ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE

F6

Fakulta dopravní
Ústav dopravní telematiky

Disertační práce

Metodika integrace a dezintegrace prvků topologického popisu železniční infrastruktury

Ing. Adam Hlubuček

2023
Školitel: Doc. Ing. Martin Leso, Ph.D.

Poděkování / Prohlášení

Děkuji svému školiteli doc. Ing. Martinu Lesovi, Ph.D. za to, že mi byl po celou dobu mého doktorského studia na ČVUT v Praze Fakultě dopravní vždy ochoten pomoci jak ve věcech odborných, tak ve věcech administrativních. Jemu, jakožto i dalším zaměstnancům Fakulty dopravní, jmenovitě Ing. Zuzaně Bělinové, Ph.D., doc. Ing. Vítu Fáberovi, Ph.D. a Ing. Michalovi Matowickému, Ph.D., děkuji též za to, že mi umožnili nebo zprostředkovali možnost podílet se na výzkumných projektech a dalších činnostech, díky nimž jsem si osvojil poznatky, které jsem mohl při tvorbě této práce a publikační činnosti s ní související aplikovat. Výše jmenovaným, stejně jako mnohým dalším kolegům a přátelům z řad stávajících i bývalých zaměstnanců a studentů Fakulty dopravní, níže jmenovaného nevyjímaje, děkuji také za spoluutváření přívětivého prostředí pro práci i rekreaci na půdě akademické Fakulty dopravní i mimo ni. Dále děkuji Ing. Mgr. Robertu Číhalovi, CSc., mimo jiné členovi předsednictva CAGI, a to za zasvěcení do tajů problematiky datového popisu železniční infrastruktury. V neposlední řadě děkuji Vascovi P. Kolmorgenovi MSc. a Christianovi Rahmigovi MSc., koordinátorům z railML.org, za přijetí a spolupráci na stáži v Drážďanech, při níž jsem mohl poznatky shrnuté v této práci v praxi uplatnit a dále je rozvinout. S poděkováním nemohu zapomenout ani na Ing. Zdeňka Michla, zaměstnance ČVUT v Praze Fakulty dopravní a studenta Masarykova ústavu vyšších studií ČVUT v Praze, který mě uvedl do světa railML a podpořil mě v udržení si kontaktu s ním, díky němuž jsem měl mimo jiné možnost výše uvedenou stáž absolvovat.

Předkládám tímto k posouzení a obhajobě disertační práci, zpracovanou během studia na ČVUT v Praze, Fakultě dopravní.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité 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 31. 3. 2023

Ing. Adam Hlubuček

Abstrakt / Abstract

Různé drážní aplikace a metody popisu železniční infrastruktury využívají pro vyjádření topologie železniční sítě různých rozlišovacích úrovní. V současné době, zejména z důvodu nároků rozvíjejících se inteligentních dopravních systémů na železnici, vzrůstá potřeba data integrovat. Tato situace je způsobena tím, že data popisující železniční infrastrukturu často pocházejí z různých zdrojů, zejména z účelově orientovaných databází jednotlivých služebních odvětví. Jednou z překážek v integraci dat je právě rozdílnost jednotlivých rozlišovacích úrovní. Cílem této práce je navrhnout metodiku, která umožní zajistit převod dat mezi referenční rozlišovací úrovní a odvozenými rozlišovacími úrovněmi. Jsou navrženy dva typy algoritmů, které s využitím definovaných operací umožňují realizovat transformační proces. Tyto algoritmy mohou být aplikovány na systém topologického popisu železniční infrastruktury, o němž je v této práci rozsáhle pojednáno jak na obecné úrovni, tak s využitím prostředků Víceúčelového modelu železniční infrastruktury navrženého autorem práce na základě principů mezinárodního doporučení UIC RailTopoModel.

Klíčová slova: železniční infrastruktura, topologický popis, rozlišovací úroveň, integrace, dezintegrace

Various railway applications and methodologies designed for description of railway infrastructure use different levels of detail to express topology of railway network. Currently, especially caused by the requirements of the emerging railway related intelligent transport systems, needs of data integration are arising. This situation is caused by the fact that the data describing the railway infrastructure often comes from different sources, especially from purpose-oriented databases of individual professional sectors. One of the obstacles in the integration of data is the difference between the individual levels of detail. The aim of this thesis is to propose a methodology, which enable to carry out a data transfer between the reference level of detail and derived levels of detail. Two types of algorithms to ensure integration and disintegration process are designed. These algorithms can be applied to the topological railway infrastructure description system, which is extensively discussed in this thesis both at a general level and with the use of the resources of the Multipurpose railway infrastructure model designed by the author of the thesis based on the principles of the international UIC recommendation RailTopoModel.

Keywords: railway infrastructure, topological description, level of detail, integration, disintegration

Obsah /

Tabulky / Obrázky

Kapitola 1
Úvod

V současné době existuje řada různých účelově orientovaných pohledů týkajících se popisu železniční infrastruktury. Jednotlivé metodiky popisu železniční sítě se liší v závislosti na konkrétních úlohách a zamýšleném využití, pro které byly původně navrženy. [1] Efektivně kombinovat data pocházející z různých zdrojů zaměřených na různé komponenty popisu sítě lze jen zřídkakdy. Přitom komplexní pohled na železniční síť jako na ucelený systém sestávající se z jednotným způsobem lokalizovaných prvků reprezentujících objekty jednotlivých služebních odvětví je nezbytným předpokladem pro zavedení efektivně fungujících inteligentních dopravních systémů na železnici i rozvoj aplikací, které popisu železniční infrastruktury využívají pro jiné účely (např. správa a evidence hmotného majetku, vizualizace železniční infrastruktury apod.).

Kromě různorodosti způsobu evidence jednotlivých objektů a vlastností železniční dopravní cesty, která zahrnuje jak věcné charakteristiky (lišící se v závislosti na konkrétním účelovém pohledu), tak lokalizaci vůči definovaným souřadným systémům (které opět mohou být definovány pro různé účely různým způsobem), je komplikací při výměně dat mezi jednotlivými aplikacemi také skutečnost, že různé systémy využívají popisu infrastruktury na různých rozlišovacích úrovních. Z hlediska systému topologie železniční sítě je možné rozlišovací úroveň chápat ve smyslu jejího členění na tzv. síťové prvky (tak, jak jsou definovány v metodice UIC RailTopoModel). Tyto síťové prvky reprezentují určité evidenčních jednotky, ze kterých je síť složena. V závislosti na uvažované rozlišovací úrovni se může jednat např. o výhybky a koleje ve stavebním smyslu, o kolejové trasy, o železniční stanice a mezistaniční traťové úseky, o železniční tratě atd.

Seskupování jednotek ekvivalentních těmto prvkům na určité rozlišovací úrovni za účelem vzniku jednotek ekvivalentních prvkům na méně podrobné rozlišovací úrovni probíhá běžně (např. v souladu se zavedenými postupy metodik využívaných i v České republice v rámci Správy železnic) definičně, tedy na základě rozhodnutí zpracovatele. To znamená, že na základě znalosti popisu železniční sítě na podrobné rozlišovací úrovni není obvykle možné deterministicky vyjádřit popis na méně podrobné rozlišovací úrovni, a to ani za předpokladu, že je explicitně definována přípustná datová struktura na obou uvažovaných rozlišovacích úrovních (zdrojové a cílové). Tato skutečnost znemožňuje jednoznačnou transformaci dat mezi jednotlivými rozlišovacími úrovněmi a komplikuje návrh rozhraní pro efektivní výměnu dat mezi aplikacemi.

Cílem této práce je navrhnout metodiku, která umožní na základě znalosti popisu železniční sítě na referenční rozlišovací úrovni a množiny transformačních operací navrhovat algoritmy, pomocí kterých bude možné strukturu systému topologického popisu železniční infrastruktury vyjádřenou rozlišovací úrovni označené jako referenční transformovat na strukturu vyjádřenou na odvozené rozlišovací úrovni. Vzhledem k tomu, že soubor transformačních operací a jejich parametrů může být jednotně definován pro stanovený okruh úloh vyžadujících transformaci mezi definovanými rozlišovacími úrovněmi, takové transformace, při kterých není nutný zásah zpracovatele (lze očekávat spíše v případě integrace), bude možné považovat za deterministické.

1.1 Současný stav problematiky

Železnice je systémem, který již historicky podléhá důkladné dokumentaci, a to ve všech fázích svého životního cyklu. Již pro samotný návrh tohoto systému existuje řada okrajových podmínek a závislostí mezi veličinami, jimiž je možné na různých rozlišovacích úrovních jednotlivé jeho subsystémy popsat. Tato pravidla, obvykle založená na technické podstatě popisovaných jevů, avšak zároveň podléhající jisté míře abstrakce, byla v průběhu vývoje systému železnice formována v rámci národních prostředí jednotlivých železničních správ a společností a ani v současnosti (a patrně ani nikdy v budoucnosti) nelze tento proces považovat za ukončený. Navzdory tomu, že v posledních letech probíhají snahy o nadnárodní standardizaci (např. mezinárodní normy UIC, a specifikace TSI, které zpracovává ERA), stále přetrvávají mnohé národní odlišnosti, a to jak ve způsobu návrhu, tak ve způsobu popisu systému a jeho částí. Tyto odlišnosti je v mnohých případech možné identifikovat i na úrovni jednotlivých společností a služebních odvětví. Uvedené platí také pro subsystém infrastruktury, tedy pro vlastní železniční dopravní cestu, a jednotlivé dílčí aspekty jejího popisu. Její návrh a popis samozřejmě není možné provádět izolovaně, nýbrž je třeba zohlednit rozhraní mezi souvisejícími subsystémy.

S rozvojem softwarových aplikací napříč různými odvětvími včetně dopravy a spolu s tendencemi v praxi uplatňovat inteligentní dopravní systémy vzrůstají též požadavky na efektivní archivaci a sdílení souvisejících dat. V oblasti popisu železniční infrastruktury vzniklo v průběhu uplynulých desetiletí značné množství datových modelů, informačních systémů a rozhraní, vzájemně se svou strukturou a celkovým pojetím lišících (např. v závislosti na dodavateli) nejen mezi jednotlivými správci infrastruktury, ale také vnitropodnikově. Příslušné databáze a rozhraní totiž často vznikaly účelově, orientovány pouze na zajištění konkrétního úkolu. [1] Tento stav vyústil v situaci, kdy jsou tytéž skutečnosti evidovány multiplicitně, pro potřeby různých specializovaných aplikací, majících mnohdy rozdílné požadavky na přesnost a detailnost poskytovaných dat, a kdy jsou tytéž objekty popisovány v různých systémech z různých pohledů bez jednoznačné vzájemné identifikační vazby.

V kontextu existence pravidel definovaných v rámci nadnárodních, národních i vnitropodnikových dokumentů (specifikací, norem), resp. vyplývajících přímo z technické (matematické, geometrické, logické) podstaty popisovaných jevů lze v mnohých případech vysledovat z hlediska struktur datového popisu redundanci. Ta se v praxi často projevuje v podobě vzájemné nekonzistence hodnot, jichž konkrétní popisné nebo polohopisné veličiny nabývají. V různých modelech železniční infrastruktury je využíváno různého členění topologického popisu sítě na dílčí prvky, a to i na rozlišovacích úrovních, které si jsou svou podrobností podobné. Uvedená situace nejen že generuje náklady na správu a provoz databázových systémů, ale stává se také překážkou v synchronizaci jednotlivých navazujících aplikací, neboť znesnadňuje (až znemožňuje) jednoznačnou lokalizaci popisných dat ve vztahu k těmto síťovým prvkům, rozdílně definovaným v závislosti na konkrétních úlohách.

Integrace datového popisu infrastruktury je mimo jiné důležitá zejména s ohledem na rozvoj inteligentních dopravních systémů na železnici, tedy pro úlohy, při kterých hraje klíčovou roli rozhraní mezi statickým a mobilním subsystémem železnice, tedy subsystémy infrastruktury a vozidla. Data o infrastruktuře i o vozidle jsou v těchto případech vstupy do procesního informačního sytému a závisí na nich také bezpečnost jízdy vozidla v podmínkách stavebního uspořádání železniční trati a ostatního železničního provozu. Z tohoto důvodu vyžadují přesná, garantovaná a komplexní data o infrastruktuře. Příkladem takových aplikací jsou systémy automatického vedení vlaku a zabezpečení jízdy

vlaku (systémy typu ATC a ATO). Tyto systémy se z hlediska své funkce vzájemně doplňují, [2] a měly by tak také využívat vždy aktuální a pokud možno společnou základnu popisu železniční infrastruktury. V současné fázi implementace v podmínkách České republiky (v podobě ETCS a AVV) ale využívá každý svůj vlastní popis. Důkladný popis železniční infrastruktury je neméně důležitý pro úlohy operativní řízení železničního provozu, které pro spolehlivé plnění své funkce vyžadují přesnou predikci polohy železničních vozidel v čase. Díky těmto aplikacím by měla být zajištěna optimalizace vzájemného pohybu vlaků, což je předpokladem eliminace konfliktů v síti (např. týkajících se čerpání disponibilní kapacity železniční dopravní cesty), snížení energetických ztrát a zvýšení plynulosti železničního provozu.

