většinou přínosnější vědět, kde se elementy nachází vzhledem ke skutečnému průběhu trati, než znát jejich geometrické a geodetické souřadnice. LRS tedy umožňuje zjišťovat polohu elementu v rámci sítě tak, že se přečte jeho vzdálenost od začátku osy.

LRS používá osu, na které probíhá měření vzdálenosti elementů od jejího počátku. Tato osa může být definována přes libovolný počet elementů. Pro použití LRS je důležité znát délku osy, ale není již nutné znát její geometrii. LRS poskytuje dvě základní metody, kterými lze popsat přesnou polohu podél osy. V absolutním LRS je poloha určena vzdáleností od počátku osy (obr. 21). [17]



obr. 21: Absolutní LRS [18]

Hlavní nevýhodou této metody je nutnost vzít v úvahu při určování polohy celou délku osy. To je nevhodné pro praktické aplikace, kdy je pro určení polohy na trati nutno jít od začátku osy, která může být velmi dlouhá. U relativního LRS je poloha prvků určena vzdáleností od referenčních bodů, jejichž poloha je známá (obr. 22). [17]



obr. 22: Relativní LRS [18]

Relativní LRS je nejčastěji využíván v železničních aplikacích. Pro označení referenčních bodů se používají hektometrovníky. Hlavní výhodou je větší stabilita popisu polohy. Nevýhodou může být nutnost přepočítávat jejich polohy při rekonstrukcích, kdy dochází ke změně délky osy, pokud správce nechce zavést skoky ve staničení. [17]

1.4.5.3 Vnitřní referenční systém (VRS)

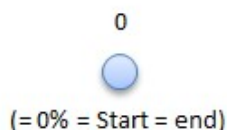
Každý element má svůj vlastní vnitřní referenční systém, který je na něj vázán. Jeho existence je na elementu závislá, tedy vzniká a zaniká spolu s elementem. Jsou pro to dva důvody. Je zajištěno, že každý element má referenční systém, který umožňuje, aby na něm mohly být lokalizovány události jako např. délka výluky koleje z důvodu údržby. Každý element má délku a orientaci. To umožňuje umísťovat k němu další elementy relativně. Umožňuje tedy rozlišit, že například zarážedlo je buď na začátku, nebo na konci daného elementu.

Využívání normalizovaného absolutního VRS umožňuje programovat algoritmy, které nejsou vázány na určité jednotky (metry, kilometry). Normalizované značení (0...1) umožňuje pracovat s konverzí jednotek odděleně.

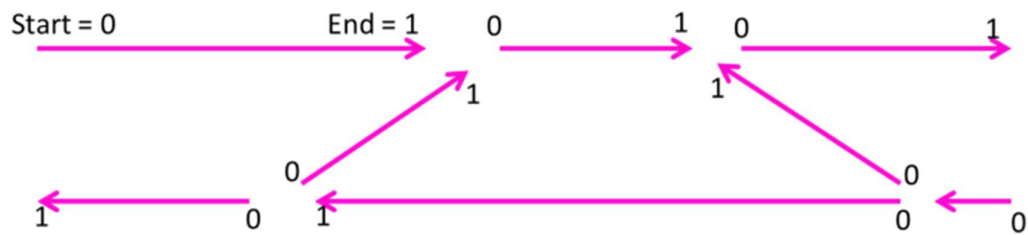
VRS byl vytvořen, aby byl co nejobecnější. Počátku lineárního elementu je přiřazena hodnota 0 a konci hodnota 1 (obr. 23). Nelineárním elementům je přiřazena hodnota 0, protože ukazují na stejné místo v síti (obr. 24). Příklad přiřazování vnitřních souřadnic elementům v síti je možné vidět na obr. 25. [19]



obr. 23: Princip VRS pro lineární element [20]



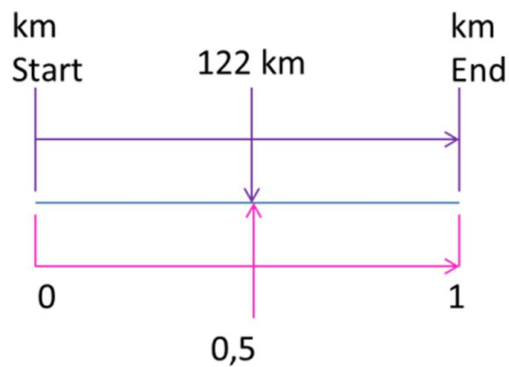
obr. 24: Princip VRS pro nelineární element [20]



obr. 25: Lineární elementy sítě s vyznačenými hodnotami počátku a konce [21]

Poté lze polohu prvků na lineárním elementu určovat jako procentuální vzdálenost od počátku elementu. Tato vzdálenost je udávána v procentech tak, aby se vyhnulo chybám při převodu jednotek. Hlavním smyslem VRS je umožnit univerzální ukládání a přenos pozice, kde dochází k interakci sítě a prvku. Není tedy určen ke správě pozice jako takové.

RTM obsahuje řešení převodu mezi LRS a VRS. Vnější lineární referenční systém může být přiřazen k prvku tak, že se vnější souřadnice LRS přiřadí k odpovídajícím vnitřním souřadnicím VRS. Např. na obr. 26 je vidět, že polovina elementu (bod 0,5) je v lineárním referenčním systému v kilometru 122. [19]



obr. 26: Příklad převodu [16]

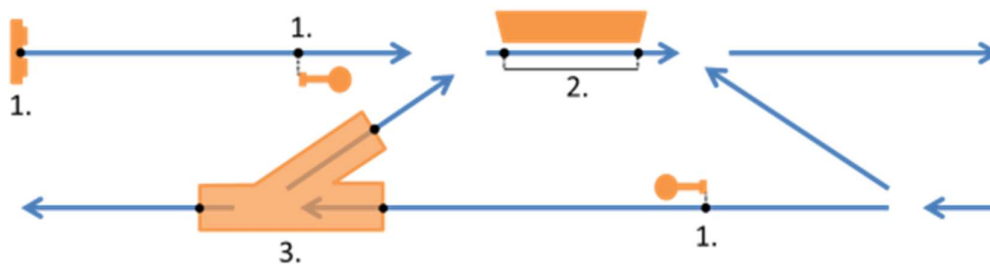
VRS je obtížně použitelný v lidské komunikaci, jelikož se běžně nepoužívá poloha vyjádřená v procentech délky elementu. Běžně se používá LRS, tedy staničení dané trati a určité jednotky. [19]

1.4.6 Umisťování objektů v síti

RailTopoModel umožňuje přidat všem fyzickým a logickým objektům jejich polohu na síti. RTM má jako základ síť. Tato síť je použita jako hlavní referenční systém pro umisťování objektů. Objekty jsou tedy vztaženy k síti tak, že jejich poloha je vázána na polohu elementů. Model je tak tvořen dvěma základními prvky. Síť je složena z elementů a na tyto elementy jsou navázány entity. Entita je jakýkoliv fyzický, nebo logický objekt, který je navázán na element.

Entity mohou být na síti umístěny třemi způsoby (jak je vidět na obr. 27):

1. Bodová lokace – entity jsou lokalizovány jedním bodem (balízy, zarážedla, návěstidla, ...)
2. Lineární lokace – entity jsou lokalizovány podél osy (nástupiště, rychlostní omezení, ...)
3. Plošná lokace – entity jsou lokalizovány dvourozměrnou plochou (výhybky, stanice, kolejové obvody, tunely, mosty, ...) [22]



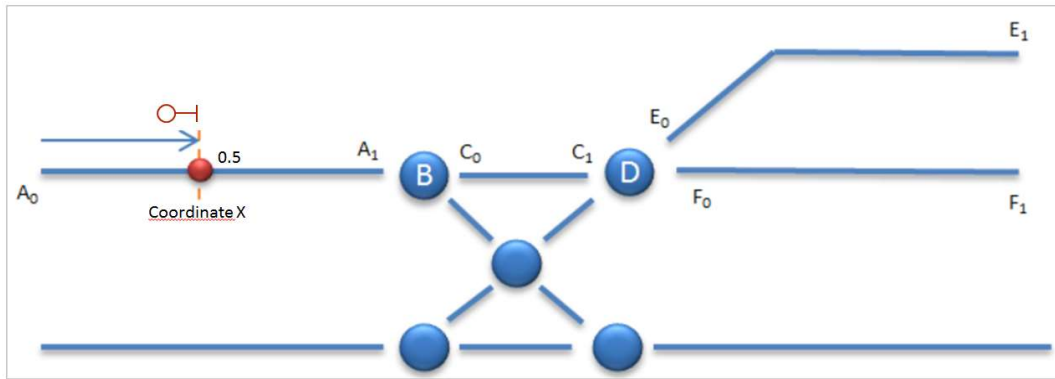
obr. 27: Příklady možných lokalizací entit [23]

1.4.6.1 Bodová lokace

Entita může být v bodové lokaci umístěna buď na nelineární element, nebo podél lineárního elementu v určité vzdálenosti od počátku elementu. Pokud se entita nachází přímo v nelineárním elementu má stejnou polohu jako celý element. Pokud má ale jen jednostranný vztah k nelineárnímu elementu (je např. na konci předchozího lineárního elementu, ale entita už patří k nelineárnímu elementu), je umístěna na jednu z hranic lineárního elementu, ke kterému se vztahuje.

Pokud je entita umístěná podél lineárního elementu v určité vzdálenosti od počátku, je dále nutné popsat její příčnou polohu (vpravo, vlevo, přímo na trati) a směr, pro který platí. Bodová lokace může mít přiřazenou jak geometrickou, tak lineární polohu (staničení), ve které se nachází.

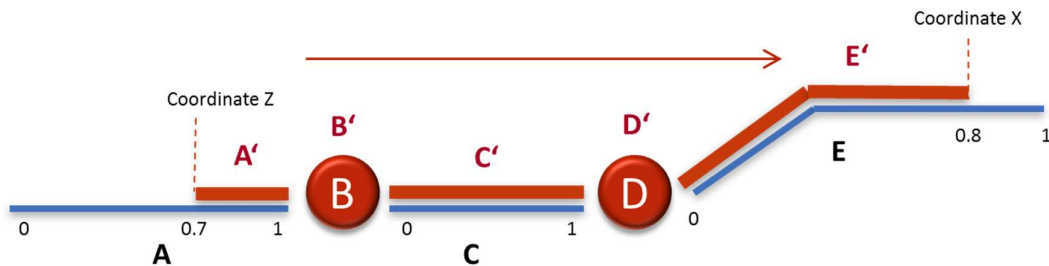
Entita může obsahovat polohu v rámci VRS daného elementu, na kterém se nachází. Na obr. 28 je návěstidlo umístěné v polovině elementu, tedy v bodě s číselnou hodnotou 0,5. V bodě A_0 má VRS hodnotu 0 a v bodě A_1 má VRS hodnotu 1. Normální směr je v tomto případě od A_0 do A_1 . Další informací tedy bude, že návěstidlo platí ve směru opačném k normálnímu směru. Poslední informací je, že se nachází vlevo od elementu při určování podél normálního směru. [22]



obr. 28: Příklad bodové entity umístěné uprostřed elementu A [23]

1.4.6.2 Lineární lokace

Některé entity se dají popsat tak, že se nachází podél tratě (např. nástupiště, kolejové obvody). Lineární lokace je definována jako uspořádaná posloupnost přidružených elementů. To mohou být buď celé elementy (lineární i nelineární), nebo pro lineární elementy jejich části patřící k lokaci dané entity. Začátek a konec přidružených elementů je dán vnitřními souřadnicemi (0...1). Entita má začátek a konec, ty se mohou nacházet buď v nelineárním elementu, nebo jsou to body na lineárním elementu. Dále je nutné znát, zda má lineární entita stejnou orientaci jako přidružený element. Přidružené elementy na sebe musí navazovat, není možné, aby mezi nimi byla mezera. Zároveň musí mít všechny stejnou orientaci. Poté jim lze přiřadit souřadnice v geometrickém, nebo lineárním souřadnicovém systému. Na obr. 29 je příklad lineární entity, v tab. 1 je výčet jejích přidružených elementů v posloupnosti, v jaké jdou za sebou. [22]



obr. 29: Příklad lineární entity [23]

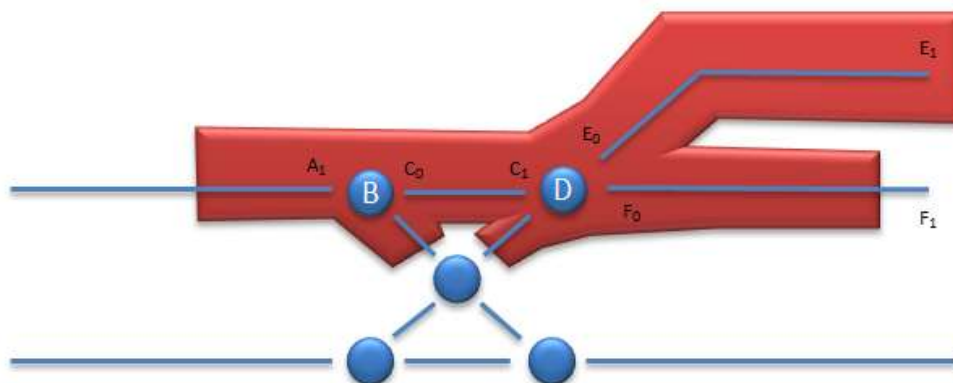
tab. 1: Elementy lineární entity a jejich vnitřní souřadnice [23]

Nr	AssociatedNetElement	netElement	intrinsicCoordBegin	intrinsicCoordEnd	keepsOrientation
1	A'	A	0.7	1	true
2	B'	B	0	0	n/a (nonlinear)
3	C'	C	0	1	true
4	D'	D	0	0	n/a (nonlinear)
5	E'	E	0	0.8	true

V předchozím příkladu jsou všechny elementy orientovány stejně. Pokud by některý element byl orientován opačně, než je normální orientace, musí to být ve výčtu zohledněno. Vezmeme-li jako příklad element A, tak pokud by byl tento element orientován opačně, potom by přidružený element A' měl atribut vnitřní souřadnice počátku (intrinsicCoordBegin) = 0, vnitřní souřadnice konce (intrinsicCoordEnd) = 0,3 a drží orientaci (keepsOrientation) = nedrží (false). [22]

1.4.6.3 Plošná lokace

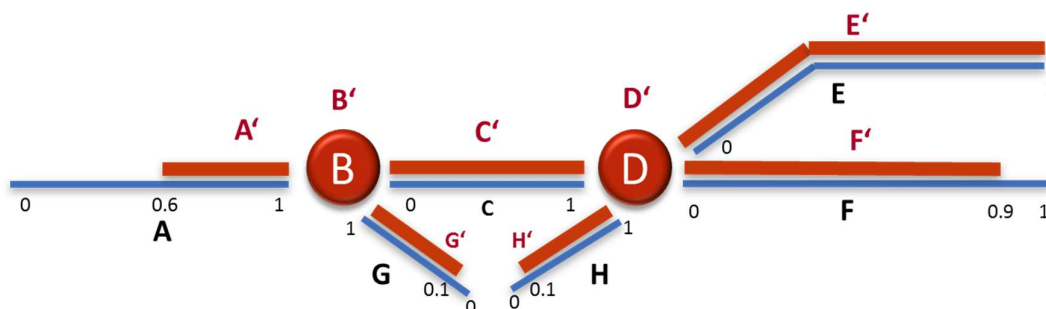
Plošnou lokaci můžeme vyjádřit jako soubor bodových a lineárních elementů v neuspořádané posloupnosti. Příklad plošné lokace je ukázán na obr. 30. [22]



obr. 30: Příklad plošné lokace [23]

Plošná lokace tvoří podgraf původní sítě. Plošnou lokaci je možné vyjádřit podobně jako lineární, tedy výčtem přidružených elementů. Na rozdíl od lineární lokace nemusí být posloupnost uspořádaná a prvky nemusí být orientované shodně. Bodové prvky jsou zahrnuty celé. Lineární jsou buď celé, nebo jejich části, které mají začátek a konec vymezený vnitřními souřadnicemi. Stejně jako u lineární lokace nesmí ani u plošné nastat situace, kdy by byla mezi elementy mezera. Plošná lokace nesmí být rozdělena na několik oddělených částí.

Na obr. 31 je příklad plošné lokace s osmi přidruženými elementy. Je možné pozorovat popis částí přidružených elementů s pomocí vnitřních souřadnic. Také lze vidět na příkladu elementů A a G, že na rozdíl od lineární lokace u plošné nezáleží na orientaci elementů. [22]

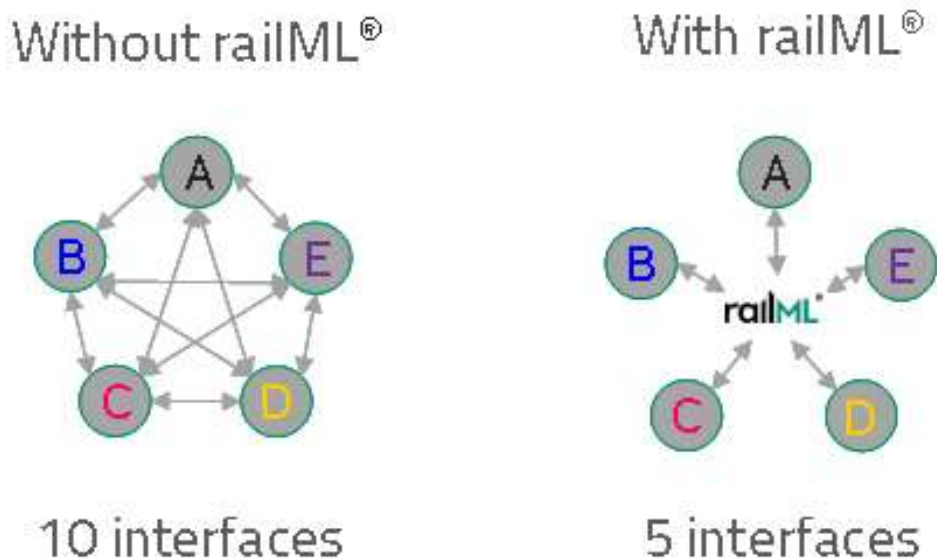


obr. 31: Příklad plošné lokace s podrobným popisem přidružených elementů [23]

2 RailML

2.1 Úvod

Pod názvem railML.org se skrývá iniciativa, která podporuje vývoj železničního značkovacího jazyka railML. RailML je vyvíjený jako open-source. Každý člověk, který má zájem se tedy může do vývoje zapojit. Iniciativa se skládá ze zástupců železničního průmyslu a výzkumných organizací. Aktuální informace o railML je možné dohledat na oficiální webové stránce s názvem „railML 3 Wiki“. Nejaktuálnější verzí je railML 3.1. Záměrem iniciativy je vytvořit řešení na bázi XML pro jednoduchou výměnu dat mezi různými železničními aplikacemi, jak je zobrazeno na obr. 32. Jak je vidět na obrázku vlevo, pro přenos dat mezi pěti různými aplikacemi je nutné vyvinout deset různých rozhraní, tedy rozhraní pro přenos mezi každými dvěma aplikacemi. Na obrázku vpravo je situace s railML. Při použití railML je nutné vytvořit pouze rozhraní mezi aplikací a railML. Do dalších aplikací lze data převést s pomocí dalšího rozhraní mezi jinou aplikací a railML. Při větším množství aplikací dochází k úsporám počtu nutných rozhraní, tedy nákladů na vytvoření těchto rozhraní a jejich údržbu. To má vést ke zjednodušení, a tedy ke zlevnění výměny dat mezi podniky.



obr. 32 Budoucí situace při přenosu dat mezi aplikacemi [24]

RailML je jazyk na základě XML. To znamená, že datový soubor obsahuje nejen data, ale i popis těchto dat. Soubory v XML jsou hierarchické, jsou tvořeny stromovou strukturou. Data jsou v XML dokumentech uspořádána do elementů, které mají atributy (vlastnosti). Elementy začínají a končí tagem. Tag obsahuje jméno elementu uzavřené ve špičatých závorkách (např. <railml>). [24] [25]

2.2 Schémata

railML je rozděleno do pěti základních schémat podle použití:

Infrastruktura (Infrastructure (IS))

Schéma IS slouží pro popis železniční infrastruktury. Obsahuje část pro popis topologie, kdy je železniční síť topologicky popsána jako model uzlu/hrana. Všechny prvky infrastruktury mohou být lokalizovány v libovolném 2- nebo 3- dimenzionálním souřadnicovém systému (např. WGS84). Dále jsou popsány prvky přidružené ke trati (nástupiště, mosty, tunely, balízy apod.) a nehmotné vlastnosti trati (stav trati, rychlostní profily apod.). [26]

Dopravní plány a stavění vlakových cest (Interlocking (IL))

V rámci schématu IL jsou řešeny informace týkající se dopravních plánů a stavění vlakových cest. [26]

Vozidlový park (Rollingstock (RS))

Schéma RS poskytuje možnost ukládání všech dat o železničních vozidlech (lokomotivy, vícedílné jednotky, osobní a nákladní vozy). Další část umožňuje popsat slučování jednotlivých vozidel do formací, a popsat tak celý vlak složený z jednotlivých částí. Toto schéma má sloužit pro management vozidlového parku a pro výpočty jízdních parametrů. [26]

Jízdní řády a oběhy vozidel (Timetable and Rostering (TT))

Schéma TT se používá pro výměnu dat o jízdních řádech a obězích vozidel. [26]

Obecné (Common (CO))

Schéma CO zahrnuje všechny ostatní prvky, které jsou obecné pro všechny předchozí schémata. [26]

2.3 Schéma IS

Tato práce se bude soustředit na infrastrukturní popis prvků železničního systému ve formátu railML. V následující kapitole je popsána základní struktura dokumentu ve formátu railML. Pro obecný popis struktury a část obrázků byl použit dokument "Simple Example" [27]. Ostatní obrázky jsou převzaté z autorem vytvořeného souboru ve formátu railML exportovaného z programu RaiLAIID.