V České republice na úrovni Správy železnic, obdobně jako v případě řady zahraničních provozovatelů železniční infrastruktury, je způsob popisu železniční sítě do značné míry poplatný rozdělení na jednotlivá služební odvětví. Informační systémy Správy železnic, vycházející z tradice ČD a ČSD, které vznikaly pro jejich potřebu, jsou tak orientovány většinou na konkrétní cíle a úkoly, které měly v rámci dané odbornosti plnit. Od této skutečnosti se odvíjí i jejich obsah, rozlišovací úroveň, rozsah, struktura, míra a způsob provázanosti s jejich informačním okolím, resp. původ, přesnost a garantovanost dat v nich obsažených. Tyto aspekty se pro jednotlivé systémy v mnohých případech poměrně významně liší, což je jedním z důvodů, který je překážkou v jejich vzájemné propojitelnosti. [3]

Na vzdory tomu, že v současné době probíhá intenzivní rozvoj nových informačních systémů Správy železnic ve směru k nástrojům využívajících mimo jiné technologií GIS (např. Digitální technická mapa železnice) a ve fázi projektové přípravy a realizace stavby také BIM, mnohá data jsou stále evidována v tabelární podobě bez vyhovujícího grafického rozhraní. Příkladem mohou být četné a rozsáhlé pasportní systémy jednotlivých služebních odvětví. V této oblasti probíhají snahy o integraci dat pod nově vnikající Pasport technické infrastruktury, společnou bázi pro lokalizaci jehož prvků se stává taktéž nový Pasport topologie sítě. [4]

1.2 RailTopoModel a railML 3

Za účelem standardizovat postupy v oblasti datového modelování železniční infrastruktury byla na UIC v roce 2013 zahájena iniciativa RailTopoModel. [6] Tento konceptuální systémový model, vydaný jako doporučení UIC IRS 30100, představuje jednotnou společnou základnu, která na značně obecné úrovni definuje, jakým způsobem mají být jednotlivé komponenty popisu sítě a vztahy mezi nimi modelovány, věcně popisovány, a to včetně jejich agregace a vizualizace na různých rozlišovacích úrovních. Principy RailTopoModelu (zejména pak jeho modul topologie) lze aplikovat na popis sítí obecně, ne tedy nutně pouze na popis železniční infrastruktury. [5, 7]

Problematiku popisu logické reprezentace topologie sítě řeší RailTopoModel zavedením síťových prvků. Ty obecně reprezentují jednotlivé komponenty infrastruktury (koleje, dopravny, traťové úseky a další), které vystupují v kontextu určité rozlišovací úrovně a mohou být vyjádřeny prostřednictvím uzlů grafu ve smyslu matematického konstruktu. Hrany tohoto grafu potom reprezentují síťové vazby, jimiž jsou jednotlivé síťové prvky vzájemně propojeny.

S využitím uživatelsky definovatelných souřadnicových systémů je možné vůči topologii sítě lokalizovat (bodově, liniově a plošně) objekty a charakteristiky železniční dopravní cesty, souhrnně nazývané síťovými entitami. Ty jsou v rámci vlastního Rail-

TopoModelu definovány pouze na zcela obecné úrovni. Teprve v rámci konkrétních případů užití RailTopoModelu dochází k jejich konkretizaci. [7–8]

Své praktické uplatnění nalézá RailTopoModel ve spojení s výměnným formát railML 3. Tento formát je vyvíjen od roku 2001 v rámci nezávislého konsorcia railML.org a představuje datové rozhraní v oblasti jízdních řádů, železničních vozidel, infrastruktury a zabezpečovacího zařízení, založené na XML. Nově v railML.org působí také pracovní skupina zabývající se ontologií, tedy formálním popisem znalostí, aplikovanou na datový popis železniční infrastruktury. [9–10]

V současnosti rozvíjená verze railML 3 využívá popisu infrastruktury založeného na RailTopoModelu. Na rozdíl od předcházejících verzí railML umožňuje railML 3 v reakci na RailTopoModel mimo jiné popsat železniční infrastrukturu na několika různých rozlišovacích úrovních. [11] Ačkoliv je prostřednictvím RTM a railML možné vyjádřit, jakým způsobem jsou síťové prvky podrobnější rozlišovací úrovně seskupeny v síťové prvky méně podrobné rozlišovací úrovně, neexistují doposud žádná pravidel a kritéria, podle nichž by bylo možné integraci nebo dezintegraci těchto prvků provádět. Jednou z aktuálních otázek řešených v rámci rozvoje tohoto výměnného datového formátu je problematika rozdělování a opětovného spojování souborů. Také v této oblasti jsou jednou z překážek rozdílnosti týkající se různým způsobem členěného popisu topologické vrstvy. [12–13] Možným řešením tohoto problému je návrh algoritmu umožňujícího datový popis realizovaný prostřednictvím tohoto výměnného datového formátu transformovat napříč jednotlivými případy užití, které se mezi sebou v těchto aspektech mohou lišit.

1.3 Víceúčelový model železniční infrastruktury

Víceúčelový model železniční infrastruktury (dále VMŽI) je datovým modelem, který díky společnému obecnému jádru a specifickým modulárním rozšířením umožňuje být uplatněn v různorodých úlohách týkajících se popisu železniční infrastruktury. Návrh logické struktury jádra VMŽI vychází z mezinárodního doporučení UIC RailTopoModel (dále RTM), vydaného pod označením IRS 30100, při jehož implementaci však došlo k provedení některých principiálních změn. Není-li uvedeno jinak, níže uvedená srovnání VMŽI s RTM se vztahují k RTM verze 1.1. [8]

Verze 1.0 jádra VMŽI vzniklo v průběhu roku 2019 jako výstup projektu Vnitřní soutěže IP 2019, dílčího úkolu Implementace moderních přístupů k popisu železniční infrastruktury do výuky na pracovišti Dopravního sálu Fakulty dopravní ČVUT v Praze, jehož řešitelem byl autor této práce. [14] Od té doby autor model průběžně rozvíjí a jeho specifická rozšíření modifikuje pro potřeby různých výzkumných projektů. [15]

Postupná implementace VMŽI při řešení konkrétních navazujících úloh si kromě rozvoje specifických modulárních rozšíření vyžádala také postupný vývoj jádra modelu. Některé vývojové mezifáze VMŽI, které už jsou v současné době překonány, popsal autor v rámci publikovaných konferenčních příspěvků. [16–17] V současnosti používaná verze 12.2 jádra VMŽI, která byla dokončena v červnu 2022, bude podrobně popsána v kapitole 2 této práce.

1.3.1 Princip VMŽI

Návrh VMŽI kombinuje přehledný objektový přístup k popisu železniční infrastruktury, který přináší RTM, vyjádřený s využitím UML diagramu tříd, s přístupem založeným na principu relační databáze, umožňujícím data z datové struktury získávat metodami

SQL dotazů, způsoby jejichž konstruování jsou v rámci odborné veřejnosti běžně známé. Tabulky relační databáze VMŽI jsou navrženy na základě tříd VMŽI, které ve vztahu k RTM mohou být:

- třídy RTM přímo implementované do VMŽI,
- třídy VMŽI, které jsou zavedeny za účelem nahradit funkci jiné třídy nebo tříd RTM,
- třídy VMŽI, které jsou zavedeny nad rámec RTM
- asociační třídy

Každý objekt VMŽI je instancí některé z tříd VMŽI a v rámci databáze VMŽI může být vyjádřen buď prostřednictvím záznamů několika tabulek (jedná se o tabulku navrženou na základě třídy VMŽI, jíž je příslušný objekt instancí, a o tabulky navržené na základě všech tříd, jejichž je tato třída specializací) propojených na základě jedinečné hodnoty atributu `id` (takovéto objekty, které jsou identifikovatelné na základě hodnoty atributu `id`, budou pro účely této práce dále nazývány primárními objekty), nebo prostřednictvím jediného záznamu tabulky, navržené na základě asociační třídy VMŽI (takovéto objekty, které nejsou identifikovatelné na základě hodnoty vlastního atributu `id`, budou pro účely této práce dále nazývány sekundárními objekty). Každá třída (resp. tabulka) VMŽI je evidenčně zařazena právě do jednoho tematického bloku.

1.3.2 Členění modelu

VMŽI je možné rozdělit na obecné jádro, jehož další členění na tematické bloky reflektuje RTM verze 1.1, a oblast specifických modulárních rozšíření, jejichž náplň je závislá na konkrétním případu užití. Primárně se jedná o specializace třídy `NetEntity`, jejichž instance, konkrétní síťové entity, reprezentují objekty a vlastnosti železniční infrastruktury a jejího okolí.

Jádro VMŽI je po vzoru RTM členěno do následnících tematických bloků:

- `Base`
- `Network`
- `Topology`
- `PositioningSystem`
- `Location`
- `NetEntity`

Nejtypičtějšími bloky figurujícími v rámci rozšíření jsou následující tematické bloky:

- `Geometry`
- `ModularEntity`

Tyto rozšiřující tematické bloky se mohou pro jednotlivá specifická rozšíření svým obsahem lišit a jsou určeny právě pro specializované třídy síťových entit. Třídy VMŽI zařazené v těchto tematických blocích již nevycházejí z tříd RTM, který je definován pouze na obecné úrovni odpovídající jádru VMŽI, mohou být ale navrženy např. na základě některých tříd railML 3.

1.4 Uložení dat o primárních objektech

Tabulka relační databáze VMŽI navržená na základě třídy primárních objektů (tedy libovolné třídy VMŽI s výjimkou tříd asociačních) získává název totožný s názvem příslušné třídy a veškeré atributy této třídy kromě atributů zděděných. Jediným zděděným atributem, který tabulka dále získává, je atribut `id`. Výjimkou je tabulka navržená na základě třídy `BaseObject`, od které atribut `id` dědí všechny ostatní třídy primárních objektů. Tabulka `BaseObject` představuje kolekci `id` všech primárních objektů. Každý z těchto objektů je instancí jedné z terminálních tříd VMŽI.

Struktura VMŽI je na úrovni tříd navržena tak, že všechny třídy kromě tříd terminálních jsou třídami abstraktními. Instance tudíž vytváří pouze terminální třídy. V případě, že mezi dvěma třídami VMŽI existuje takový vztah, že jedna z těchto tříd je specializací druhé třídy, projevuje se tato skutečnost v relační databázi existencí identifikačního vztahu 1:1 mezi odpovídajícími tabulkami. Příslušné tabulky jsou propojeny právě na základě hodnoty atributu `id`. S využitím této hodnoty je možné v konkrétní tabulce identifikovat konkrétní záznam a také konkrétní primární objekt (instanci některé z terminálních tříd primárních objektů). Postupným propojením tabulek, mezi nimiž existuje identifikační vztah 1:1 (od tabulky `BaseObject` až po tabulku navrženou na základě některé terminální třídy) lze získat ucelený obraz o objektech příslušné terminální třídy. Každý primární objekt je možné uceleně popsat s využitím záznamů všech tabulek, pro něž je hodnota atributu `id` totožná s ID příslušného objektu.

V případě, že mezi dvěma třídami VMŽI existuje asociace, již je v rámci relační databáze VMŽI možné vyjádřit jako neidentifikační vztah 1:n, je tento vztah mezi odpovídajícími tabulkami databáze zaveden. K jeho realizaci slouží referenční atribut odkazující se tabulky, který se pro každý záznam této tabulky odkazuje na ten záznam odkazované tabulky, hodnota jehož atributu `id` odpovídá ID odkazovaného primárního objektu. Název uvedeného referenčního atributu je potom ve tvaru `id_OdkazovanaTabulka`. Pokud asociací mezi dvěma stejnými třídami existuje více, nebo je z jiného důvodu nutné rozlišit roli příslušného referenčního atributu, je jeho název upraven do tvaru `id_OdkazovanaTabulka_Role`. Hodnota referenčního atributu v záznamu týkajícím se určitého odkazujícího se objektu potom odpovídá hodnotě atributu `id` odkazovaného primárního objektu.

1.5 Uložení dat o sekundárních objektech

Tabulka relační databáze VMŽI navržená na základě třídy sekundárních objektů (tedy libovolné asociační třídy VMŽI) získává název totožný s názvem příslušné asociační třídy a veškeré atributy této třídy. Tyto třídy nejsou třídami implementovanými do VMŽI přímo z RTM, ale v mnohých případech jsou zavedeny VMŽI za účelem realizovat vztah mezi třídami RTM v rámci UML digramu tříd vyjádřený jiným způsobem. VMŽI oproti RTM obecně modeluje vztahy mezi třídami volněji. Tímto směrem se ubíral již vývoj RTM při přechodu z verze 1.0 na 1.1, při němž došlo k nahrazení některých kompozic agregacemi. VMŽI přináší oproti RTM mimo jiné navíc např. také koncept sdílených lokací, což umožňuje využít tutéž lokaci několika síťovými entitami opakovaně. Tento přístup na obecné úrovni vede ke zvýšenému výskytu vztahů mezi třídami typu n:m, což v případě vyjádření struktury modelu s využitím relační databáze znamená hojné využívání propojovacích tabulek.