2.3.1 Kostra souboru railML

RailML je postaveno na syntaxi jazyka XML. Základní struktura vypadá podle obr. 33. Atribut *version* zobrazuje verzi, v jaké je dokument vytvořen, v tomto případě je použita verze railML 3.1. Druhá část struktury obsahuje metadata, tedy základní informace o původu dokumentu. Jedná se např. o jméno autora, časovou značku, jazyk, název nebo autorská práva k obsahu. V tagu <common> se nachází obecné vlastnosti, které mohou být použity pro popis nejen infrastrukturní části, ale i ostatních schémat. Prvek <infrastructure> plní roli hlavní schránky, do které jsou vnořeny všechny prvky infrastruktury. [27]

```

<?xml version="1.0" encoding="UTF-8"?>
<railML xmlns="https://www.railML.org/schemas/3.1"
  xmlns:gml="http://www.opengis.net/gml/3.2/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="https://www.railML.org/schemas/3.1
  https://www.railML.org/schemas/railML-3.1/railML3.xsd"
  version="3.1">

  <metadata>
    <dc:format>3.1</dc:format>
    <dc:identifier>2</dc:identifier>
    <dc:source>railML.org</dc:source>
    <dc:title xml:lang="en">Simple Example v11 railML 3.1</dc:title>
    <dc:language>en</dc:language>
    <dc:date>2018-11-05T14:31:00+01:00</dc:date>
    <dc:creator xml:lang="de">Christian Rahmig</dc:creator>
    <dc:rights>Copyright (c) railML.org e.V. Dresden/Germany. All Rights Reserved.
      For more information, visit https://www.railml.org/licence
      Content of this file: railML 3.1 Simple Example</dc:rights>
  </metadata>

  <common>
</common>

  <infrastructure>
</infrastructure>

</railML>

```

obr. 33 Základní tagy v railML [27]

2.3.2 Common

Tag <common> obsahuje vlastnosti vztážené k prvkům ve všech schématech. Na obr. 34 je možné nalézt vnořený tag <organizationalUnits>, který obsahuje podniky, které mají k dané infrastruktuře vztah. Jedná se například o správce infrastruktury, v tomto případě je to „Správa železnic, státní organizace“. Dalším prvkem je tag <positioning>, který obsahuje různé lokalizační systémy pro popis polohy prvků. Poskytuje informace konkrétně o geometrickém, lineárním a obrazovkovém lokalizačním systému. [27]

```

<common id="co_01">
  <organizationalUnits>
    <infrastructureManager id="Správa železnic, státní organizace"/>
  </organizationalUnits>
  <positioning>
    <geometricPositioningSystems>
      <geometricPositioningSystem id="gps01">
</geometricPositioningSystem>
    </geometricPositioningSystems>
    <linearPositioningSystems>
      <linearPositioningSystem id="loc-1" units="km" startMeasure="0" endMeasure="720" linearReferencingMethod="absolute">
</linearPositioningSystem>
    </linearPositioningSystems>
    <screenPositioningSystems>
      <screenPositioningSystem id="sps01" pxX="1024" pxY="768">
</screenPositioningSystem>
    </screenPositioningSystems>
  </positioning>
</common>

```

obr. 34 Obsah tagu common [autor – vystřiženo z railML]

2.3.3 Infrastructure

Popis topologie je obsažen v tagu <topology> (obr. 35). Tento tag obsahuje tři základní vnořené tagy. V tagu <netElements> jsou popsány všechny elementy, ze kterých se síť skládá. Poté relace, tedy vztahy mezi těmito elementy v tagu <netRelations> a poté je popsáno, k jaké síti daný element nebo relace patří v tagu <networks>. V další části je popsána funkční infrastruktura <functionalInfrastructure> na obr. 36, tedy prvky, které jsou přítomny v kolejišti a zajišťují různé funkce. Jedná se například o balízy, zarážedla, nástupiště, návěstidla, výhybky atp. [27]

```
<topology>
  <netElements>
    ...
  </netElements>
  <netRelations>
    ...
  </netRelations>
  <networks>
    ...
  </networks>
</topology>
```

obr. 35 Obsah tagu topology [autor – vystřiženo z railML]

```
<functionalInfrastructure>
  <balises>
    ...
  </balises>
  <borders>
    ...
  </borders>
  <bufferStops>
    ...
  </bufferStops>
  <derailersIS>
    ...
  </derailersIS>
  <platforms>
    ...
  </platforms>
  <signalsIS>
    ...
  </signalsIS>
  <switchesIS>
    ...
  </switchesIS>
  <tracks>
    ...
  </tracks>
  <trainDetectionElements>
    ...
  </trainDetectionElements>
</functionalInfrastructure>
```

obr. 36 Obsah tagu functionalInfrastructure [autor – vystřiženo z railML]

2.3.3.1 Elementy

Element je základní jednotkou sítě. Síť se z nich skládá. Na obr. 37 je možné vidět příklad popisu jednoho lineárního elementu. Jméno elementu je uvedeno v atributu *id*, jméno je v tomto případě ne5. Dále je v tagu <relation> popsán vztah tohoto elementu k ostatním. Tady se jedná o vztah elementu ne5 k elementům ne11 a ne23 v místě výhybky s označením swi20. Druhá část popisuje, v jaké poloze se element nachází. Tag <associatedPositioningSystem> obsahuje informace o pozičním systému přidruženém k elementu. Dojde tedy ke spárování dané vnitřní souřadnice s vnějším referenčním systémem. Jako první je přiřazen počátek elementu, v tagu <intrinsicCoordinate id="ne5_aps01_ic01" intrinsicCoord="0"> je počátku elementu (vnitřní souřadnici „0“) přiřazena hodnota 0 jednotek od počátku osy. Nachází se tak na úplném počátku osy. Ta je zároveň na souřadnicích x = "- 1150,420" a y = "30,000" v geometrickém referenčním systému "gps01". V tagu <intrinsicCoordinate id="ne5_aps01_ic02" intrinsicCoord="1"> dochází ke spárování vnitřní souřadnice „1“ s lineárním referenčním systémem "loc-1", v LRS se tento bod nachází 80 jednotek od začátku osy. V geometrickém referenčním systému s názvem "gps01" je poloha tohoto bodu dána souřadnicemi x = "- 700,426" a y = "30,000". [27]

```
<netElement id="ne5">
  <relation ref="nr_ne5ne11_swi20"/>
  <relation ref="nr_ne5ne23_swi20"/>
  <associatedPositioningSystem id="ne5_aps01">
    <intrinsicCoordinate id="ne5_aps01_ic01" intrinsicCoord="0">
      <linearCoordinate positioningSystemRef="loc-1" measure="0"/>
      <geometricCoordinate positioningSystemRef="gps01" x="-1150.420" y="30.000"/>
    </intrinsicCoordinate>
    <intrinsicCoordinate id="ne5_aps01_ic02" intrinsicCoord="1">
      <linearCoordinate positioningSystemRef="loc-1" measure="80"/>
      <geometricCoordinate positioningSystemRef="gps01" x="-700.426" y="30.000"/>
    </intrinsicCoordinate>
  </associatedPositioningSystem>
</netElement>
```

obr. 37 Obsah tagu netElement [autor – vystřiženo z railML]

2.3.3.2 Relace

Relace popisují vztahy mezi jednotlivými elementy. V atributu *id* je popsáno, mezi jakými elementy určitý vztah existuje a jakým bodem jsou spojeny. Obr. 38 popisuje relaci elementu 5 a 11, která je umístěna ve výhybce číslo 20 ("swi20"). Element s označením A je v tomto případě element 11, element s označením B element s číslem 5. Podle atributů *positionOnX*="..." je možné odhalit, jestli se relace nachází na začátku, nebo na konci elementu. V tomto případě se relace nalézá na začátku elementu A (elementu 11) a na konci elementu B (elementu 5). Atribut *navigability* obsahuje informaci o tom, zda je danou relaci možné používat oběma směry, pouze jedním směrem, nebo ji nelze využívat vůbec. [27]

```
<netRelation id="nr_ne5ne11_swi20" positionOnA="0" positionOnB="1" navigability="Both">
  <elementA ref="ne11"/>
  <elementB ref="ne5"/>
</netRelation>
```

obr. 38 Obsah tagu netRelation [autor – vystřiženo z railML]

2.3.3.3 Sítě

V tagu <network> jsou uloženy elementy a relace jednotlivých rozlišovaných sítí. Jednotlivé sítě mají vlastní identifikátor, kterým se rozlišují. Poté lze v této síti definovat jednotlivé rozlišovací úrovně. Na obr. 39 je zobrazena síť s názvem nw1. Tato síť obsahuje jednu rozlišovací úroveň, a to s názvem lv0. Jedná se o popis v úrovni „Micro“. Tagy <networkResource> obsahují všechny základní prvky, které síť na dané úrovni popisu obsahuje, tedy jednotlivé elementy a relace. [27]

```
<network id="nw01">
  <level id="lv0" descriptionLevel="Micro">
    <networkResource ref="ne5"/>
    <networkResource ref="ne11"/>
    <networkResource ref="ne16"/>
    ...
    <networkResource ref="nr_ne5ne11_swi20"/>
    <networkResource ref="nr_ne11ne23_swi20"/>
    <networkResource ref="nr_ne11ne18_swi28"/>
    ...
  </level>
</network>
```

obr. 39 Obsah tagu network [autor – vystřiženo z railML]

2.3.3.4 Prvky funkční infrastruktury

V tagu <functionalInfrastructure> jsou vnořeny tagy prvků funkční infrastruktury. Jedná se o všechny prvky, které se nachází v kolejišti a nepatří do topologie. Na obr. 40 je vidět základní struktura popisu každého elementu. Nejvyšší tag v hierarchii obsahuje název typu elementu spolu se základními atributy. Vnořené tagy mohou obsahovat jméno i s jazykem, ve kterém toto jméno je, a polohový tag, který obsahuje informaci o tom, zda má entita bodovou, lineární, nebo plošnou lokaci. V příkladě na obr. 40 je vidět popis entity balíza s *id* bal44, jejíž označení je 27. Má bodovou lokaci na elementu ne5, platí pro směr opačný ke staničení a má vnitřní souřadnici 0,2397. V tagu <linearCoordinate> je zobrazené propojení jejího VRS a vnějšího LRS. V tomto případě se jedná o propojení s LRS s názvem loc-1, balíza se nachází vpravo ve směru staničení a je v bodě 0 osy, tedy na úplném začátku osy. [27]

```
<balise id="bal44" type="transparent" isBaliseGroup="false">
  <name name="27" language="en"/>
  <spotLocation id="bal44_loc-1" netElementRef="ne5" applicationDirection="reverse" intrinsicCoord="0.2397">
    <linearCoordinate positioningSystemRef="loc-1" lateralSide="right" measure="0"/>
  </spotLocation>
</balise>
```

obr. 40 Obsah tagu balise [autor – vystřiženo z railML]

3 ERSA operační simulátor

Z důvodů postupného přechodu evropských států na systém ERTMS/ETCS se objevila příležitost pro softwarové výrobce naprogramovat systémy, na kterých bude možné simulovat funkci ERTMS/ETCS v online prostředí. Tyto simulace umožňují navrhovat, simulovat a ověřit návrh konkrétního řešení systému před jeho fyzickou implementací. Tím může dojít k odstraňování chyb v návrhu ještě před fyzickou implementací a k úspoře nákladů.

Úkolem práce je zjistit, zda je možné použít tento software pro popis infrastruktury v DSFD a zda je dále možné tato data exportovat do univerzálně použitelného formátu.

3.1 ERTMS/ETCS

Evropský systém řízení železničního provozu ERTMS/ETCS je dalším vývojovým stupněm zabezpečení železničních tratí v Evropě. ERTMS se skládá ze dvou částí. První je ETCS, který má palubní a traťovou část a zajišťuje bezpečnou jízdu vlaku. Další částí je GSM-R, který zajišťuje datovou a hlasovou komunikaci. Důležitou součástí jsou také provozní pravidla.

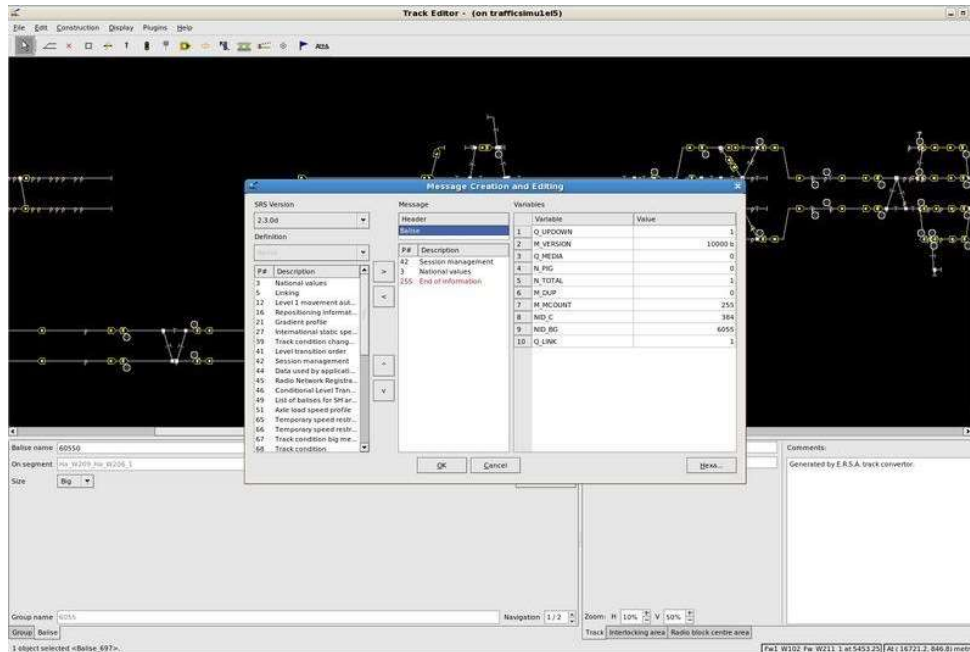
Použitím tohoto systému dojde k zajištění interoperability evropské železniční sítě. Systém je určen i pro použití ve vysokých rychlostech, zvyšuje kapacitu dráhy a zvyšuje bezpečnost provozu oproti většině aktuálně využívaným zabezpečovacím systémům. V budoucnu přispěje ke snížení nákladů dopravců, kteří nebudou muset své lokomotivy vybavovat několika národními zabezpečovači, ale pro pohyb po Evropě jim bude stačit pouze palubní část ETCS. Správci infrastruktury budou moci využívat služeb více dodavatelů traťové části ETCS. [28]

3.2 ERSA

ERSA je francouzský online simulátor ERTMS/ETCS. Součástí balíčku je komplexní soubor programů pro ověřování funkčnosti návrhů ERTMS/ETCS. Pro potřeby práce byla prozkoumána funkce dvou programů. Jednalo se o ERSA Track Editor a ERSA ERTMS/ETCS Operational Simulator. [29]

3.2.1 Nástroj na vytváření infrastruktury

ERSA Track Editor umožňuje vytvářet fyzickou infrastrukturu. Podporuje kreslení kolejí, ke kterým lze dále přidat jejich vlastnosti, tedy sklon nebo poloměr oblouku. Dále lze na koleje umisťovat objekty, a to jak bodové, tak plošné. Poté se objektům přidají vlastnosti. V nových verzích je možné přidávat obsahy zpráv umístěné v balících (obr. 41). [30] Ve verzi nalézající se v DSFD je tato část oddělena a nalézá se v programu Scenario Editor.



obr. 41: Ukázka úpravy obsahu zpráv balíz v Track Editoru [30]

3.2.2 Nástroj na simulaci

Po vytvoření infrastruktury lze daný návrh otestovat v reálném čase pomocí ERSA ERTMS/ETCS simulátoru. Vybere se soubor určený k testování a spustí se. Proběhne SoM, kde se vyplní nutné údaje, a poté lze zahájit jízdu. Simulátor potom nabízí všechny funkce, kterými disponuje i skutečné ETCS. Tedy monitoruje rychlost, počítá brzdné křivky, zpracovává přijaté MA a další zprávy atp. Na konci je vytvořen diagnostický soubor, který lze dále analyzovat. Na obr. 42 je zobrazen HMI v prostředí simulátoru, dále ovládací prvky a boční pohled na simulovaný vlak. [31]



obr. 42: Ukázka HMI a ovládacích prvků v prostředí ERSA simulátoru [31]

3.2.3 Formát ukládání dat

ERSA používá pro ukládání a správu souborů databázi MySQL. Program obsahuje i vlastního správce databází. Verze softwaru ERSA, která se nachází v dopravním sálu, neumožňuje export ani import dat ze souboru ve standardizovaném formátu railML. Jedná se o verzi 4.14.0.r1204 z roku 2017. Novější verze softwaru možnosti exportu a importu do a z railML již, podle stránek výrobce [30], podporují. Zároveň je možné v této nové verzi exportovat i do formátu SUBSET-112 využitelného pro ETCS aplikace. [30]

3.3 Závěr

ERSA nacházející se v DSFD nabízí kvalitní nástroj na základní modelování infrastruktury a simulaci řízení podle ERTMS/ETCS. Neumožňuje ale export dat ve formátu railML. Jelikož je konečným cílem popisu dopravní infrastruktury v DSFD použití těchto dat i v jiných aplikacích, tak se použití tohoto vydání softwaru ERSA bez možnosti exportu nejeví jako efektivní. ERSA ale může být využívána pro výuku návrhu a funkce infrastrukturní části ETCS. Popis infrastruktury ale bude muset být pro tento program vytvořen samostatně.

4 Návrh a implementace v DSFD

4.1 Možnosti využití různých popisů v DSFD

Datový popis kolejíště může v budoucnu posloužit pro různé aplikace, např. jako základní data pro ETCS. Pro účely popisu infrastruktury v DSFD bylo ručně odměřeno několik částí sálu tak, aby model v budoucnu obsahoval skutečné a co nejpřesnější hodnoty. Část těchto dat byla uložena do tabulky v Excelu, část byla použita pro nakreslení části kolejíště v AutoCADu. V případě, že se tento způsob ručního měření prokáže jako nejlepší řešení, je možné ho v budoucnu využít jako základní způsob získávání přesných dat o infrastruktuře.

Nejdříve bylo zvažováno použít pro popis kolejíště softwarového nástroje ERSA Track Editor. V předchozí kapitole bylo vysvětleno, proč není použití softwaru ERSA dostupného v sále efektivním řešením. Jedná se hlavně o nemožnost exportovat a importovat data v univerzálně přenositelném formátu, protože data jsou uložena pouze ve vlastní databázi. V případě nových verzí softwaru ERSA Track Editor export a import dat v railML možný je, pokud by se tedy dostala do DSFD novější verze programu, bylo by vhodné možnost použití tohoto programu znovu ověřit.

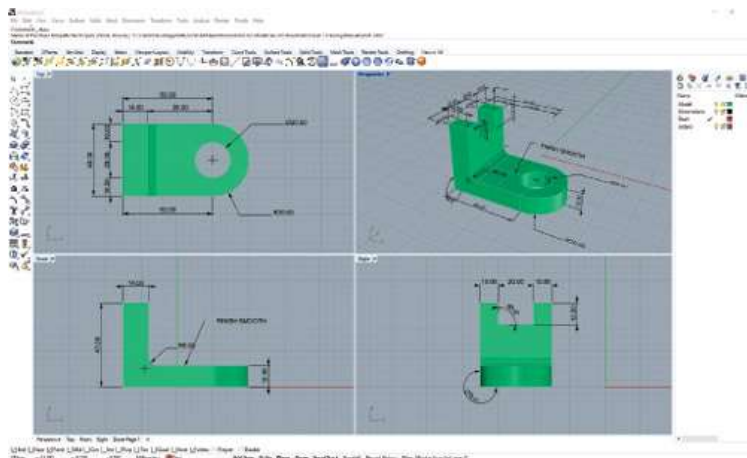
Na druhou stranu se popis kolejíště pomocí metodiky RTM/railML jeví jako dobrý základ popisu, a to pro svou univerzální přenositelnost. Nicméně při zapisování přímo v textové podobě by pravděpodobně došlo k mnoha chybám a překlepům. Bylo tedy nutné nalézt program, který je schopný exportovat a importovat soubory ve formátu railML. Díky tomu, že se railML již několik let vyvíjí se už začínají objevovat programy, které s formátem railML dokážou pracovat.

Jako vhodné se jeví využití programu RailAiD Desktop [32], který je v DSFD také k dispozici a má podobné rozhraní jako ERSA Track Editor. Nabízí nejen popis infrastruktury v grafickém rozhraní, ale navíc umožňuje definovat a nakreslit vlastní entity. Tyto entity se kreslí a definují v přidruženém programu s názvem RailAiD Object Modeler [33]. Při popisu tedy není nutné spoléhat na předem vytvořenou knihovnu, která neobsahuje všechny typy entit vyskytující se na železniční síti. Příkladem mohou být zvláštní typy proměnných návěstidel nebo neproměnná návěstidla s českým textem. Těmto entitám lze poté přiřazovat důležité vlastnosti. Na vytvořenou síť lze umisťovat objekty a přiřazovat jim jedno nebo i více staničení. Poté lze data exportovat do souboru ve formátu railML. Pokud je cílem získat univerzální datový popis infrastruktury v railML, považuji za vhodné rozvíjet právě toto řešení. Následující kapitola ukáže možnosti jeho využití.

V budoucnu by se mohlo využívat modelování v programu Rhinoceros, který nabízí modelování kolejíště ve 3D. Použití tohoto programu závisí na možnosti importovat a exportovat z něj data tak, aby se dala využívat i v jiných programech. V následující kapitole je tento program popsán podrobněji.