VMŽI proto zavádí asociační třídy takovým způsobem, aby byly na propojovací tabulky s využitím výše uvedených postupů snadno převoditelné. Hlavním účelem pro-

pojovací tabulky v rámci relační databáze VMŽI je přiřadit primární objekt (obvykle) jinému primárnímu objektu. Uvedenému přiřazení, které je realizováno prostřednictvím záznamu v propojovací tabulce, mohou být v závislosti na konkrétní asociační třídě VMŽI, na jejímž základě je příslušná propojovací tabulka navržena, přiděleny hodnoty dalších atributů. Instance asociačních tříd tedy chápeme jako objekty identifikovatelné nikoliv na základě hodnoty vlastního atributu `id` ale na základě uspořádané dvojice (případně rozsáhlejší n-tice) hodnot jiných atributů. Asociační třída v pojetí VMŽI je vždy třídou terminální a vytváří instance (není tudíž třídou abstraktní). Každý z těchto sekundárních objektů je přitom zcela popsán odpovídajícím záznamem propojovací tabulky, která byla navržena na základě asociační třídy, jíž je příslušný objekt instancí. Na rozdíl od primárních objektů, data o nichž jsou zpravidla roztříštěna do několika propojitelných tabulek, v takovýchto případech k propojování záznamů různých tabulek již nedochází.

V závislosti na konkrétní asociační třídě mohou být jednotlivé objekty, které jsou jejími instancemi, identifikovány buď na základě uspořádané dvojice hodnot atributů `id` primárních objektů, které jsou prostřednictvím příslušné instance jeden druhému přiřazovány, nebo na základě hodnoty atributu `id` toho z primárních objektů, jemuž je ten druhý prostřednictvím příslušné instance přiřazován v určitém pořadí nebo roli, vyjádřené s využitím hodnoty některého z dalších atributů této třídy, a hodnoty tohoto atributu. Mezi tabulkou navrženou na základě třídy přiřazovaných primárních objektů nebo třídou obecnější (nemusí se nutně jednat o třídu terminální, může se jednat také o abstraktní třídu, tabulka na jejímž základě vytvořená obsahuje záznamy týkající se primárních objektů více terminálních tříd) a propojovací tabulkou, která přiřazení zprostředkovává, existuje vztah 1:n (jednomu záznamu v tabulce navržené na základě třídy přiřazovaných objektů může odpovídat n záznamů v propojovací tabulce). Tento vztah je realizován prostřednictvím referenčního atributu propojovací tabulky a platí pro něj stejná pravidla jako pro asociaci s tím rozdílem, že pokud je záznam propojovací tabulky identifikován za součinnosti ID primárního objektu, na nějž se příslušná instance asociační třídy odkazuje, jedná se o vztah identifikační. Každá propojovací tabulka je s využitím vztahu 1:n (resp. 1:m) napojena na dvě tabulky navržené na základě tříd přiřazovaných primárních objektů nebo obecnější s tím, že alespoň jeden z těchto vztahů je vztahem identifikačním.

Kapitola 2
Jádro VMŽI

V rámci této kapitoly budou popsány jednotlivé tematické bloky VMŽI a jejich třídy jádra VMŽI. Úvodem pojednání o každém z popisovaných tematických bloků bude daný blok krátce charakterizován včetně rámcového představení jednotlivých tříd bloku. V případě bloků jádra VMŽI, které jsou vesměs implementovanými bloky RTM, bude uvedeno, jakých odlišností daný blok oproti původní struktuře zavedené RTM doznává. V některých specifických případech bude toto porovnání provedeno také ve vztahu k railML 3.

Pro každou abstraktní třídu bude uvedeno, obecným konceptem kterého objektu je. V pro každou neabstraktní neabstraktní třídy bude uvedena charakteristika instancí dané třídy. Pro každou z popisovaných tříd bude specifikováno, zda se jedná o třídu RTM, příp. jaké změny byly učiněny při její implementaci do VMŽI, nebo zda se jedná o třídu zavedené VMŽI nad rámec RTM.

Dále bude pro každou třídu uvedeno, které dědí atributy (v případě, že je specializací jiné třídy) a jaké atributy jsou pro danou třídu zavedeny (nad rámec těch zděděných). Pro jednotlivé atributy zavedené danou třídou bude opět specifikováno, zda se jedná o atributy RTM, příp. jaké změny byly učiněny při jejich implementaci do VMŽI, nebo zda se jedná o atributy zavedené VMŽI nad rámec RTM. V rámci třídy, která daný atribut zavádí, bude každý atribut popsán podrobněji, ze zaměřením na hodnoty, kterých může nabývat a jaké skutečnosti tyto hodnoty vyjadřují.

2.1 Base

Blok `Base` je vrcholovým blokem, jehož třídy reprezentují koncepty objektů na vysoce obecné úrovni. Blok je v rámci VMŽI tvořen abstraktními třídami `BaseObject` a `NamedResource`. Atributy od nich dědí všechny třídy, jejichž instance mají vlastní ID (všechny třídy kromě asociačních), resp. pojmenování.

Při implementaci tematického bloku `Base` do VMŽI jsou jeho třídy nově klasifikovány jako abstraktní. V ostatních aspektech blok `Base` VMŽI reflektuje blok `Base` RTM.

2.1.1 BaseObject

Abstraktní třída `BaseObject` je obecným konceptem objektu nazývaného v rámci VMŽI primárním objektem. Primární objekt je objektem jednoznačně identifikovatelným prostřednictvím hodnoty atributu `id`.

Třída `BaseObject` je třídou RTM, který ji však na rozdíl od VMŽI nepovažuje za třídu abstraktní. VMŽI nicméně nepředpokládá, že by na základě této třídy byly vytvářeny instance. Jedná se o vrcholovou třídu, tudíž nedědí žádné atributy od žádných jiných tříd. Pro tuto třídu je zaveden atribut `id`. Tento atribut je atributem RTM.

Hodnota atributu `id` vyjadřuje jednoznačný identifikátor objektu. Tato hodnota musí být pro všechny objekty, které jsou instancemi některé ze specializací třídy `BaseObject`, jedinečná. Pokud tato hodnota není předem známá, je generována vždy při přidávání

nové instance. Ačkoliv se nejedná o zcela nezbytnou podmínku, předpokládáme hodnotu atributu `id` ve formátu UUID. Tuto hodnotu je možné uložit jako textový řetězec o délce 36 znaků.

2.1.2 NamedResource

Abstraktní třída `NamedResource` je obecným konceptem pojmenovaného objektu. Pojmenovaný objekt je takovým objektem, kterému je přiděleno pojmenování, a to na dvou úrovních, ve zkrácené a rozvinutelnější podobě.

Třída `NamedResource` je třídou RTM, který ji však na rozdíl od VMŽI nepovažuje za třídu abstraktní. VMŽI nicméně nepředpokládá, že by na základě této třídy byly vytvářeny instance. Jedná se o specializaci třídy `BaseObject`, tudíž dědí atribut `id`. Dále jsou pro tuto třídu zavedeny atributy `name` a `longname`. Oba tyto atributy jsou atributy RTM.

Hodnota atributu `name` vyjadřuje zkrácené pojmenování objektu ve formě textového řetězce. V rámci VMŽI předpokládáme, že pro každou instanci je tato hodnota automaticky generovaná na základě zkratky třídy, ke které příslušná instance náleží a číslici označující pořadí přidání dané instance v rámci třídy.

Hodnota atributu `longname` vyjadřuje pojmenování objektu ve formě textového řetězce, které může být delší než hodnota atributu `name`. To ale není podmínkou. V rámci VMŽI chápeme tuto hodnotu jako uživatelské pojmenování, které uživatel obvykle zadává při přidávání nové instance, přičemž výchozí hodnota je automaticky generována na základě názvu třídy, ke které příslušná instance náleží a číslici označující pořadí přidání dané instance v rámci třídy.

2.2 Network

Blok `Network` je blokem, jehož třídy umožňují popsat modelovanou síť jako celek a přiřadit jí jednotlivé síťové úrovně. Na obecné úrovni umožňují taktéž vyjádřit přiřazení síťových objektů síti a síťovým úrovním. Blok je v rámci VMŽI tvořen terminálními třídami systémového charakteru `Network` a `LevelNetwork`, jejímiž instancemi jsou jednotlivé sítě a síťové úrovně, abstraktní třídou `NetworkResource`, představující obecný koncept síťového objektu, a asociačními třídami `NetworkLevelAssignment`, `NetworkResourceAssignment` a `LevelResourceAssignment`, jejichž instance vyjadřují přiřazení jednotlivých úrovní síti a jednotlivých síťových objektů síti nebo úrovni. S jejich využitím může být prováděno např. filtrování síťových objektů podle příslušnosti k síti a úrovni.

Při implementaci tematického bloku `Network` do VMŽI dochází k zavedení uvedených asociačních tříd a k doplnění některých atributů třídy `LevelNetwork` nad rámec RTM. Nejsou implementovány atributy třídy RTM `NetworkResource` vyjadřující časovou platnost síťových objektů. V ostatních aspektech blok `Network` VMŽI reflektuje blok `Network` RTM.

2.2.1 Network

Instancemi terminální třídy `Network` jsou jednotlivé sítě. Síť je takovým objektem, jemuž je přiřazena množinou objektů, jejichž prostřednictvím je na určité síťové úrovni nebo na několika síťových úrovních popisována určitá prostorová oblast železniční infrastruktury. Každé síti může být přiřazena jedna nebo několik síťových úrovní. Každé

síti může být přiřazeno libovolné množství síťových objektů. Ačkoliv se nejedná o nezbytně nutnou podmínku, lze předpokládat, že v jedné databázi se bude vyskytovat pouze jedna instance třídy `Network`.

Třída `Network` je třídou RTM. Jedná se o specializaci třídy `NamedResource`, tudíž dědí atributy `id`, `name` a `longname`. Další atributy pro ni v rámci VMŽI nejsou zavedeny. V souladu s RTM sice třída `Network` obsahuje atributy `validFrom` a `validTo`, aktuální verze VMŽI nicméně časovou platnost objektů, kterou hodnoty těchto atributů vyjadřují, zatím nepodporuje. Výhledovým záměrem je, aby VMŽI podporoval přidělování časové platnosti objektům více dynamickým způsobem. Koncept RTM umožňuje každému objektu přidělit vždy pouze jeden interval časové platnosti.

2.2.2 LevelNetwork

Instancemi terminální třídy `LevelNetwork` jsou jednotlivé síťové úrovně. Síťová úroveň je takovým objektem, jemuž je přiřazena množina objektů, jejichž prostřednictvím je na definované rozlišovací úrovni, resp. úrovni způsobu vyjádření sítě, popsána síť. Každá síťová úroveň může být přiřazena k libovolnému množství sítí. Každé síťové úrovni může být přiřazeno libovolné množství síťových objektů.

Třída `LevelNetwork` je třídou RTM. Jedná se o specializaci třídy `BaseObject`, tudíž dědí atribut `id`. Dále jsou pro tuto třídu zavedeny atributy `descriptionLevel`, `dimension` a `representation`. Atribut `descriptionLevel` je atributem RTM. Atributy `dimension` a `representation` byly zavedeny VMŽI nad rámec RTM.

Hodnota atributu `descriptionLevel` vyjadřuje rozlišovací úroveň, která odpovídá příslušné síťové úrovni. V souladu s RTM může tento atribut nabývat následujících hodnot:

- `micro` – vyjadřuje popis sítě na úrovni jednotlivých kolejí a výhybek
- `micro` – vyjadřuje popis sítě na úrovni jednotlivých dopravních (příp. přepravních) bodů propojených jednotlivými traťovými kolejemi
- `macro` – vyjadřuje popis sítě na úrovni jednotlivých dopravních (příp. přepravních) bodů propojených jednotlivými traťovými úseky

Ačkoliv je možné si vytvořit rámcovou představu o tom, s jakou mírou podrobnosti je síť popisovaná na síťové úrovni, jíž byla přidělena některá z výše uvedených hodnot atributů, vyjádřena, nejedná se o výčet, pomocí něhož by bylo možné ve všech případech příslušnost identifikovat rozlišovací úroveň, resp. způsob topologického popisu sítě, zcela jednoznačně. Proto VMŽI připouští zavedení dalších hodnot atributu `descriptionLevel`, které by dokázaly popsat rozlišovací úroveň výstižněji. Zároveň je možné si představit síť popsanou na takové rozlišovací úrovni, kterou výše uvedené hodnoty atributu `descriptionLevel` nejsou schopny vystihnout vůbec. Jedná se např. o úroveň obvykle označovanou jako nano. Tato rozlišovací úroveň vyjadřuje popis sítě na úrovni jednotlivých kolejnic. Z tohoto důvodu VMŽI připouští v případě potřeby zavést také hodnotu `nano` atributu `descriptionLevel`. Je také možné zavést hodnotu `network` atributu `descriptionLevel` vyjadřující, že je síť na příslušné úrovni naopak popsána jako nedělitelný celek.