4.1.1 Rhinoceros

Rhinoceros je placený program sloužící k modelování ve 3D. Umožňuje vytvářet, editovat a animovat objekty jako křivky, plochy, nebo trojrozměrné objekty. Byl vytvořen firmou Robert McNeel & Associates. Jako většinu ostatních grafických programů ho je možné ovládat s pomocí vestavěné příkazové řádky. Rhinoceros pracuje s geometrií založenou na matematickém modelu, umožňuje tedy modelovat složité 3D tvary. Pro program existuje mnoho různých pluginů, které jsou vytvářeny širokou sítí vývojářů. Hlavní grafické rozhraní je v základu rozloženo do čtyř obrazovek. Umožňuje současný horní, přední a pravý pohled. Čtvrtá obrazovka obsahuje perspektivní pohled. Hlavní grafické rozhraní programu je vidět na obr. 43. [34]

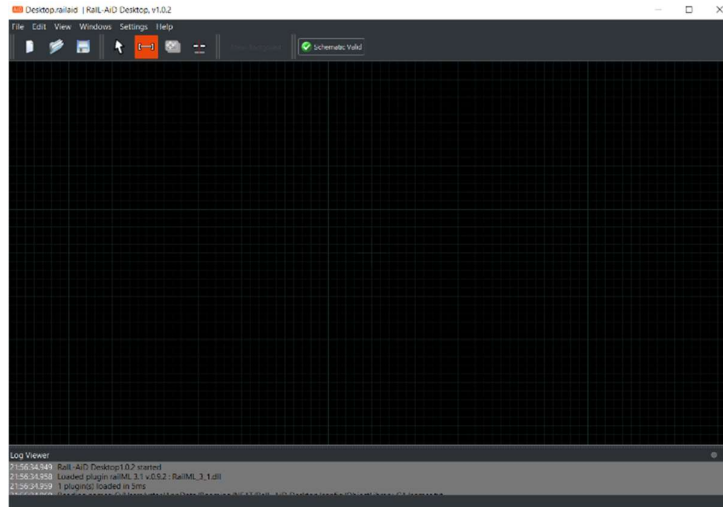


obr. 43: Grafické rozhraní programu Rhinoceros 6 [34]

Rhinoceros podporuje import a export mnoha druhů formátů. Jedná se i o soubory, které jsou vytvořeny v AutoCADu. [35] Jelikož je kolejiště v DSFD již částečně popsáno v AutoCADu, nebude nutné při přechodu na Rhinoceros modelovat tuto již vymodelovanou část znovu. Grafické rozhraní v Rhinoceru je jednoduché a intuitivní, při použití v DSFD by mělo být jeho používání jednoduché pro všechny, kteří již dříve pracovali v nějakém programu jako AutoCAD. Zároveň Rhinoceros obsahuje možnost rozšíření o různé pluginy, které mohou usnadnit práci.

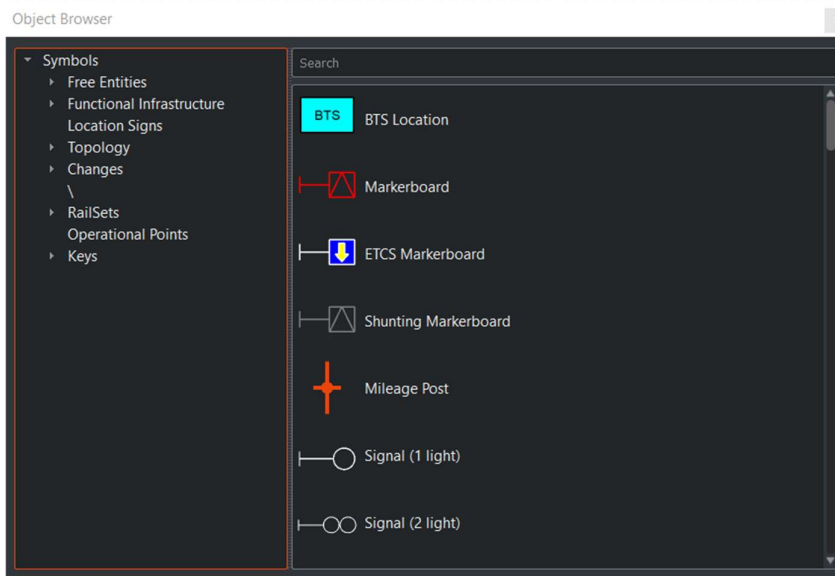
4.2 RailAiD

Tato kapitola slouží k pochopení práce a ovládání v grafickém prostředí programu. RailAiD je nástroj, který umožňuje návrh železniční infrastruktury a jejích prvků v grafickém prostředí. Poté je možné tento návrh exportovat do různých formátů jako railML, CSV, nebo PDF. Program zároveň kontroluje návrh a upozorňuje na případné základní chyby v návrhu. Program má modulární architekturu, je tedy jednoduše rozšiřitelný. Obrázky použité v této kapitole jsou vystřižené přímo z grafického prostředí RailAiD. Ovládání je podrobněji popsáno v manuálu. [32] Na obr. 44 je hlavní okno programu, ve kterém probíhá modelování infrastruktury.



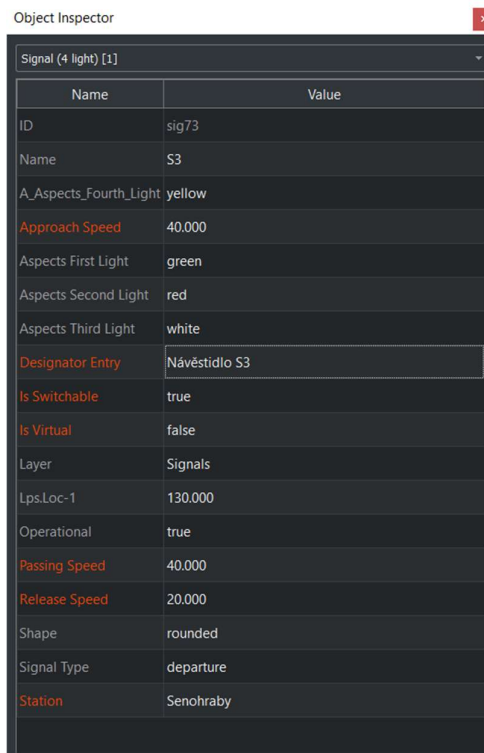
obr. 44: Grafické rozhraní v RaiLAiDu [autor – vystřiženo z RaiLAiDu]

Na obr. 45 je zobrazena knihovna prvků, které lze do modelu vložit. Prvky se načítají z externí knihovny. Tuto knihovnu je možné rozšiřovat o další prvky.



obr. 45: Vyhledávač objektů v RaiLAiDu [autor – vystřiženo z RaiLAiDu]

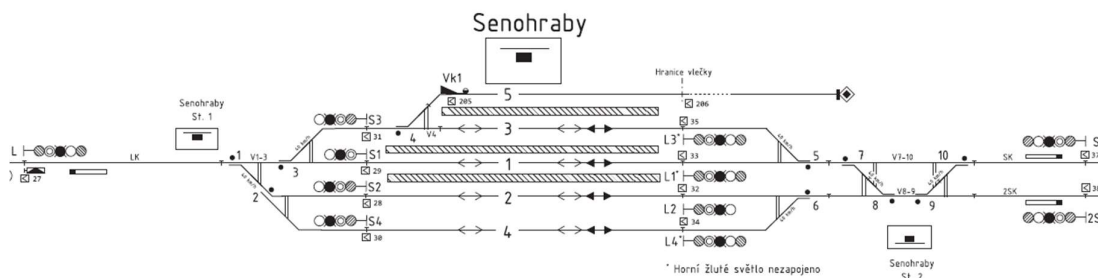
Na obr. 46 je zobrazeno okno “Object Inspector“, které umožňuje nastavovat a upravovat vlastnosti objektů přidanych do schématu. [32]



obr. 46: Nastavování vlastností objektů v RaiLaiDu [autor – vystřiženo z RaiLaiDu]

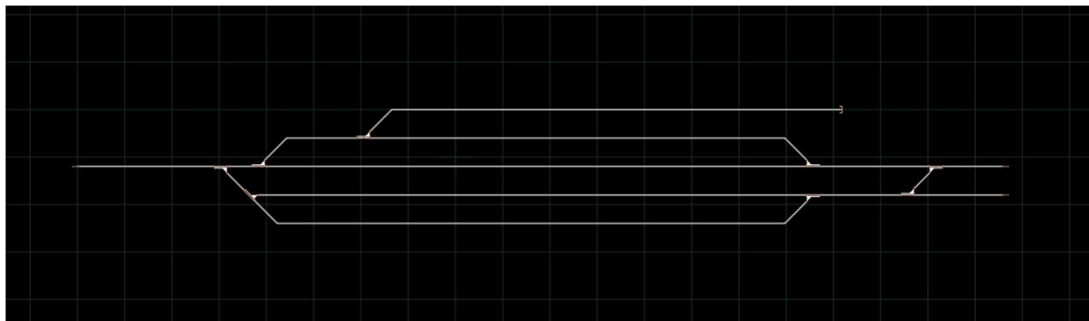
4.3 Popis stanice Senohraby a přilehlého úseku v grafickém rozhraní

V rámci implementace bude jako příklad možného datového popisu železničního systému Dopravního sálu FD navržena stanice Senohraby podle metodiky RTM/railML. Pro popis kolejiště bude použit program RaiLaiD, ze kterého je možné exportovat soubor ve formátu railML. Jako základ byl s dovolením autora použit plánek kolejiště od Ing. Petra Kouteckého. Jedná se o interní materiál, který zobrazuje schéma kolejiště. [36] Originál je k nahlédnutí u autora, nebo v DSFD. Pro účely práce byla použita část zobrazující stanici Senohraby, jak je vidět na obr. 47.



obr. 47: Schéma stanice Senohraby [36]

Schéma stanice bylo následně překresleno v grafickém rozhraní programu. Koleje se kreslí tak, že se označí požadované pole pro kreslení kolejí. Myší se označí požadovaný počátek koleje a poté požadovaný konec koleje. V základu se na obou koncích kolejí vytvoří zarážedla. Pokud má kolej další pokračování lze zarážedlo vyměnit za hranici, která definuje, že kolej zde dále pokračuje, ale už není popisována v rámci tohoto schématu. Stanice Senohraby obsahuje čtyři staniční koleje, ze stanice odbočuje jedna vlečková kolej, která na svém konci obsahuje zarážedlo. Nachází se mezi stanicemi Čerčany a Strančice. Od Čerčan vede do stanice jedna kolej, od Strančic vedou dvě koleje. Tyto koleje jsou, jak bylo popsáno výše, zakončeny hranicí modelovaného úseku. Schéma stanice v grafickém rozhraní je na obr. 48. Po nakreslení základního schématu je možné přidávat prvky nacházející se v kolejšti.