Hodnota atributu `dimension` vyjadřuje, s využitím jakého prostoru je síť na příslušné síťové úrovni popisována. Tento atribut byl v rámci VMŽI zaveden, aby bylo možné rozlišit mezi jednotlivými způsoby konstruování popisu sítě v prostředí grafického editoru. Atribut `dimension` může nabývat následujících hodnot:

- `p` – vyjadřuje jednorozměrný prostor v podobě přímky p s hodnotami pozic na úsečkách reprezentujících síťové prvky odpovídajícími měření vzdálenosti podél průmětu definičních křivek reálných liniových objektů do horizontální roviny
- `d` – vyjadřuje jednorozměrný prostor v podobě přímky d s hodnotami pozic na úsečkách reprezentujících síťové prvky odpovídajícími měření vzdálenosti podél definičních křivek reálných liniových objektů v prostoru
- `pz` – vyjadřuje dvourozměrný prostor v podobě roviny pz zkonstruované na základě přímky p rozšířením o vertikální rozměr z s hodnotami pozic na dvourozměrných křivkách reprezentujících síťové prvky odpovídajícími měření vzdálenosti podél definičních křivek reálných liniových objektů v prostoru (tímto způsobem je možné vyjádřit např. podélný profil)
- `xy` – vyjadřuje dvourozměrný prostor v podobě horizontální roviny xy s hodnotami pozic na dvourozměrných křivkách reprezentujících síťové prvky odpovídajícími měření vzdálenosti podél průmětu definičních křivek reálných liniových objektů do horizontální roviny
- `xyz` – vyjadřuje trojrozměrný prostor s hodnotami pozic na trojrozměrných křivkách reprezentujících síťové prvky odpovídajícími měření vzdálenosti podél definičních křivek reálných liniových objektů v prostoru

Je možné si povšimnout, že počet znaků textového řetězce, který je hodnotou atributu `dimension` zároveň udává, v kolikarozměrném prostoru je síť na příslušné síťové úrovni vyjádřena.

Hodnota atributu `representation` vyjadřuje, zda je reprezentace příslušné síťové úrovně pouze schematické nebo realistické. Tento atribut byl v rámci VMŽI zaveden, aby bylo možné rozlišit mezi jednotlivými způsoby konstruování popisu sítě v prostředí grafického editoru. Atribut `representation` může nabývat následujících hodnot:

- `schematic` – vyjadřuje schematickou reprezentaci, u níž není vyžadována plynulá návaznost křivek reprezentujících síťové prvky ani není nutné, aby rozdíly hodnot pozic na těchto křivkách odpovídaly vzdálenostem po těchto křivkách mezi příslušnými pozicemi měřeným
- `realistic` – vyjadřuje realistickou reprezentaci, u níž je vyžadována plynulá návaznost křivek reprezentujících síťové prvky a je nutné, aby rozdíly hodnot pozic na těchto křivkách odpovídaly vzdálenostem po těchto křivkách mezi příslušnými pozicemi měřeným

2.2.3 NetworkResource

Abstraktní třída `NetworkResource` je obecným konceptem síťového objektu. Síťový objekt je objektem, který může být přímo přiřazen síti a úrovni. Přímým přiřazením rozumějme přiřazení s využitím některé z instancí tříd `NetworkResourceAssignment`, resp. `LevelResourceAssignment`. Každý síťový objekt musí být přiřazen alespoň jedné síti a síťové úrovni. Přiřazení síťového objektu síti a síťové úrovni v mnohých případech závisí také na přiřazení jiných souvisejících síťových objektů síti a síťové úrovni. V souladu s RTM verze 1.1 jsou síťovými objekty síťové prvky, síťové vazby a síťové entity, VMŽI za síťové objekty dále považuje po vzoru RTM verze 1.0 také lokace (u nich však není vyžadováno přímé přiřazení síti a úrovni). Třída `NetworkResource` je třídou RTM. Jedná se o specializaci třídy `NamedResource`, tudíž dědí atributy `id`, `name` a `longname`. Další atributy pro ni nejsou zavedeny.

2.2.4 NetworkLevelAssignment

Instance asociační třídy `NetworkLevelAssignment` slouží k přiřazení síťové úrovně síti. Pro tuto třídu jsou zavedeny atributy `id_LevelNetwork` a `id_Network`. Oba tyto atributy jsou atributy referenčními. Každá instance třídy `NetworkLevelAssignment` je jednoznačně identifikovatelná kombinací hodnot atributů `id_LevelNetwork` a `id_Network`.

Hodnota atributu `id_LevelNetwork` je totožná s hodnotou atributu `id` síťové úrovně, která je síti přiřazována.

Hodnota atributu `id_Network` je totožná s hodnotou atributu `id` sítě, již je síťová úroveň přiřazována.

2.2.5 NetworkResourceAssignment

Instance asociační třídy `NetworkResourceAssignment` slouží k přiřazení síťového objektu síti. Pro tuto třídu jsou zavedeny atributy `id_NetworkResource` a `id_Network`. Oba tyto atributy jsou atributy referenčními. Každá instance třídy `NetworkResourceAssignment` je jednoznačně identifikovatelná kombinací hodnot atributů `id_NetworkResource` a `id_Network`.

Hodnota atributu `id_NetworkResource` je totožná s hodnotou atributu `id` síťového objektu, který je síti přiřazován.

Hodnota atributu `id_Network` je totožná s hodnotou atributu `id` sítě, již je síťový objekt přiřazován.

2.2.6 LevelResourceAssignment

Instance asociační třídy `LevelResourceAssignment` slouží k přiřazení síťového objektu síťové úrovni. Pro tuto třídu jsou zavedeny atributy `id_NetworkResource` a `id_NetworkResource`. Oba tyto atributy jsou atributy referenčními. Každá instance třídy `LevelResourceAssignment` je jednoznačně identifikovatelná kombinací hodnot atributů `id_NetworkResource` a `id_LevelNetwork`.

Hodnota atributu `id_NetworkResource` je totožná s hodnotou atributu `id` síťového objektu, který je síťové úrovni přiřazován.

Hodnota atributu `id_LevelNetwork` je totožná s hodnotou atributu `id` síťové úrovně, již je síťový objekt přiřazován.

2.3 Topology

Blok `Topology` je blokem, jehož třídy umožňují popsat topologii sítě s využitím síťových prvků a vazeb a nabízejí aparát, s jehož pomocí je prvky možné seskupovat do kolekcí. K reprezentaci síťových prvků slouží v rámci VMŽI abstraktní třídy `NetElement`, `CompositionNetElement`, `PositioningNetElement` a terminální třídy `NonLinearElement` a `LinearElement`. Síťové vazby jsou potom reprezentovány v rámci VMŽI abstraktní třídou `Relation` a terminální třídou `PositionedRelation`. Aparát sloužící k vyjádření kolekcí potom zahrnuje v rámci VMŽI abstraktní třídu `ElementPartCollection`, terminální třídy `UnorderedCollection` a `OrderedCollection` a asociační třídy `UnorderedCollectionElement` a `OrderedCollectionElement`. Tento aparát je možné využít k agregaci síťových prvků napříč síťovými úrovněmi.

Při implementaci tematického bloku `Topology` do VMŽI není zohledněna asociace mezi třídami RTM `NetElement` a `Relation` a některé z tříd bloku jsou nově klasifikovány jako abstraktní. Dále dochází k zavedení uvedených asociačních tříd, přičemž do třídy `OrderedCollectionElement` je přesunut atribut `sequence` ze třídy RTM `OrderedCollection`. V ostatních aspektech blok `Topology` VMŽI reflektuje blok `Topology` RTM.

2.3.1 NetElement

Abstraktní třída `NetElement` je obecným konceptem síťového prvku. Síťový prvek je objektem, který figuruje jako prvek v systému topologie sítě vyjádřené na určité síťové úrovni. Na základě příslušnosti ke třídě `NetElement` (resp. k její specializaci) může síťový prvek ve vztahu ke kolekcím síťových prvků vystupovat v roli části. Každý takovýto síťový prvek může vystupovat v roli části pro libovolné množství kolekcí síťových prvků.

Třída `NetElement` je třídou RTM, který ji však na rozdíl od VMŽI nepovažuje za třídu abstraktní. VMŽI nicméně nepředpokládá, že by na základě této třídy byly vytvářeny instance. V rámci RTM existuje asociace mezi třídou `NetElement` a třídou `Relation`, tuto asociaci ale VMŽI nereflektuje. Propojení síťových prvků síťovými vazbami je totiž v souladu s RTM možné vyjádřit také na základě referenčních atributů třídy `PositionedRelation`, odkazujících se navíc na určitou pozici v rámci propojovaných síťových prvků. Tohoto přístupu využívá také VMŽI. Třída `NetElement` je specializací třídy `NetworkResource`, tudíž dědí atributy `id`, `name` a `longname` a veškeré obecné vlastnosti této třídy. Další atributy pro ni nejsou zavedeny.

2.3.2 CompositionNetElement

Abstraktní třída `CompositionNetElement` je obecným konceptem síťového prvku, který navíc oproti vlastnostem získaným na základě příslušnosti k třídě `NetElement` může ve vztahu ke kolekcím síťových prvků vystupovat také v roli celku. Každý takovýto síťový prvek může vystupovat v roli celku ve vztahu k libovolnému množství kolekcí síťových prvků. Třída `CompositionNetElement` je třídou RTM. Jedná se o specializaci třídy `NetElement`, tudíž dědí atributy `id`, `name` a `longname` a veškeré obecné vlastnosti této třídy. Další atributy pro ni nejsou zavedeny.

2.3.3 PositioningNetElement

Abstraktní třída `PositioningNetElement` je obecným konceptem síťového prvku, který navíc oproti vlastnostem získaným na základě příslušnosti k třídě `CompositionNetElement` může být propojen s jiným síťovým prvkem síťovou vazbou (instancí třídy `PositionedRelation`), vázanou na určitou pozici v rámci obou propojovaných síťových prvků a jemuž je přiřazen alespoň jeden souřadnicový systém a mohou vůči němu být lokalizovány síťové entity. Na každý takovýto síťový prvek může být vázáno libovolné množství síťových vazeb.

Třída `PositioningNetElement` je třídou RTM, který ji však na rozdíl od VMŽI nepovažuje za třídu abstraktní. VMŽI nicméně nepředpokládá, že by na základě této třídy byly vytvářeny instance. VMŽI při její implementaci dále zavádí určité změny týkající se přiřazování souřadnicových systémů a navazování vnitřních souřadnic určených k lokalizaci síťových entit k síťovým prvkům. Vnitřní souřadnice síťových prvků vystupují v pojetí VMŽI v podobě atributů přidružených pozic, a to nezávisle na jejich přiřazení souřadnicovým systémům. Lokace síťových entit se neodkazují přímo na síťové prvky, k nimž jsou připojeny, ale k tomuto připojení dochází právě s využitím přidružených

prvků. Třída `PositioningNetElement` je specializací třídy `CompositionNetElement`, tudíž dědí atributy `id`, `name` a `longname` a veškeré obecné vlastnosti této třídy. Další atributy pro ni nejsou zavedeny.

2.3.4 NonLinearElement

Instancemi terminální třídy `NonLinearElement` jsou jednotlivé neliniové síťové prvky. Neliniový síťový prvek je takovým síťovým prvkem, který reprezentuje část reality neliniového charakteru, tedy bod nebo oblast, vystupující na určité síťové úrovni jako prvek systému topologie sítě. Neliniový síťový prvek na rozdíl od liniového síťového prvku nemá orientaci a je v celém rozsahu pokryt vnitřní souřadnicí o hodnotě `0`.

Třída `NonLinearElement` je třídou RTM. Jedná se o specializaci třídy `PositioningNetElement`, tudíž dědí atributy `id`, `name` a `longname` a veškeré obecné vlastnosti této třídy. Další atributy pro ni nejsou zavedeny.

2.3.5 LinearElement

Instancemi terminální třídy `LinearElement` jsou jednotlivé liniové síťové prvky. Liniový síťový prvek je takovým síťovým prvkem, který reprezentuje část reality liniového charakteru, vystupující na určité síťové úrovni jako prvek systému topologie sítě. Liniový síťový prvek má na rozdíl od neliniového síťového prvku definovanou orientaci a je možné v jeho rámci určit vnitřní souřadnici v rozmezí hodnot `0` a `1`. Orientace liniového síťového prvku je dána jeho začátkem a koncem. Začátek liniového síťového prvku se nachází v bodě, kde jeho vnitřní souřadnice nabývá hodnoty `0`. Konec liniového síťového prvku se nachází v bodě, kde jeho vnitřní souřadnice nabývá hodnoty `1`.

Třída `LinearElement` je třídou RTM. Jedná se o specializaci třídy `PositioningNetElement`, tudíž dědí atributy `id`, `name` a `longname` a veškeré obecné vlastnosti této třídy. Dále je pro tuto třídu zaveden atribut `length`. Atribut `lengthje` zaveden VMŽI v souladu s konceptem railML 3 nad rámec RTM 1.1.

Hodnota atributu `length` vyjadřuje délku liniového síťového prvku v metrech. Způsob stanovení délky liniového síťového prvku závisí na hodnotě atributu `dimension` síťové úrovně, jíž je síťový prvek přiřazen. Pro hodnoty atributu `dimension p`, a `xy` se jedná o délku průmětu definiční křivky reálného objektu, který je předobrazem liniového síťového prvku do roviny xy. Pro hodnoty `d`, `pz` a `xyz` se jedná přímo o délku definiční křivky reálného objektu, který je předobrazem liniového síťového prvku, měřenou v prostoru xyz. Z tohoto pravidla vyplývá omezení týkající se přiřazování liniových síťových prvků síťovým úrovním. Liniový síťový prvek přiřazený síťové úrovni s hodnotou atributu `dimension p` nebo `xy` nesmí být zároveň přiřazen žádné síťové úrovni s hodnotou atributu `dimension d`, `pz` nebo `xyz`.

2.3.6 Relation

Abstraktní třída `Relation` je obecným konceptem síťové vazby. Síťová vazba je objektem, který figuruje v roli vazby v systému topologie sítě vyjádřené na určité síťové úrovni.

Třída `Relation` je třídou RTM, který ji však na rozdíl od VMŽI nepovažuje za třídu abstraktní. VMŽI nicméně nepředpokládá, že by na základě této třídy byly vytvářeny instance. V rámci RTM existuje asociace mezi třídou `NetElement` a třídou `Relation`, tuto asociaci ale VMŽI nereflektuje. Propojení síťových prvků síťovými vazbami je totiž v souladu s RTM možné vyjádřit také na základě referenčních atributů třídy `PositionedRelation`, odkazujících se navíc na určitou pozici v rámci propojovaných

síťových prvků. Tohoto přístupu využívá také VMŽI. Třída `Relation` je specializací třídy `NetworkResource`, tudíž dědí atributy `id`, `name` a `longname` a veškeré obecné vlastnosti této třídy. Další atributy pro ni nejsou zavedeny.

2.3.7 PositionedRelation

Instancemi terminální třídy `PositionedRelation` jsou jednotlivé síťové vazby, jejichž prostřednictvím jsou v případě každé instance propojeny vždy právě dva síťové prvky (vyjádřené jako instance tříd `NonLinearElement` nebo `LinearElement`). V rámci každého z nich je síťová vazba vázán a na určitou pozici, jíž je možné vyjádřit jako vnitřní souřadnici, avšak s tím omezením, že síťová vazba může být vázána vždy jen na začátek, nebo na konec síťového prvku. Síťovou vazbu je možné využít pro vyjádření propojení libovolnými dvěma síťovými prvky, reprezentujícími dvojici reálných objektů, mezi nimiž existuje fyzický kontakt.

Třída `PositionedRelation` je třídou RTM. Jedná se o specializaci třídy `Relation`, tudíž dědí atributy `id`, `name` a `longname`. Dále jsou pro tuto třídu zavedeny atributy `id_PositioningNetElement_A`, `id_PositioningNetElement_B`, `positionOnA`, `positionOnB` a `navigability`. Atributy `id_PositioningNetElement_A` a `id_PositioningNetElement_B` jsou atributy referenčními. Atributy `positionOnA`, `positionOnA` a `navigability` jsou atributy RTM.

Hodnota atributu `id_PositioningNetElement_A` je totožná s hodnotou atributu `id` toho ze síťových prvků, k němuž je vázána síťová vazba, který v kontextu této síťové vazby vystupuje v roli A.

Hodnota atributu `id_PositioningNetElement_A` je totožná s hodnotou atributu `id` toho ze síťových prvků, k němuž je vázána síťová vazba, který v kontextu této síťové vazby vystupuje v roli B.

Hodnota atributu `positionOnA` vyjadřuje vnitřní referenci určující pozici v rámci síťového prvku, k němuž je síťová vazba vázána, který v kontextu této síťové vazby vystupuje v roli A. Atribut `positionOnA` Může nabývat následujících hodnot:

- `0`– vyjadřuje navázání síťové vazby na začátek síťového prvku v roli A, použije se také v případě, že je síťový prvek v roli a neliniový
- `1`– vyjadřuje navázání síťové vazby na konec síťového prvku v roli A, nelze použít v případě, že je síťový prvek v roli a neliniový

Hodnota atributu `positionOnB` vyjadřuje vnitřní referenci určující pozici v rámci síťového prvku, k němuž je síťová vazba vázána, který v kontextu této síťové vazby vystupuje v roli B. Atribut `positionOnB` může nabývat následujících hodnot:

- `0`– vyjadřuje navázání síťové vazby na začátek síťového prvku v roli B, použije se také v případě, že je síťový prvek v roli B neliniový
- `1`– vyjadřuje navázání síťové vazby na konec síťového prvku v roli B, nelze použít v případě, že je síťový prvek v roli B neliniový

Hodnota atributu `navigability` vyjadřuje průchodnost síťové vazby, tedy možnost přechodu ze síťového prvku v roli a na síťový prvek v roli B, resp. ze síťového prvku v roli B na síťový prvek v roli A. Fyzický kontakt mezi reálnými objekty reprezentovanými síťovými prvky propojenými síťovou vazbou ještě nutně neznamená, že na sebe tyto prvky navazují také z funkčního hlediska. S přihlédnutím ke skutečnosti, že jednotlivé

reálné objekty reprezentované síťovými prvky jsou zařízeními nebo oblastmi dopravní infrastruktury, jejichž funkcí je umožňovat provoz vozidel, lze toto hledisko nazývat též hlediskem provozním. Atribut `navigability` může nabývat následujících hodnot:

- `ab` – vyjadřuje síťovou vazbu průchozí pouze ve směru ze síťového prvku v roli a na síťový prvek v roli B
- `ba` – vyjadřuje síťovou vazbu průchozí pouze ve směru ze síťového prvku v roli B na síťový prvek v roli A
- `both` – vyjadřuje síťovou vazbu průchozí oběma směry
- `none` – vyjadřuje síťovou vazbu neprůchozí v žádném směru

Při konstruování síťové vazby mezi dvěma síťovými prvky obecně nezáleží na tom, který z dvojice propojovaných síťových prvků má vystupovat v roli a a který v roli B. Pokud má však síťová vazba vyjádřit možnost přechodu mezi těmito síťovými prvky pouze v jednom směru, je nutné hodnotu atributu `navigability` stanovit s ohledem na role v nichž tyto síťové prvky v kontextu síťové vazby vystupují.

2.3.8 ElementPartCollection

Abstraktní třída `ElementPartCollection` je obecným konceptem kolekce síťových prvků. Kolekce síťových prvků je objektem, jemuž jsou přiřazeny síťové prvky, každý z nichž vystupuje ve vztahu k této kolekci v roli její části a jejichž integrací vzniká síťový prvek, vystupující ve vztahu této kolekci v roli celku. Každá kolekce síťových prvků přísluší právě jednomu síťovému prvku, který ve vztahu k ní vystupuje jako celek.

Třída `ElementPartCollection` je třídou RTM, který ji však na rozdíl od VMŽI nepovažuje za třídu abstraktní. VMŽI nicméně nepředpokládá, že by na základě této třídy byly vytvářeny instance. Jedná se o specializaci třídy `BaseObject`, tudíž dědí atribut `id`. Dále je pro tuto třídu zaveden atribut `id_CompositionNetElement`. Tento atribut je atributem referenčním.

Hodnota atributu `id_CompositionNetElement` je totožná s hodnotou atributu `id` síťového prvku, který ve vztahu ke kolekci síťových prvků vystupuje jako celek.

2.3.9 UnorderedCollection

Instancemi třídy `UnorderedCollection` jsou jednotlivé neuspořádané kolekce. Neuspořádaná kolekce je takovou kolekcí síťových prvků, jíž jsou jednotlivé síťové prvky, které ve vztahu k ní vystupují v roli částí, přiřazeny bez stanoveného pořadí. Každé neuspořádané kolekci musí být přiřazen alespoň jeden prvek, který ve vztahu k ní vystupuje v roli části (a to prostřednictvím instance asociační třídy `UnorderedCollectionElement`).

Třída `UnorderedCollection` je třídou RTM. Jedná se o specializaci třídy `ElementPartCollection`, tudíž dědí atribut `id` a `id_CompositionNetElement` a veškeré obecné vlastnosti této třídy. Další atributy pro ni nejsou zavedeny.

2.3.10 OrderedCollection

Instancemi třídy `OrderedCollection` jsou jednotlivé uspořádané kolekce. Uspořádaná kolekce je takovou kolekcí síťových prvků, jíž jsou jednotlivé síťové prvky, které ve vztahu k ní vystupují v roli částí, přiřazeny ve stanoveném pořadí. Každé uspořádané kolekci musí být přiřazen alespoň jeden prvek, který ve vztahu k ní vystupuje v roli části (a to prostřednictvím instance asociační třídy `OrderedCollectionElement`).

Třída `OrderedCollection` je třídou RTM. Jedná se o specializaci třídy `Element-PartCollection`, tudíž dědí atribut `id` a `id_CompositionNetElement` a veškeré obecné vlastnosti této třídy. Další atributy pro ni v rámci VMŽI nejsou zavedeny. V souladu s RTM sice třída `OrderedCollection` obsahuje atribut `sequence`, ten však zjevně nemá smysl uvádět pro uspořádanou kolekci jako celek, ale pouze za účelem přiřazení síťového prvku jako části této kolekce. Proto jej VMŽI zavádí pro asociační třídu `Ordered-CollectionElement`.

2.3.11 UnorderedCollectionElement

Instance asociační třídy `UnorderedCollectionElement` slouží k přiřazení síťového prvku, který ve vztahu k neuspořádané kolekci, jíž je přiřazován, vystupuje v roli její části, této neuspořádané kolekci.

Pro tuto třídu jsou zavedeny atributy `id_NetElement` a `id_UnorderedCollection`. Oba tyto atributy jsou atributy referenčními. Každá instance třídy `Unordered-CollectionElement` je jednoznačně identifikovatelná kombinací hodnot atributů `id_NetElement` a `id_UnorderedCollection`.

Hodnota atributu `id_NetElement` je totožná s hodnotou atributu `id` síťového prvku, který je neuspořádané kolekci přiřazován v roli její části.

Hodnota atributu `id_UnorderedCollection` je totožná s hodnotou atributu `id` neuspořádané kolekce, jíž je síťový prvek v roli její části přiřazován.

2.3.12 OrderedCollectionElement

Instance asociační třídy `OrderedCollectionElement` slouží k přiřazení síťového prvku, který ve vztahu k uspořádané kolekci, jíž je přiřazován, vystupuje v roli její části, této uspořádané kolekci.

Pro tuto třídu jsou zavedeny atributy `id_NetElement`, `sequence` a `id_Ordered-Collection`. Atributy `id_NetElement` a `id_OrderedCollection` jsou atributy referenčními. Atribut `sequence` je zaveden VMŽI za účelem nahradit atribut `sequence` RTM třídy `OrderedCollection`. Každá instance třídy `OrderedCollectionElement` je jednoznačně identifikována kombinací hodnot atributů `id_OrderedCollection` a `sequence`.

Hodnota atributu `id_OrderedCollection` je totožná s hodnotou atributu `id` uspořádané kolekce, jíž je síťový prvek v roli její části přiřazován.

Hodnota atributu `sequence` vyjadřuje pořadí přiřazení síťového prvku v roli části uspořádané kolekce příslušné uspořádané kolekci. Atribut `sequence` může nabývat hodnoty přirozeného čísla od `1` do celkového počtu síťových prvků příslušné uspořádané kolekci v roli její části přiřazených.

Hodnota atributu `id_NetElement` je totožná s hodnotou atributu `id` síťového prvku, který je uspořádané kolekci přiřazován v roli její části.

2.4 PositioningSystem

Blok `PositioningSystem` je blokem, jehož třídy umožňují popsat souřadnicové systémy a jim přiřazené souřadnice. Umožňuje také přiřadit síťový prvek souřadnicovému systému. VMŽI navíc doplňuje možnost vyjádřit natočení síťové entity v rámci souřadnicového systému. Blok je v rámci VMŽI tvořen abstraktní třídou `PositioningSystem` a terminálními třídami `LinearPositioningSystem` a `GeoPositioningSystem` reprezentujícími vlastní souřadnicové systém, dále terminální třídou `LinearAnchorPoint`

a asociačními třídami `PositioningSystemNetElement`, `GeoPointLinearCoordinate`, `GeoPointGeoCoordinate` a `EntityOrientation`.