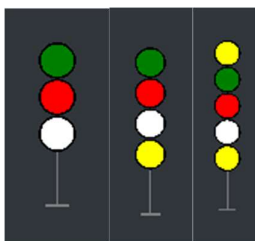


obr. 48: Schéma stanice Senohraby v RailAiDu [autor – vystřiženo z RailAiDu]

Knihovna prvků dodávaná s programem neobsahovala všechny prvky, které jsou v kolejšti v DSFD. Proto byly tyto prvky nově vytvořeny, případně dodatečně upraveny v programu Object Modeler. Ovládání se řídí Object Modeler manuálem. [33] Vymodelovány nebo upraveny byly dodatečně tyto prvky: přepínatelná balíza na obr. 49, tři typy proměnných návěstidel na obr. 50 a označník na obr. 51.



obr. 49: Zobrazení přepínatelné balízy [autor – vystřiženo z RailAiDu]

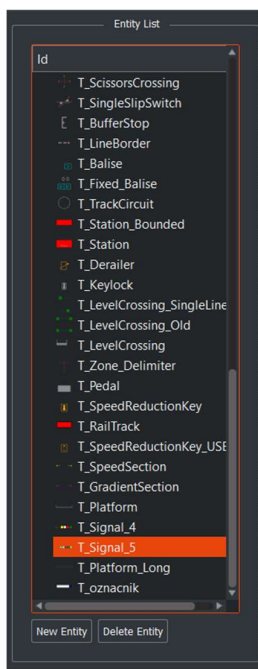


obr. 50: Zobrazení třech typů proměnných návěstidel [autor – vystřiženo z RailAiDu]



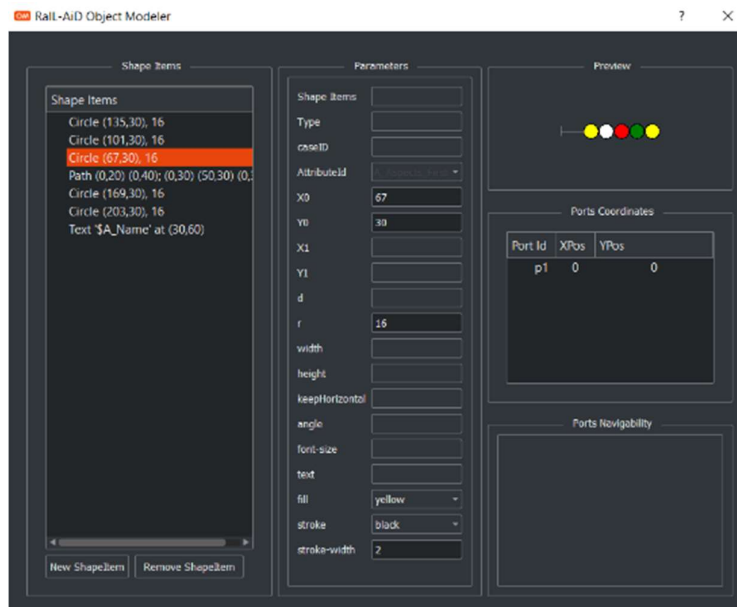
obr. 51: Zobrazení označníku [autor – vystřiženo z RailAiDu]

Proces modelování v Object Modeleru byl následující. Nejdříve byla založena nová entita v tabulce na obr. 52. Při zakládání bylo entitě přiřazeno jméno a prefix. Prefix je na začátku názvu každé nově vytvořené entity, poté při vytvoření následuje pořadové číslo.



obr. 52: Tabulka entit [autor – vystřiženo z RaiLAI Du]

Dále je s pomocí programu vytvořena podoba jednotlivých entit (obr. 53). Podobu je možné sestavit z různých nabízených tvarů (úsečky, kruhy, obdélníky atp.). Nastaví se barva jednotlivých prvků, jejich poloha v rámci entity, tloušťka čar a místo, kam se zobrazí název entity. Je možné vytvořit přepínač, který změní barvy, nebo tvary entity podle nastavených vlastností.



obr. 53: Tabulka tvarů [autor – vystřiženo z RaiLAI Du]

Dále se vytvoří atributy, kterým půjde v grafickém rozhraní přiřazovat hodnoty nad rámec skupiny entit, mezi které daná entita patří. Zde se jedná o atributy proměnného návěstidla s pěti světly (obr. 54). Zatímco poloha je společná pro všechny návěstidla, tak návěstidlu s pěti světly musí být přiřazeny atributy, které umožňují v grafické verzi popsat, jaké jsou barvy jednotlivých světel. Obecně se vkládá se jméno atributu, typ atributu a základně nastavená hodnota atributu. Mezi typy atributů, které lze vložit patří: text, číslo, nebo různé přednastavené hodnoty.

Id	Type	Default
A_Aspects_First_Light	string	
A_Aspects_Second_Light	string	
A_Aspects_Third_Light	string	
A_Aspects_Fourth_Light	string	
A_Aspects_Fifth_Light	string	
A_Shape	E_signalShape	V_rounded

obr. 54: Tabulka vytváření atributů [autor – vystřiženo z RaiLAIdu]

Všem prvkům byly poté v rámci zasazování do schématu jednotlivě přiřazeny jejich vlastnosti a poloha. Jednalo se celkově o sedmdesát pět entit devíti různých druhů. Příklad, jak byly vlastnosti přiřazovány, je vidět na obr. 55. V tomto případě se jednalo o návěstidlo s pěti světly (obr. 50 vpravo). To se ve stanici nachází celkem šestkrát. Z toho třikrát jako vjezdové a třikrát jako odjezdové. Bylo jim postupně přiřazeno jméno, barvy jednotlivých světel a různé rychlosti, kterými se lze v okolí návěstidel pohybovat. Dále bylo možné rozhodnout, zda se jedná o proměnné, nebo neproměnné návěstidlo a zda je, či není virtuální. Příklad na obr. 55 také obsahuje informaci o typu návěstidla, v tomto případě šlo o odjezdové návěstidlo ve stanici Senohraby. Dále se ve stanici nachází čtyři návěstidla se čtyřmi světly (obr. 50 uprostřed). a jedno návěstidlo se třemi světly (obr. 50 vlevo). Všem návěstidlům byly, po vložení do schématu, postupně přiřazeny vlastnosti. Celkově se jednalo o jedenáct proměnných návěstidel.

Ve schématu se nachází celkem deset výhybek těm byla nejdříve přiřazena poloha. Dále také rychlosti, kterými lze výhybku pojíždět v různých směrech, čas, za který je výhybka schopna změnit polohu výměny, zda je funkční a je tedy možné jí používat, zda se vrací do původní polohy samostatně, a nakonec o jaký typ výhybky se jedná.

Poté bylo doplněno dvanáct balíz. Mimo hlavní společné vlastnosti, jako je poloha, je možné rozhodnout, zda se jedná, nebo nejedná o balízovou skupinu. Případně zda je pevná, nebo přepínatelná.

Dále jedno zarážedla, o kterém lze rozhodnout, jestli je funkční, kde se nachází a jakého je typu.

Ve stanici se podle schématu nacházejí tři nástupiště s pěti nástupištními hranami. Jelikož program zatím neumožňuje připojovat jeden objekt k více kolejím najednou, byly nástupiště rozděleny a popsány podle svých nástupních hran. Bylo tedy popsáno pět objektů třídy nástupiště. Vlastnostmi, které umožňuje program přidat jsou výška nástupní hrany, délka nástupiště, poloha středu nástupiště, zda je funkční a zda platí pro oba směry jízdy.

Při jízdě z vlečky se na kolejišti, jako ochrana nachází jedna výkolejka. Ve vlastnostech je možné definovat, kde se nachází, zda je zamčená, jak rychle je schopna se převést do polohy, kdy je možné jí bezpečně přejet, zda je funkční, jaký je její základní stav a na jakou stranu je vykolejení případně ujetého vozidla nastaveno.

Stanice obsahuje kolejové obvody pro určování obsazení jednotlivých kolejí vlaky. Kolejových obvodů je ve schématu celkem třináct. Je možné uvést jejich délku, zda jsou funkční a lze nastavit jejich zpoždění.

Dále osmnáct izolovaných styků, u nichž je uvedena jejich poloha, zda jsou funkční a jakého jsou typu.

Posledním prvkem, který byl v rámci vytváření modelu v rámci práce umístěn jsou neproměnná návěstidla. Jedná se o jednu značku pro posun a tři označnický. Těm byla přiřazena poloha a název.

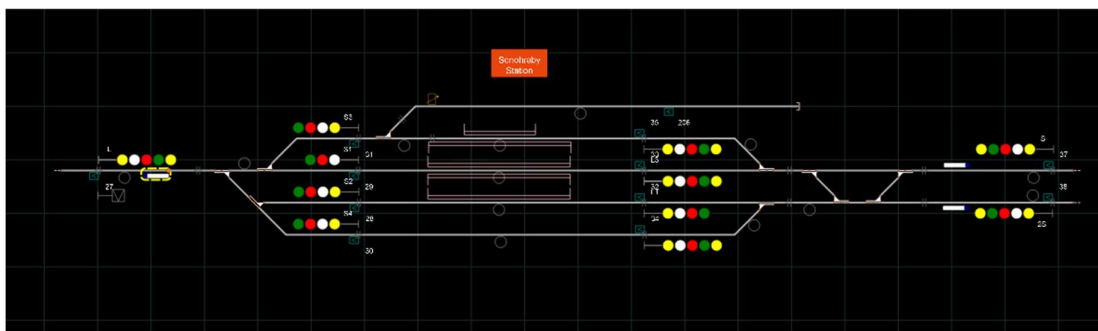
Všechny tyto informace jsou obsaženy v exportovaném souboru v railML v Příloze č. 1.



Name	Value
ID	sig79
Name	L3
A_Aspects_Fifth_Light	yellow
A_Aspects_Fourth_Light	white
Approach Speed	40.000
Aspects First Light	yellow
Aspects Second Light	green
Aspects Third Light	red
Designator Entry	Odjezdové návěstidlo
Is Switchable	true
Is Virtual	false
Layer	Default
Lps.Loc-1	400.000
Operational	true
Passing Speed	40.000
Release Speed	20.000
Shape	rounded
Signal Type	departure
Station	Senohraby

obr. 55: Vlastnosti proměnného návěstidla [autor – vystřiženo z RaiLAI Du]

Následující obrázek (obr. 56) zobrazuje konečnou podobu stanice Senohraby nakreslenou v grafickém rozhraní programu RaiLAiD. Byly zapracovány všechny prvky tak, jak jsou zaznamenány v použitém podkladu. V Příloze č. 2 je uloženo schéma této stanice ve větším rozlišení v PDF a v Příloze č. 3 je schéma přímo ve formátu RaiLAiD.



obr. 56: Konečná podoba stanice Senohraby v grafickém rozhraní [autor – vystřiženo z RaiLAiDu]

4.4 Popis stanice Senohraby v railML

Po dokončení grafického návrhu následuje export dat ve formátu railML. V Příloze č. 1 je uložen celý popis této stanice v railML, s tímto souborem lze tedy už dále pracovat a využívat ho pro různé aplikace. Tento soubor obsahuje všechny prvky popisu tak, jak je popsáno v teorii k railML. Základní informace o dokumentu jsou na obr. 57. Metadata obsahují hlavně název stanice, datum exportu, jméno autora, zdroj, ze kterého data pocházejí, a informace o autorských právech.

```
<metadata>
  <dc:title>Senohraby_stanice_railml</dc:title>
  <dc:date>2020-08-04T17:59:08Z</dc:date>
  <dc:creator>Tomáš Staný</dc:creator>
  <dc:source>Rail-AID V 1.0.2</dc:source>
  <dc:identifier>1</dc:identifier>
  <dc:subject>railML.org</dc:subject>
  <dc:format>3.1</dc:format>
  <dc:description>Schéma stanice Senohraby</dc:description>
  <dc:publisher>This file has been generated using Rail-AID framework, a NEAT S.r.l. Intellectual property. (http://www.rail-aid.com/) (c) NEAT S.r.l., Rome, Italy (http://www.neat.it/)</dc:publisher>
</metadata>
```

obr. 57: Metadata souboru se stanicí Senohraby [autor – vystřiženo z railML]

V souboru jsou také zahrnuty základní tagy common, infrastructure a interlocking. Tag infrastructure dále obsahuje základní tagy topology, functionalInfrastructure a infrastructureVisualizations. Do nich jsou vnořeny další tagy tak, jak je popsáno v teorii ke struktuře railML dokumentů.

V modelu bylo popsáno celkem sedmáct elementů, deset výhybek, jedenáct proměnných návěstidel, dvanáct balíz, jedno zarážedlo, pět nástupištních hran, jedna výkolejka, osmnáct izolovaných styků, čtyři neproměnná návěstidla a třináct kolejových obvodů.

4.5 Závěr

Grafické rozhraní pro modelování je intuitivní a jednoduché. Obsahuje funkce pro usnadnění modelování, které zabraňují některým chybám. Object Modeler má zatím stále nevýhody. Jako zásadní nedostatek se jeví nemožnost kopírovat jednotlivé prvky a dále je upravovat. Musí se tedy tvořit znovu, což je zdlouhavé. Dalším nedostatkem je nutnost rozdělit jedno nástupiště mezi dvěma kolejemi na dvě, jelikož program neumí připojovat prvek jako nástupiště ke dvěma kolejím najednou.

I přes nedostatky v ovládání, které RaiLaiD zatím má, je to celkově software, který dokáže popsat infrastrukturu a poté ji vyexportovat v railML a po drobných úpravách použít. Lze v něm navrhnout i vlastní entity, což je výhoda, protože základní knihovna neobsahuje všechny prvky vyskytující se v praxi. Pro výuku a základní popis lze program doporučit. Při použití na velkých schématech ale chybí prvky, které by mohly navrhování urychlit.

Závěr

Prvním cílem práce bylo seznámení se s popisem infrastruktury ve formátu RailTopoModel. V první kapitole se čtenář dozví, že RTM má sloužit jako budoucí univerzální základ pro popis železniční infrastruktury. Tento přístup má v budoucnosti přinést zjednodušení a zlevnění přenosu dat mezi různými aplikacemi. Nicméně provedená studie proveditelnosti upozorňuje na možné hrozby pro tento nový formát. Jde hlavně o nedostatečné financování a politickou podporu. RTM popisuje infrastrukturu s použitím definice souvislého grafu. Základní model sítě se tak skládá z vrcholů a hran. Vrcholy grafu jsou jednotlivé elementy a hrany představují relace mezi nimi, tedy jejich vzájemné vztahy. Lze v něm popsat infrastrukturu na několika úrovních. Ty si lze samostatně vytvořit. Existují nicméně tři základní (Micro, Meso a Macro). V další části práce je popsána struktura RTM, tedy jaký je rozdíl mezi topologickou a hierarchickou strukturou. Topologická struktura popisuje vztah prvků a dané úrovní popisu. Hierarchická struktura se zabývá popisem vztahů elementů na jednotlivých úrovních popisu. Zároveň jsou popsány možnosti přechodu mezi jednotlivými úrovněmi. Jedná se o agregaci a disagregaci. Dále byly prozkoumány možnosti, jak lze jednotlivé body v modelu lokalizovat. Jednalo se hlavně o geometrický, lineární a vnitřní referenční systém. Poslední věcí byla možnost seznámit se s tím, jak je možné v modelu umisťovat entity. Ty mohou být bodové, lineární nebo plošné.

V druhé kapitole, věnující se railML, byly sděleny základní informace k formátu. Jedná se o jazyk na základě XML vyvíjený jako open-source. V budoucnosti má sloužit k přenosu dat mezi aplikacemi tak, aby bylo nutné vytvářet méně rozhraní. Tím má dojít ke zlevnění těchto přenosů dat. Obsahuje pět základních schémat (Infrastruktura, Dopravní plány a stavění vlakových cest, Vozidlový park, Jízdní řády a oběhy vozidel, Obecné). Práce se zaměřila na schéma věnující se infrastruktuře. Byly rozebrány jednotlivé části souboru ve formátu railML a jejich význam. Čtenář se tak mohl dozvědět o jednotlivých částech souboru, a to hlavně o schématech Common, obsahující informace obecné pro všechna ostatní schémata, a Infrastructure obsahující informace o infrastruktuře.

V třetí kapitole byla prozkoumána možnost popisu infrastruktury v programu ERSA Track Editor. ERSA vznikla za účelem simulace a ověřování ERTMS/ETCS. Nejdříve bylo popsáno, z čeho se tento nový systém zabezpečení skládá. K simulaci je nutné mít popis infrastruktury, na které lze tuto simulaci provádět. Před zahájením bylo probádáno, jestli je možné exportovat tato data ve formátu railML, protože program standardně ukládá data do databáze MySQL. Bylo zjištěno, že data nelze z této verze programu exportovat ve formátu railML. Proto bylo od této možnosti popisu opuštěno. Na stránkách výrobce je nicméně uvedeno, že nové verze už export v railML umožňují. Pokud se tato novější verze dostane do DSFD je možné popis v programu ERSA znovu ověřit. Program lze ale nadále využívat pro výuku zabezpečení a řízení podle ERTMS/ETCS, pro niž představuje kvalitní nástroj. Bude pouze nutné popisovat pro něj infrastrukturu samostatně.

V poslední kapitole došlo ke zhodnocení možného využití těchto formátů a programů pro popis železniční infrastruktury v DSFD. Z programu ERSA v DSFD není možné exportovat soubor ve formátu railML. Proto nebyl pro účely práce dále zkoumán. Formát RTM/railML se ale jeví jako vhodný pro popis kolejíště. Vzhledem ke své struktuře ale není efektivní psát soubor v railML přímo v textové podobě. Proto byla snaha najít jiný vhodný program. V DSFD se nachází program, ve kterém je možné popsat železniční infrastrukturu v grafickém rozhraní a poté vyexportovat ve formátu railML. Jedná se o program RaiLaiD. Program RaiLaiD byl pro účely této práce vybrán jako náhrada za program ERSA.

V praktické části již došlo k popisu části železniční infrastruktury v grafickém rozhraní programu RaiLaiD. V tomto případě se jednalo o popis stanice Senohraby. Základní schéma bylo poskytnuto Ing. Petrem Kouteckým. V grafickém rozhraní byly zakresleny staniční koleje

a poté došlo k přidání ostatních prvků nacházejících se v kolejišti. Prvky, které nebyly součástí základní knihovny, byly vymodelovány v programu Object Modeler, který je dodáván spolu se základním programem. Těmto prvkům byly následně po přidání do schématu přiřazeny hodnoty jejich vlastností. Jednalo se o sedmdesát pět různých entit. Poté bylo možné vyexportovat z programu soubor ve formátu railML, který popisuje danou stanici.

Mým hlavním cílem při vypracování bakalářské práce bylo získat ucelené informace o možnostech popisu infrastruktury. Získal jsem cenné znalosti týkající se popisu železniční infrastruktury ve formátu RailTopoModel/railML. Zároveň jsem při vytváření stanice Senohraby nabyl zkušenosti s používáním programu RaiLAIID, které mohu uplatnit v budoucnosti při popisu železniční infrastruktury v DSFD. Také jsem získal základní znalost ovládání programu ERSA. Tento program, jak bylo řečeno výše, nesplňuje v této verzi požadavky na univerzální popis železniční infrastruktury v DSFD. Je ale možné v něm v budoucnosti testovat scénáře pro ERTMS/ETCS. I když bude nutné pro něj vytvořit popis železniční infrastruktury samostatně.

Zdroje

- [1] Business objectives of the RailTopoModel. RailTopoModel [online]. Paris: UIC, 2020 [cit. 2020-06-28]. Dostupné z: <http://www.railtopomodel.org/en/business-objectives-railtopomodel.html>
- [2] State of development. RailTopoModel [online]. Paris: UIC, 2020 [cit. 2020-06-28]. Dostupné z: <https://www.railtopomodel.org/en/state-of-development.html>
- [3] FEASIBILITY STUDY - UIC RailTopoModel and data exchange format [online]. Zurich/Paris: trafIT solutions gmbh, 2013 [cit. 2020-06-28]. Dostupné z: http://www.railtopomodel.org/files/download/RailTopoModel/270913_trafIT_FinalReportFeasibilityStudyRailTopoModel.pdf
- [4] RailTopoModel - Railway infrastructure topological model [online]. 1st edition. Paris: International Union of Railways (UIC), 2016 [cit. 2020-08-04]. ISBN 978-2-7461-2513-1. Dostupné z: http://www.railtopomodel.org/en/download/irs30100-apr16-7594BCA1524E14224D0.html?file=files/download/RailTopoModel/180416_uic_irs30100.pdf
- [5] Connexity graph. RailTopoModel Wiki [online]. Paris: ERIM, 2017 [cit. 2020-08-04]. Dostupné z: http://wiki.railtopomodel.org/index.php?title=Connexity_graph&oldid=866
- [6] Sled (graf). In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-2020, 15. 6. 2019 [cit. 2020-08-04]. Dostupné z: [https://cs.wikipedia.org/wiki/Sled_\(graf\)](https://cs.wikipedia.org/wiki/Sled_(graf))
- [7] Core elements. RailTopoModel Wiki [online]. Paris: ERIM, 2017 [cit. 2020-08-07]. Dostupné z: http://wiki.railtopomodel.org/index.php?title=Core_elements&oldid=867
- [8] RIGAZZI, Beat. File:LanguageUnitsNew.png: A diagram describing the language units. Belongs to "Core elements". In: RailTopoModel Wiki [online]. Paris: ERIM, 2016 [cit. 2020-08-10]. Dostupné z: <http://wiki.railtopomodel.org/index.php?title=File:LanguageUnitsNew.png>
- [9] LOOF, Nina. Levels of detail. In: RailTopoModel Wiki [online]. Paris: ERIM, 2016 [cit. 2020-08-10]. Dostupné z: <http://wiki.railtopomodel.org/index.php?title=File:SingleSwitchExample.png>
- [10] Levels of detail. RailTopoModel Wiki [online]. Paris: ERIM, 2017 [cit. 2020-08-04]. Dostupné z: http://wiki.railtopomodel.org/index.php?title=Levels_of_detail&oldid=868
- [11] Topological structure (network). RailTopoModel Wiki [online]. Paris: ERIM, 2017 [cit. 2020-08-04]. Dostupné z: [http://wiki.railtopomodel.org/index.php?title=Topological_structure_\(network\)&oldid=886](http://wiki.railtopomodel.org/index.php?title=Topological_structure_(network)&oldid=886)
- [12] DESSAGNE, Gilles. Topological structure (network). In: RailTopoModel Wiki [online]. Paris: Snf Réseau, 2016 [cit. 2020-08-10]. Dostupné z: [http://wiki.railtopomodel.org/index.php?title=Topological_structure_\(network\)](http://wiki.railtopomodel.org/index.php?title=Topological_structure_(network))
- [13] Hierarchical structure (levels). RailTopoModel Wiki [online]. Paris: ERIM, 2017 [cit. 2020-08-04]. Dostupné z: [http://wiki.railtopomodel.org/index.php?title=Hierarchical_structure_\(levels\)&oldid=887](http://wiki.railtopomodel.org/index.php?title=Hierarchical_structure_(levels)&oldid=887)