Při implementaci tematického bloku `PositioningSystem` do VMŽI dochází k přejmenování třídy RTM `GeometricPositioningSystem` na `GeoPositioningSystem`. Pro třídu `LinearAnchorPoint` VMŽI nad rámec RTM umožňuje definovat, který kotvicí bod liniového souřadnicového systému vystupuje v roli dalšího. Souřadnice souřadnicových systémů jsou v rámci VMŽI vyjádřeny s využitím asociačních tříd formou přiřazení geobodu souřadnicovému systému příslušné terminální třídy. Tento přístup byl uplatněn z toho důvodu, že třídy RTM `PositioningSystemCoordinate`, resp. specializované třídy `LinearCoordinate` a `GeoCoordinate`, které VMŽI neimplementuje (resp. implementuje v podobě uvedených asociačních tříd) nejsou specializacemi třídy `BaseObject`, tudíž je v rámci VMŽI není možné považovat za primární objekty, zároveň je však třeba jejich instance nějak identifikovat, k čemuž je kromě souřadnicových systémů využito právě v rámci VMŽI zavedené třídy `GeoPoint` bloku `Location` (identifikace pomocí všech třech souřadnicových složek se přitom nejeví být příliš praktickou). Novým způsobem je v rámci VMŽI řešeno také přiřazení síťového prvku souřadnicovému systému. Místo třídy RTM `AssociatedPositioningSystem` zavádí VMŽI třídu `PositioningSystemNetElement`. Přiřazení vyjádřené od touto třídou VMŽI odpoutává od vnitřních souřadnic, a neimplementuje třídu RTM `IntrinsicCoordinate` (ta je v určitých ohledech nahrazena třídou VMŽI `AssociatedPosition` bloku `Location`). Třída `EntityOrientation` je zavedena VMŽI nad rámec RTM.

2.4.1 PositioningSystem

Abstraktní třída `PositioningSystem` představuje obecný koncept souřadnicového systému. Souřadnicový systém je objektem umožňujícím popsat polohu bodů v prostoru pomocí souřadnic. Každému souřadnicovému systému může být přiřazeno libovolné množství síťových prvků. V rámci každého souřadnicového systému může být v souladu s VMŽI přiřazeno pootočení libovolnému množství síťových entit.

Třída `PositioningSystem` je třídou RTM, který ji však na rozdíl od VMŽI nepovažuje za třídu abstraktní. VMŽI nicméně nepředpokládá, že by na základě této třídy byly vytvářeny instance. Jedná se o specializaci třídy `NamedResource`, tudíž dědí atributy `id`, `name` a `longname`. Další atributy pro ni v rámci VMŽI nejsou zavedeny. V souladu s RTM sice třída `PositioningSystem` obsahuje atributy `validFrom` a `validTo`, aktuální verze VMŽI nicméně časovou platnost objektů, kterou hodnoty těchto atributů vyjadřují, zatím nepodporuje.

2.4.2 LinearPositioningSystem

Instancemi terminální třídy `LinearPositioningSystem` jsou jednotlivé liniové souřadnicové systémy. Liniový souřadnicový systém je souřadnicovým systémem, umožňující popsat polohu bodů v prostoru pomocí souřadnic vztažených k definované ose liniového souřadnicového systému a k němuž může být vázán libovolný počet kotvicích bodů liniového souřadnicového systému. Typicky se jedná o systém staničení s definovanou osou staničení. V rámci VMŽI je souřadnice bodu liniového souřadnicového systému vyjádřena formou přiřazení geobodu liniovému souřadnicovému systému.

Třída `LinearPositioningSystem` je třídou RTM. Jedná se o specializaci třídy `PositioningSystem`, tudíž dědí atributy `id`, `name` a `longname` a veškeré obecné vlastnosti této třídy. Dále jsou pro tuto třídu zavedeny atributy `linearReferencingMethod`, `startMeasure`, `endMeasurea` `units`. Všechny tyto atributy jsou atributy RTM.

Hodnota atributu `linearReferencingMethod` vyjadřuje metodu liniového referencování přidělenou příslušnému liniovému souřadnicovému systému. Atribut `linearReferencingMethod` může nabývat následujících hodnot:

- `absolute` – vyjadřuje využití absolutního liniového referencování
- `relative` – vyjadřuje využití relativního liniového referencování
- `interpolation` – vyjadřuje využití interpolačního liniového referencování

Hodnota atributu `startMeasure` vyjadřuje hodnotu staničení na začátku liniového souřadnicového systému v definovaných jednotkách. Atribut `startMeasure` může nabývat hodnoty reálného čísla, nikoliv však vyšší, než je hodnota atributu `startMeasure` příslušného liniového souřadnicového systému.

Hodnota atributu `endMeasure` vyjadřuje hodnotu staničení na konci liniového souřadnicového systému v definovaných jednotkách. Atribut `endMeasure` může nabývat hodnoty reálného čísla, nikoliv však nižší, než je hodnota atributu `endMeasure` příslušného liniového souřadnicového systému.

Atribut `units` vyjadřuje jednotky staničení liniového souřadnicového systému. V rámci RTM je hodnota atributu `units` vyjádřena ve formě textového řetězce. VMŽI předpokládá vyjádření hodnoty staničení v metrech nebo kilometrech, proto pro atribut `units` zavádí následující hodnoty:

- `m` – vyjadřuje metry
- `km` – vyjadřuje kilometry

2.4.3 LinearAnchorPoint

Instancemi třídy `LinearAnchorPoint` jsou jednotlivé kotvicí body liniového souřadnicového systému. Kotvicí bod liniového souřadnicového systému je objektem, který představuje bod osy liniového souřadnicového systému, jemuž je přidělena hodnota staničení a vzdálenost k následujícímu kotvicímu bodu příslušného liniového souřadnicového systému měřená po ose staničení tohoto liniového souřadnicového systému. Každý kotvicí bod liniového souřadnicového systému je vázán k právě jednomu liniovému souřadnicovému systému. V rámci VMŽI může každému kotvicímu bodu liniového souřadnicového systému příslušet maximálně jeden kotvicí bod liniového souřadnicového systému v roli následujícího a každý kotvicí bod liniového souřadnicového systému může v roli následujícího příslušet také maximálně jednomu kotvicímu bodu liniového souřadnicového systému. Všechny takto spojené kotvicí body liniového souřadnicového systému přitom musejí být vázány k témuž liniovému souřadnicovému systému. Kotvicí body liniového souřadnicového systému je možné využít mimo jiné k modelování skoků ve staničení.

Třída `LinearAnchorPoint` je třídou RTM. Jedná se o specializací třídy `BaseObject`, tudíž dědí atribut `id`. Pro třídu `LinearAnchorPoint` jsou dále zavedeny atributy `id_LinearPositioningSystem`, `anchorName`, `measure`, `id_LinearAnchorPoint_Next` a `measureToNext`. Atributy `id_LinearPositioningSystem` a `id_LinearAnchorPoint_Next` jsou atributy referenčními, přičemž atribut `id_LinearAnchorPoint_Next` je zaveden VMŽI nad rámec RTM. Atributy `anchorName`, `measure` a `measureToNext` jsou atributy RTM.

Hodnota atributu `id_LinearPositioningSystem` je totožná s hodnotou atributu `id` liniového souřadnicového systému, k němuž je kotvicí bod liniového souřadnicového systému vázán.

Hodnota atributu `anchorName` vyjadřuje pojmenování příslušného kotvícího bodu ve formě textového řetězce.

Hodnota atributu `measure` vyjadřuje hodnotu staničení liniového kotvícího bodu v rámci liniového souřadnicového systému, k němuž kotvicí bod liniového souřadnicového systému vázán, v jednotkách staničení tohoto liniového souřadnicového systému. Atribut `measure` může nabývat hodnoty reálného čísla v rozmezí hodnot atributů `startMeasure` a `endMeasure` liniového souřadnicového systému, k němuž je příslušný kotvicí bod liniového souřadnicového systému vázán.

Hodnota atributu `id_LinearAnchorPoint_Next` je totožná s hodnotou atributu `id` kotvicího bodu liniového souřadnicového systému, který následuje po příslušném kotvicím bodu liniového souřadnicového systému. Atribut je zaveden za účelem specifikovat kotvicí bod, k němuž se vztahuje hodnota atributu `measureToNext`. V případě, že po příslušném kotvicím bodě liniového souřadnicového systému již žádný další kotvicí bod liniového souřadnicového systému nenásleduje, nabývá atribut `id_LinearAnchorPoint_Next` hodnoty Null.

Hodnota atributu `measureToNext` vyjadřuje vzdálenost od příslušného kotvicího bodu ke kotvicímu bodu, který po něm následuje, měřenou po ose a udávanou v jednotkách staničení liniového souřadnicového systému, k němuž je příslušný kotvicí bod liniového souřadnicového systému vázán. Atribut `measureToNext` může nabývat hodnoty reálného čísla.

2.4.4 GeoPointLinearCoordinate

Instance asociační třídy `GeoPointLinearCoordinate` slouží k přiřazení souřadnic liniového souřadnicového systému geobodu.

Třída `GeoPointLinearCoordinate` je zavedena VMŽI mimo jiné za účelem nahradit funkci třídy RTM `LinearCoordinate`. Pro tuto třídu jsou zavedeny atributy `id_GeoPoint`, `id_LinearPositioningSystem`, `measure`, `lateralOffset` a `verticalOffset`. Atributy `id_GeoPoint` a `id_LinearPositioningSystem` jsou atributy referenčními. Atributy `measure`, `lateralOffset` a `verticalOffset` jsou atributy RTM, původem ze třídy `LinearCoordinate`. Každá instance třídy `GeoPointLinearCoordinate` je jednoznačně identifikovatelná kombinací hodnot atributů `id_GeoPoint` a `id_LinearPositioningSystem`.

Hodnota atributu `id_GeoPoint` je totožná s hodnotou `id` geobodu, jemuž jsou souřadnice liniového souřadnicového systému přiřazovány.

Hodnota atributu `id_LinearPositioningSystem` je totožná s hodnotou atributu `id` liniového souřadnicového systému, v rámci něhož jsou souřadnice geobodu přiřazovány.

Hodnota atributu `measure` vyjadřuje hodnotu staničení geobodu v rámci liniového souřadnicového systému, jemuž je geobod přiřazován, v jednotkách staničení tohoto liniového souřadnicového systému. Atribut `measure` může nabývat hodnoty reálného čísla v rozmezí hodnot atributů `startMeasure` a `endMeasure` liniového souřadnicového systému, v rámci něhož jsou souřadnice geobodu přiřazovány.

Hodnota atributu `lateralOffset` vyjadřuje boční odsazení geobodu od osy liniového souřadnicového systému v bodě definovaném hodnotou staničení odpovídající hodnotě atributu `measure` příslušného liniového souřadnicového systému v jednotkách tohoto liniového souřadnicového systému. Atribut `lateralOffset` může nabývat hodnoty reálného čísla. Pokud se geobod nachází při pohledu po směru narůstajícího staničení

od tohoto bodu vpravo, je hodnota atributu kladná. Pokud se geobod nachází při pohledu po směru narůstajícího staničení od tohoto bodu vlevo, je hodnota atributu záporná.

Hodnota atributu `verticalOffset` vyjadřuje vertikální odsazení geobodu od osy liniového souřadnicového systému v bodě definovaném hodnotou staničení odpovídající hodnotě atributu measure příslušného liniového souřadnicového systému v jednotkách tohoto liniového souřadnicového systému. Atribut `verticalOffset` může nabývat hodnoty reálného čísla. Pokud se geobod nachází nad tímto bodem, je hodnota atributu kladná. Pokud se geobod nachází pod tímto bodem, je hodnota atributu záporná.

2.4.5 GeoPositioningSystem

Instancemi terminální třídy `GeoPositioningSystem` jsou jednotlivé geometrické (resp. zeměpisné) souřadnicové systémy. Geometrický (resp. zeměpisný) souřadnicový systém je takový souřadnicový systém, umožňující popsat polohu bodů buď pomocí souřadnic kartézské soustavy (v souladu s RTM a VMŽI maximálně trojrozměrné), nebo pomocí sférických souřadnic (zeměpisné šířky, zeměpisné délky a nadmořské výšky). V rámci VMŽI je souřadnice bodu geometrického (resp. zeměpisného) souřadnicového systému vyjádřena formou přiřazení geobodu geometrickému (resp. zeměpisnému) souřadnicovému systému.

Třída `GeoPositioningSystem` je zavedena VMŽI za účelem nahradit funkci třídy RTM `GeometricPositioningSystem`. Zkrácený název byl v rámci VMŽI použit z toho důvodu, aby lépe vystihoval skutečnost, že instance této třídy mohou reprezentovat jak zástupce geometrických souřadnicových systémů, tak zástupce souřadnicových systémů zeměpisných, a také s ohledem na názvy souvisejících tříd VMŽI. Jedná se o specializaci třídy `PositioningSystem`, tudíž dědí atributy `id`, `name` a `longname` a veškeré obecné vlastnosti této třídy. Dále je pro tuto třídu zaveden atribut `crsDefinition`. Tento atribut je atributem RTM, původem ze třídy `GeometricPositioningSystem`.