- [14] LOOF, Nina. Hierarchical structure (levels). In: RailTopoModel Wiki [online]. Brussels: InfraBel, 2016 [cit. 2020-08-10]. Dostupné z: [http://wiki.railtopomodel.org/index.php?title=Hierarchical_structure_\(levels\)](http://wiki.railtopomodel.org/index.php?title=Hierarchical_structure_(levels))
- [15] Coordinate positioning. RailTopoModel Wiki [online]. Paris: ERIM, 2017 [cit. 2020-08-04]. Dostupné z: http://wiki.railtopomodel.org/index.php?title=Coordinate_positioning&oldid=925
- [16] LOOF, Nina. Coordinate positioning. In: RailTopoModel Wiki [online]. Paris: RFF/SNCF Réseau, 2016 [cit. 2020-08-10]. Dostupné z: http://wiki.railtopomodel.org/index.php?title=Coordinate_positioning
- [17] Linear Positioning / referencing. RailTopoModel Wiki [online]. Paris: ERIM, 2017 [cit. 2020-08-04]. Dostupné z: http://wiki.railtopomodel.org/index.php?title=Linear_Positioning_/_referencing&oldid=933
- [18] AUGELE, Vivien. Linear Positioning / referencing. In: RailTopoModel Wiki [online]. Paris: ERIM, 2016 [cit. 2020-08-10]. Dostupné z: http://wiki.railtopomodel.org/index.php?title=Linear_Positioning_/_referencing
- [19] Intrinsic positioning / referencing. RailTopoModel Wiki [online]. Paris: ERIM, 2017 [cit. 2020-08-04]. Dostupné z: http://wiki.railtopomodel.org/index.php?title=Intrinsic_positioning_/_referencing&oldid=934
- [20] LOOF, Nina. Intrinsic positioning / referencing. In: RailTopoModel Wiki [online]. Brussels: InfraBel, 2016 [cit. 2020-08-10]. Dostupné z: http://wiki.railtopomodel.org/index.php?title=Intrinsic_positioning_/_referencing
- [21] LOOF, Nina. Intrinsic positioning / referencing. In: RailTopoModel Wiki [online]. Paris: RFF/SNCF Réseau, 2016 [cit. 2020-08-10]. Dostupné z: http://wiki.railtopomodel.org/index.php?title=Intrinsic_positioning_/_referencing
- [22] Object positioning in the network. RailTopoModel Wiki [online]. Paris: ERIM, 2017 [cit. 2020-08-04]. Dostupné z: http://wiki.railtopomodel.org/index.php?title=Object_positioning_in_the_network&oldid=889
- [23] LOOF, Nina. Object positioning in the network. In: RailTopoModel Wiki [online]. Brussels: InfraBel, 2016 [cit. 2020-08-10]. Dostupné z: http://wiki.railtopomodel.org/index.php?title=Object_positioning_in_the_network
- [24] Introduction - railML.org (EN) [online]. Dresden: railML.org, c2002-2020 [cit. 2020-08-07]. Dostupné z: <https://www.railml.org/en/introduction.html>
- [25] RailML – a standard interface for railway data. In: Global Railway Review [online]. Brasted (United Kingdom): Russell Publishing Limited, 2005 [cit. 2020-08-07]. Dostupné z: <https://www.globalrailwayreview.com/article/2780/railml-a-standard-interface-for-railway-data/>
- [26] RailML 3 Wiki [online]. railML.org, 2019 [cit. 2020-08-07]. Dostupné z: https://wiki3.railml.org/index.php?title=Main_Page
- [27] RAHMIG, Christian. RailML® 3.1 Tutorial: Simple Example Step-by-Step [online]. Version 1.2. Braunschweig (Německo): Rahmig, 2017, 12.11.2018, 46 s. [cit. 2020-08-07]. Dostupné z: https://svn.railml.org/railML3/tags/railML-3.1-rc/doc/tutorial/railML_SimpleExample_Tutorial-IS_v01.02_20181112.pdf

- [28] European Rail Traffic Management System (ERTMS). European Union Agency for Railways [online]. Valenciennes (Francie): ERA, 2020 [cit. 2020-08-04]. Dostupné z: https://www.era.europa.eu/activities/european-rail-traffic-management-system-ertms_en
- [29] Home - ERSA spécialiste des solutions ERTMS/ETCS [online]. Aix-en-Provence: CLEARSY, 2018 [cit. 2020-08-07]. Dostupné z: <https://ersa.clearsy.com/>
- [30] ETCS TRACK EDITOR. Home - ERSA spécialiste des solutions ERTMS/ETCS [online]. Aix-en-Provence (Francie): CLEARSY, 2018 [cit. 2020-08-04]. Dostupné z: <https://ersa.clearsy.com/products/etcs-track-editor/?lang=en>
- [31] ERTMS/ETCS OPERATIONAL SIMULATOR. Home - ERSA spécialiste des solutions ERTMS/ETCS [online]. Aix-en-Provence: CLEARSY, 2018 [cit. 2020-08-07]. Dostupné z: <https://ersa.clearsy.com/products/ertms-etcs-operational-simulator/?lang=en>
- [32] Rail-AiD Desktop: User Manual [online]. Rev. 2.0. Rome: NEAT, 2019, 17/04/2020, 80 s. [cit. 2020-08-07]. Dostupné z: <https://www.rail-aid.com/download/rail-aid-desktop/>. Rail-AiD_D_UserManual.pdf.
- [33] Rail-AiD Object Modeler: User Manual [online]. Rev. 1.0. Rome: NEAT, 2019, 20/04/2020, 54 s. [cit. 2020-08-07]. Dostupné z: <https://www.rail-aid.com/download/rail-aid-object-modeler/>. Rail-AiD_OM_UserManual.pdf.
- [34] DAVID, Cohn. Software Review: Rhino 6 Is A Powerful Creature: Rhino, the popular NURBS-based modeler continues to grow. Digital Engineering 24/7 - Optimal Technology for Engineering Design, Simulation, Prototyping, Testing and Computing [online]. Northbrook (Illinois): Peerless Media, 2020 [cit. 2020-08-10]. Dostupné z: <https://www.digitalengineering247.com/article/rhino-6-is-a-powerful-creature/>
- [35] Supported File Formats. Rhino 6 for Windows and Mac [online]. Seattle: Robert McNeel & Associates., 2020 [cit. 2020-08-10]. Dostupné z: <https://www.rhino3d.com/formats/>
- [36] KOUTECKÝ, Petr. Plánek kolejíště Dopravního sálu Fakulty dopravní ČVUT. Praha, 2017.

Obrázky

obr. 1: Rozdělení podle barev balíčků ve schématu UML [8].....	9
obr. 2: Databázová struktura RTM [4]	10
obr. 3: Popis infrastruktury v úrovni Micro [4]	10
obr. 4: Popis infrastruktury v úrovni Meso [4]	11
obr. 5: Popis infrastruktury v úrovni Macro [4].....	11
obr. 6: Popis infrastruktury v úrovni Nano (jednoduchá výhybka) [9]	11
obr. 7: Klasická reprezentace železniční sítě [12]	12
obr. 8: Elementy vzniklé z původní sítě [12]	12
obr. 9: Elementy původní sítě s vzájemnými relacemi [12].....	12
obr. 10: Popis relace R1 [12].....	13
obr. 11: Výsledná síť s vyobrazením nemožnosti přímé cesty mezi elementy 2 a 3 [12]	13
obr. 12: Původní síť [14]	13
obr. 13: Elementy původní sítě [14].....	14
obr. 14: Rozdělení lineárních elementů [14].....	14
obr. 15: Princip dělení lineárních elementů [14]	14
obr. 16: Rozdělení elementů na části [14].....	14
obr. 17: Postup spojení elementů do funkčních celků [14]	15
obr. 18: Spojení elementů do funkčních celků [14].....	15
obr. 19: Výsledná síť [14].....	15
obr. 20: Schematické souřadnice bodu [16]	16
obr. 21: Absolutní LRS [18].....	16
obr. 22: Relativní LRS [18].....	17
obr. 23: Princip VRS pro lineární element [20]	17
obr. 24: Princip VRS pro nelineární element [20]	17
obr. 25: Lineární elementy sítě s vyznačenými hodnotami počátku a konce [21].....	18
obr. 26: Příklad převodu [16].....	18
obr. 27: Příklady možných lokalizací entit [23]	19
obr. 28: Příklad bodové entity umístěné uprostřed elementu A [23]	20
obr. 29: Příklad lineární entity [23].....	20
obr. 30: Příklad plošné lokace [23].....	21
obr. 31: Příklad plošné lokace s podrobným popisem přidružených elementů [23].....	21
obr. 32 Budoucí situace při přenosu dat mezi aplikacemi [24].....	22
obr. 33 Základní tagy v railML [27]	24
obr. 34 Obsah tagu common [autor – vystřiženo z railML]	24
obr. 35 Obsah tagu topology [autor – vystřiženo z railML].....	25
obr. 36 Obsah tagu functionalInfrastructure [autor – vystřiženo z railML]	25
obr. 37 Obsah tagu netElement [autor – vystřiženo z railML]	26
obr. 38 Obsah tagu netRelation [autor – vystřiženo z railML]	26
obr. 39 Obsah tagu network [autor – vystřiženo z railML].....	27
obr. 40 Obsah tagu balise [autor – vystřiženo z railML].....	27
obr. 41: Ukázka úpravy obsahu zpráv balíz v Track Editoru [30].....	29
obr. 42: Ukázka HMI a ovládacích prvků v prostředí ERSA simulátoru [31]	29
obr. 43: Grafické rozhraní programu Rhinoceros 6 [34].....	32
obr. 44: Grafické rozhraní v RaiLAIdu [autor – vystřiženo z RaiLAIdu]	33
obr. 45: Vyhledávač objektů v RaiLAIdu [autor – vystřiženo z RaiLAIdu].....	33
obr. 46: Nastavování vlastností objektů v RaiLAIdu [autor – vystřiženo z RaiLAIdu]	34
obr. 47: Schéma stanice Senohraby [36]	34

obr. 48: Schéma stanice Senohraby v RaiLaiDu [autor – vystřiženo z RaiLaiDu]	35
obr. 49: Zobrazení přepínatelné balízy [autor – vystřiženo z RaiLaiDu]	35
obr. 50: Zobrazení třech typů proměnných návěstidel [autor – vystřiženo z RaiLaiDu]	35
obr. 51: Zobrazení označnicku [autor – vystřiženo z RaiLaiDu]	35
obr. 52: Tabulka entit [autor – vystřiženo z RaiLaiDu].....	36
obr. 53: Tabulka tvarů [autor – vystřiženo z RaiLaiDu].....	36
obr. 54: Tabulka vytváření atributů [autor – vystřiženo z RaiLaiDu]	37
obr. 55: Vlastnosti proměnného návěstidla [autor – vystřiženo z RaiLaiDu]	38
obr. 56: Konečná podoba stanice Senohraby v grafickém rozhraní [autor – vystřiženo z RaiLaiDu]	39
obr. 57: Metadata souboru se stanicí Senohraby [autor – vystřiženo z railML].....	39

Tabulky

tab. 1: Elementy lineární entity a jejich vnitřní souřadnice [23]	20
-------------------------------------------------------------------------	----

Přílohy

Příloha č. 1: Stanice Senohraby ve formátu railML

Příloha č. 2: Stanice Senohraby ve formátu PDF

Příloha č. 3: Stanice Senohraby ve formátu RaiLAiDu