Hodnota atributu `crsDefinition` vyjadřuje definici souřadnicového systému. Je vyjádřena ve formě textového řetězce. Pro standardní zeměpisné souřadnicové systémy s přiděleným kódem EPSG odpovídá hodnota tohoto atributu kódu EPSG (např. `4326` pro souřadnicový systém WGS 84). Pro takové geometrické (resp. zeměpisné) souřadnicové systémy, které kód EPSG přidělen nemají, je možné využít jiný unikátní textový řetězec.

2.4.6 GeoPointGeoCoordinate

Instance asociační třídy `GeoPointGeoCoordinate` slouží k přiřazení souřadnic geometrického (resp. zeměpisného) souřadnicového systému geobodu.

Třída `GeoPointGeoCoordinate` je zavedena VMŽI mimo jiné za účelem nahradit funkci třídy RTM `GeoCoordinate`. Pro tuto třídu jsou zavedeny atributy `id_GeoPoint`, `id_GeoPositioningSystem`, `x`, `y` a `z`. Atributy `id_GeoPoint` a `id_GeoPositioningSystem` jsou atributy referenčními. Atributy `x`, `y` a `z` jsou atributy RTM, původem ze třídy `GeoCoordinate`. Každá instance třídy `GeoPointGeoCoordinate` je jednoznačně identifikovatelná kombinací hodnot atributů `id_GeoPoint` a `id_GeoPositioningSystem`.

Hodnota atributu `id_GeoPoint` je totožná s hodnotou atributu `id` geobodu, jemuž jsou souřadnice geometrického (resp. zeměpisného) souřadnicového systému přiřazovány.

Hodnota atributu `id_GeoPositioningSystem` je totožná s hodnotou atributu `id` geometrického (resp. zeměpisného) souřadnicového systému, v rámci něhož jsou souřadnice geobodu přiřazovány.

Hodnota atributu `x` vyjadřuje buď x-ovou souřadnici (v případě kartézské soustavy), nebo zeměpisnou délku (v případě sférických souřadnic) geobodu v rámci geometrického (resp. zeměpisného) souřadnicového systému, jemuž je geobod přiřazován. Atribut `x` může nabývat hodnoty reálného čísla a, není-li stanoveno jinak, je udáván v metrech, resp. ve stupních.

Hodnota atributu `y` vyjadřuje buď y-ovou souřadnici (v případě kartézské soustavy), nebo zeměpisnou šířku (v případě sférických souřadnic) geobodu v rámci geometrického (resp. zeměpisného) souřadnicového systému, jemuž je geobod přiřazován. Atribut `y` může nabývat hodnoty reálného čísla a, není-li stanoveno jinak, je udáván v metrech, resp. ve stupních.

Hodnota atributu `z` vyjadřuje buď z-ovou souřadnici (v případě kartézské soustavy), nebo nadmořskou (v případě sférických souřadnic) geobodu v rámci geometrického (resp. zeměpisného) souřadnicového systému, jemuž je geobod přiřazován. Atribut `z` může nabývat hodnoty reálného čísla a, není-li stanoveno jinak, je udáván v metrech.

2.4.7 EntityOrientation

Instance asociační třídy `EntityOrientation` slouží k přiřazení pootočení v rámci souřadnicového systému síťové entitě.

Třída `EntityOrientation` je zavedena VMŽI nad rámec RTM. Pro tuto třídu jsou zavedeny atributy `id_NetEntity`, `id_PositioningSystem`, `deltaHorizontalRotation` a `deltaHorizontalRotation`. Atributy `id_NetEntity` a `id_PositioningSystem` jsou atributy referenčními. Atributy `deltaHorizontalRotation` a `deltaHorizontalRotation` jsou atributy VMŽI. Aktuální verze VMŽI zatím neumožňuje vztáhnout pootočení síťové entity v rámci souřadnicového systému k libovolné lokaci a lokalizačnímu prvku s jejichž využitím je síťová entita lokalizována, proto VMŽI předpokládá, že uvedené hodnoty pootočení jsou platné pro výchozí bod (s případným přihlédnutím k hodnotě atributu `featureOrientation`) prvního lokalizačního prvku nejvíce prioritní lokace příslušné síťové entity. Každá instance třídy `EntityOrientation` je jednoznačně identifikována kombinací hodnot atributů `id_NetEntity` a `id_PositioningSystem`.

Hodnota atributu `id_NetEntity` je totožná s hodnotou atributu `id` síťové entity, jíž je natočení v rámci souřadnicového systému přiřazováno.

Hodnota atributu `id_PositioningSystem` je totožná s hodnotou atributu `id` souřadnicového systému, v rámci něhož je prostorové natočení síťové entitě přiřazováno.

Hodnota atributu `deltaHorizontalRotation` vyjadřuje hodnotu horizontálního pootočení síťové entity vůči jejímu výchozímu směru v rámci příslušného souřadnicového systému ve stupních. Atribut `deltaHorizontalRotation` může nabývat hodnoty reálného čísla.

Hodnota atributu `deltaHorizontalRotation` vyjadřuje hodnotu vertikálního pootočení síťové entity vůči horizontální rovině v rámci příslušného souřadnicového systému ve stupních. Atribut `deltaHorizontalRotation` může nabývat hodnoty reálného čísla.

2.4.8 PositioningSystemNetElement

Instance asociační třídy `PositioningSystemNetElement` slouží k přiřazení síťového prvku souřadnicového systému. Účelem tohoto přiřazení v rámci VMŽI je specifikovat, souřadnice jakých souřadnicových systémů jsou (mají být) přiřazeny geobodům, jimž jsou přiřazeny přidružené pozice spojené s příslušným síťovým prvkem. S využitím instance této třídy je dále možné specifikovat roli síťového prvku v rámci základny přiřazovaného souřadnicového systému.

Třída `PositioningSystemNetElement` je zavedena VMŽI mimo jiné za účelem nahradit funkci třídy RTM `AssociatedPositioningSystem`. Pro tuto třídu jsou zavedeny atributy `positioningNetElement` a `id_PositioningSystem`. Oba tyto atributy jsou atributy referenčními. Každá instance třídy `PositioningSystemNetElement` je jednoznačně identifikovatelná kombinací hodnot atributů `positioningNetElement` a `id_PositioningSystem`.

Hodnota atributu `PositioningSystemNetElement` je totožná s hodnotou `id` síťového prvku, který je souřadnicovému systému přiřazován.

Hodnota atributu `id_PositioningSystem` je totožná s hodnotou `id` souřadnicového systému, jemuž je síťový prvek přiřazován.

2.5 Location

Blok `Location` je blokem, jehož třídy umožňují popsat lokace, s jejichž využitím je možné lokalizovat síťové entity. Je možné definovat buď s využitím pozic (v původním pojetí RTM vnitřních souřadnic) na síťových prvcích, nebo s využitím souřadnic souřadnicových systémů. VMŽI chápe lokaci jako uskupení jednoho nebo více lokalizačních prvků. K reprezentaci lokalizačních prvků a vztahů mezi nimi slouží v rámci VMŽI abstraktní třídy `LocalizationFeature`, `AssociatedFeature`, `GeoFeature`, terminální třídy `AssociatedPosition`, `AssociatedSection`, `GeoPoint` a `GeoPolygonalChain` a asociační třídy `AssociatedPositionAssignment`, `GeoPointAssociatedPosition` a `GeoPolygonalChainGeoPoint`. K vyjádření vlastních lokací a k přiřazení lokalizačních prvků a síťových entit těmto lokacím slouží v rámci VMŽI abstraktní třída `EntityLocation`, terminální třídy `AssociatedLocation` a `GeoLocation` a asociační třídy `AssociatedLocationFeature`, `GeoLocationFeature`, `AssociatedLocationEntity` a `GeoLocationEntity`.

Při implementaci tematického bloku `Location` do VMŽI dochází oproti RTM k nejvýraznějším změnám. Cílem provedených úprav bylo umožnit opakovaně využít tutéž pozici definovanou v rámci síťového prvku, a to i pro konstrukci sekcí vytvářených pro účely liniové a plošné lokace, dokázat odlišit přidruženou lokaci a geolokaci a rozlišit mezi referencí lokace funkční a fyzické. Základní typy lokace, jak je definuje RTM, tedy bodovou, liniovou a plošnou, VMŽI taktéž rozlišuje, avšak v rámci již nikoliv třídou, ale hodnotou atributu `associatedLocationType`, resp. `geoLocationType` příslušných asociačních tříd. Neimplementuje tedy přímo třídy `SpotLocation`, `LinearLocation`, `AreaLocation` a jejich specializace, které RTM zavádí za účelem rozlišit mezi vyjádřením lokace pomocí přidružení síťovému prvku a souřadnic, tedy `SpotLocationIntrinsic`, `SpotLocationCoordinate` a `LinearLocationCoordinate`, ani neimplementuje třídu, jejímž prostřednictvím je možné některé typy lokací svázat se síťovým prvkem tedy třídu `AssociatedNetElement` a její specializace `OrderedAssociatedNetElement`, `AssociatedNetElementIntrinsic` a `AssociatedNetElementCoordinate`, nýbrž původních tříd RTM přejímá pouze třídu

EntityLocation. I ta však má v pojetí VMŽI poněkud jiné vlastnosti, protože jednak získává atribut numberOfFeatures, jednak se stává třídou abstraktní a specializací třídy NetworkResource (jako je tomu v RTM verze 1.0) a jednak instance jejích specializací mohou být sdíleny více síťovými entitami. Struktura bloku Location je s využitím tříd zavedených VMŽI vystavěna zcela nově. Jako celek však původní funkce nesené blokem Location v pojetí RTM zachovává a k tomu nabízí nové. Při lokalizaci síťových entit VMŽI umožňuje nad rámec RTM mimo jiné využít lokalizačních atributů asociačních tříd AssociatedLocationEntity a GeoLocationEntity, které v prvním případě zčásti vycházejí z atributů některých specializací třídy NetEntity uplatněných v railML 3, tímto vyzdvižených na obecnější úroveň.

2.5.1 LocalizationFeature

Abstraktní třída LocalizationFeature představuje obecný koncept lokalizačního prvku. Lokalizační prvek je objektem s definovanou polohou v prostoru, na jehož základě může být zkonstruována lokace.

Třída LocalizationFeature je zavedena VMŽI nad rámec RTM jako obecná třída zastřešující přidružené prvky a geoprvky. Jedná se o specializaci třídy NamedResource, tudíž dědí atributy id, name a longname. Další atributy pro ni nejsou zavedeny.

2.5.2 AssociatedFeature

Abstraktní třída AssociatedFeature představuje obecný koncept přidruženého prvku. Přidružený prvek je lokalizačním prvkem, který je (buď bezprostředně nebo zprostředkovaně) vázán na síťový prvek, díky čemuž je možné určit jeho polohu v prostoru. Každý přidružený prvek může být přiřazen libovolnému množství přidružených lokací, které utváří nebo spoluutváří.

Třída AssociatedFeature je zavedena VMŽI nad rámec RTM jako obecná třída zastřešující přidružené pozice a přidružené sekce. Jedná se o specializaci třídy LocalizationFeature, tudíž dědí atributy id, name a longname. Další atributy pro ni nejsou zavedeny.

2.5.3 AssociatedPosition

Instancemi třídy AssociatedPosition jsou jednotlivé přidružené pozice. Přidružená pozice je přidruženým prvkem, který je bezprostředně vázán k síťovému prvku, a to jako bod ve výsledné pozici, již přidružená pozice vyjadřuje. Každá přidružená pozice je vázána právě k jednomu síťovému prvku. Jedna výsledná pozice na síťovém prvku může být vyjádřena několika různými způsoby s využitím několika různých přidružených pozic. Pokud je přidružená pozice vázána na liniový síťový prvek, přebírá od něj jeho směr v bodě výsledné pozice, již vyjadřuje. Na každou přidruženou pozici může být vázáno libovolné množství přidružených sekcí. Každá přidružená pozice může být přiřazena libovolnému množství geobodů. Každé přidružené pozici může být přiřazeno libovolné množství jiných přidružených pozic.

Třída AssociatedPosition je zavedena VMŽI mimo jiné za účelem nahradit funkci třídy RTM IntrinsicCoordinate a částečně AssociatedPositioningSystem a SpotLocationIntrinsic a zároveň umožnit definovat pozici v rámci síťového prvku nejen relativně (s využitím vnitřní souřadnice) ale také v absolutních jednotkách. Jedná se o specializaci třídy AssociatedFeature, tudíž dědí atributy id, name a longname a veškeré obecné vlastnosti této třídy. Dále jsou pro tuto třídu zavedeny atributy positioningNetElement, intrinsicReference a deltaPosition. Atribut positioningNetElement je atributem referenčním. Atribut intrinsicReference je

zaveden VMŽI za účelem nahradit atribut RTM `intrinsicCoordinate`, původem ze třídy `IntrinsicCoordinate`. Atribut `deltaPosition` je zaveden VMŽI nad rámec RTM.

Hodnota atributu `positioningNetElement` je totožná s hodnotou atributu `id` síťového prvku, k němuž je příslušná přidružená pozice vázaná.

Hodnota atributu `intrinsicReference` vyjadřuje vnitřní referenci v rámci síťového prvku, k němuž je příslušná přidružená pozice vázána. Vnitřní reference je vnitřní souřadnice, která vystupuje v roli relativní složky sloužící k vyjádření výsledné pozice v rámci předmětného síťového prvku. Atribut `intrinsicReference` může nabývat hodnoty `0` pro přidruženou pozici vázanou na neliniový síťový prvek a hodnoty reálného čísla v rozmezí `0` až `1` pro přidruženou pozici vázanou na liniový síťový prvek. Hodnota `0` přitom indikuje začátek síťového prvku a hodnota `1` jeho konec. Pokud je žádoucí pozici v rámci síťového prvku vyjádřit přehledně v absolutních jednotkách, je vhodné zvolit hodnotu tohoto atributu buď `0`, nebo `1`.

Hodnota atributu `deltaPosition` vyjadřuje přírůstek mezi pozicí odpovídající vnitřní referenci v rámci síťového prvku, k němuž je příslušná přidružená pozice vázána, a výslednou pozicí, již je žádoucí s využitím příslušné přidružené pozice vyjádřit, udávaný v metrech. Pokud požadujeme, aby se výsledná pozice nacházela v rámci síťového prvku, k němuž je příslušná přidružená pozice vázána, může atribut `deltaPosition` nabývat hodnoty `0` pro přidruženou pozici vázanou na neliniový síťový prvek a hodnoty reálného čísla v rozmezí $-\texttt{intrinsicReference} \cdot \texttt{length}$ až $(1-\texttt{intrinsicReference}) \cdot \texttt{length}$ pro liniový síťový prvek, kde `length` je hodnota atributu `length` příslušného liniového síťového prvku. Pro typické hodnoty atributu `intrinsicReference` potom platí:

$$\texttt{intrinsicReference} = 0 \Rightarrow \texttt{deltaPosition} \in \langle 0; \texttt{length}\rangle$$

$$\texttt{intrinsicReference} = 1 \Rightarrow \texttt{deltaPosition} \in (-\texttt{length}; 0\rangle$$

Hodnota parametru `position` vyjadřuje výslednou pozici v rámci síťového prvku, k němuž je příslušná přidružená pozice vázána. Pro neliniový síťový prvek je vždy rovna `0` a pro liniový síťový prvek je možné ji vypočítat následujícím způsobem:

$$\texttt{position} = \texttt{intrinsicReference} \cdot \texttt{length} + \texttt{deltaPosition}$$

Pro efektivní práci s VMŽI je doporučeno, aby měl každý neliniový síťový prvek k sobě vázanou přidruženou pozici o hodnotách atributů $\texttt{intrinsicReference} = 0$ a $\texttt{deltaPosition} = 0$ a každý liniový síťový prvek alespoň dvě přidružené pozice, a to o hodnotách atributů $\texttt{intrinsicReference} = 0$ a $\texttt{deltaPosition} = 0$ (tyto hodnoty vyjadřují přidruženou pozici na začátku síťového prvku) a $\texttt{intrinsicReference} = 1$ a $\texttt{deltaPosition} = 0$ (tyto hodnoty vyjadřují přidruženou pozici na konci síťového prvku).

2.5.4 AssociatedSection

Instancemi třídy `AssociatedSection` jsou jednotlivé přidružené sekce. Přidružená sekce je přidruženým prvkem, který je k síťovému prvku vázán zprostředkovaně, a to s využitím dvou přidružených pozic. Obě tyto přidružené pozice musí být vázány na tentýž síťový prvek a vyjadřovat v rámci tohoto síťového prvku dvě různé výsledné pozice. Z uvedených skutečností vyplývá, že se musí jednat o liniový síťový prvek. Tyto dvě

přidružené pozice přidruženou sekci v rámci příslušného liniového síťového prvku ohraničují. Přidružená sekce vymezuje úsek na liniovém síťovém prvku mezi příslušnými přidruženými pozicemi. Od tohoto liniového síťového prvku přebírá přidružená sekce svůj směr, který je tím pádem nezávislý na pořadí (roli) jednotlivých přidružených pozic. (Je sice možné zavést pravidlo, že např. přidružená pozice v roli a vyjadřuje nižší výslednou pozici než přidružená pozice v roli B, zavedení tohoto pravidla by však vyžadovalo znemožnit takové dodatečné změny atributů `intrinsicReference` a `deltaPosition` přidružených pozic a atributu `length` liniových síťových prvků, které by mohly způsobit, že dojde k porušení pravidla.)

Třída `AssociatedSection` je zavedena VMŽI za účelem částečně nahradit funkci třídy RTM `AssociatedNetElement`, resp. `AssociatedNetElementIntrinsic`. Jedná se o specializaci třídy `AssociatedFeature`, tudíž dědí atributy `id`, `name` a `longname` a veškeré obecné vlastnosti této třídy. Dále jsou pro tuto třídu zavedeny atributy `id_AssociatedPosition_A` a `id_AssociatedPosition_B`. Oba tyto atributy jsou atributy referenčními.

Hodnota atributu `id_AssociatedPosition_A` je totožná s hodnotou atributu `id` té z přidružených pozic, k níž je vázána přidružená sekce, která v kontextu této přidružené sekce vystupuje v roli A.

Hodnota atributu `id_AssociatedPosition_B` je totožná s hodnotou atributu `id` té z přidružených pozic, k níž je vázána přidružená sekce, která v kontextu této přidružené sekce vystupuje v roli B.

2.5.5 AssociatedPositionAssignment

Instance asociační třídy `AssociatedPositionAssignment` slouží k přiřazení přidružené pozice jiné přidružené pozici. Toto přiřazení vyjadřuje vzájemné ztotožnění přidružených pozic a má smysl být zaváděno zejména napříč síťovými úrovněmi (tedy za účelem ztotožnění dvou různých přidružených pozic, každá z nichž je vázána na síťový prvek jiné síťové úrovně). Je tedy možné předpokládat, že jedna z přiřazovaných přidružených pozic patří k síťovému prvku úrovně s abstraktnější reprezentací, zatímco druhá z přiřazovaných přidružených pozic patří k síťovému prvku úrovně s realističtější reprezentací.

Pro tuto třídu jsou zavedeny atributy `id_AssociatedPosition_A` a `id_AssociatedPosition_B`. Oba tyto atributy jsou atributy referenčními. Každá instance třídy `AssociatedPositionAssignment` je jednoznačně identifikovatelná kombinací hodnot atributů `id_AssociatedPosition_A` a `id_AssociatedPosition_B`.

Hodnota atributu `id_AssociatedPosition_A` je totožná s hodnotou atributu `id` jedné z přiřazovaných přidružených lokací. Pokud je možné tuto skutečnost posoudit, jedná se o tu z přidružených lokací, která je vázána na síťový prvek přiřazený síťové úrovni s vyšším stupněm abstrakce (menší dimenzí, resp. méně realistickou reprezentací).

Hodnota atributu `id_AssociatedPosition_B` je totžná s hodnotou atributu `id` druhé z přiřazovaných přidružených lokací. Pokud je možné tuto skutečnost posoudit, jedná se o tu z přidružených lokací, která je vázána na síťový prvek přiřazený síťové úrovni s nižším stupněm abstrakce (vyšší dimenzí, resp. realističtější reprezentací).

2.5.6 GeoFeature

Abstraktní třída `GeoFeature` představuje obecný koncept geoprvku. Geoprvek je lokalizačním prvkem, který na rozdíl od přidruženého prvku není přímo vázán na žádný

síťový prvek a jehož poloha v prostoru je obecně na síťových prvcích nezávislá. Lze ji vyjádřit s využitím souřadnic definovaných souřadnicových systémů. Geoprvek možné ztotožnit s prvkem GIS. Každý geoprvek může být přiřazen libovolnému množství geolokací, které utváří nebo spoluutváří.

Třída `GeoFeature` je zavedena VMŽI nad rámec RTM jako obecná třída zastřešující geobody a geořetězce. Jedná se o specializaci třídy `LocalizationFeature`, tudíž dědí atributy `id`, `name` a `longname`. Další atributy pro ni nejsou zavedeny.

2.5.7 GeoPoint

Instancemi třídy `GeoPoint` jsou jednotlivé geobody. Geobod je bodovým geoprvkem, jemuž je možné přímo přiřadit souřadnice definovaných souřadnicových systémů, a to jak geometrických (resp. zeměpisných), tak liniových. Každému geobodu mohou být přiřazeny souřadnice v rámci libovolného množství souřadnicových systémů. Každému geobodu může být přiřazeno libovolné množství přidružených pozic.

Třída `GeoPoint` je zavedena VMŽI mimo jiné za účelem nahradit funkci třídy RTM `SpotLocationCoordinate`. Jedná se o specializaci třídy `GeoFeature`, tudíž dědí atributy `id`, `name` a `longname` a veškeré obecné vlastnosti této třídy. Další atributy pro ni nejsou zavedeny.

2.5.8 GeoPolygonalChain

Instancemi třídy `GeoPolygonalChain` jsou jednotlivé geořetězce. Geořetězec je liniovým nebo plošným geoprvkem jíž jsou ve stanoveném pořadí přiřazeny geobody, které určují jeho tvar a polohu v prostoru. Každému geořetězci musí být přiřazeny alespoň dva geobody. Geořetězec může být buď otevřený (lomená čára určující linii) nebo uzavřený (polygon určující plochu).

Třída `GeoPolygonalChain` je zavedena VMŽI mimo jiné za účelem nahradit funkci třídy RTM `LinearLocationCoordinate`. Jedná se o specializaci třídy `GeoFeature`, tudíž dědí atributy `id`, `name` a `longname` a veškeré obecné vlastnosti této třídy. Dále je pro tuto třídu zaveden atribut `isClosed`. Jedná se o atribut VMŽI.

Hodnota atributu `isClosed` vyjadřuje, zda příslušná instance reprezentuje geořetězec, který je uzavřený, nebo nikoliv. Jedná se o kontrolní atribut. Uvedenou skutečnost lze zjistit také na základě existence instance `GeoPolygonalChainGeoPoint`, která příslušnému řetězci přiřazuje některý geobod, o hodnotě atributu `sequence` = `0`. Atribut `isClosed` může nabývat následujících hodnot:

- `0` – vyjadřuje skutečnost, že příslušná instance reprezentuje otevřený geořetezec
- `1` – vyjadřuje skutečnost, že příslušná instance reprezentuje uzavřený geořetězec

2.5.9 GeoPolygonalChainGeoPoint

Instance třídy `GeoPolygonalChainGeoPoint` slouží k přiřazení geobodu geořetězci.

Třída `GeoPolygonalChainGeoPoint` je zavedena VMŽI nad rámec RTM. Pro tuto třídu jsou zavedeny atributy `id_GeoPolygonalChain`, `sequence` a `id_GeoPoint`. Atributy `GeoPolygonalChainGeoPoint` a `id_GeoPoint` jsou atributy referenčními. Atribut `sequence` je zaveden VMŽI za účelem umožnit identifikovat geobod přiřazený příslušnému geořetězci na základě pořadí, které v tomto geořetězci zaujímá. Každá instance třídy `GeoPolygonalChainGeoPoint` je jednoznačně identifikovatelná kombinací hodnot atributů `id_GeoPolygonalChain` a `sequence`.

Hodnota atributu `id_GeoPolygonalChain` je totožná a hodnotou atributu `id` geořetězce, jemuž je geobod přiřazován.

Hodnota atributu `sequence` vyjadřuje pořadí přiřazení geobodu příslušnému geořetězci. Pro otevřený geořetězec sestavený z n geobodů nabývá atribut sequence hodnoty přirozeného čísla od `0` do n. Pro uzavřený geořetězec sestavený z n geobodů nabývá atribut `sequence` hodnoty `0` pro první a zároveň poslední vrchol a hodnoty přirozeného čísla od `1` do n pro ostatní vrcholy.

Hodnota atributu `id_GeoPoint` je totožná s hodnotou atributu `id` geobodu, který je přiřazován geořetězci.

2.5.10 GeoPointAssociatedPosition

Instance asociační třídy `GeoPointAssociatedPosition` slouží k přiřazení přidružené pozice geobodu. Toto přiřazení vyjadřuje prostorové ztotožnění přidružené pozice s geobodem. Je díky němu možné zprostředkovaně vyjádřit polohu přidružené pozice s využitím souřadnic souřadnicových systémů a také vzájemně propojit různé přidružené pozice, které mají totožnou polohu v prostoru (typicky se jedná o přidružené pozice na koncích na sebe navazujících síťových prvků).

Třída `GeoPointAssociatedPosition` je zavedena VMŽI nad rámec RTM. Pro tuto třídu jsou zavedeny atributy `idAssociatedPosition` a `id_GeoPoint`. Oba tyto atributy jsou atributy referenčními. Každá instance třídy `GeoPointAssociatedPosition` je jednoznačně identifikována kombinací hodnot atributů `idAssociatedPosition` a `id_GeoPoint`.

Hodnota atributu `idAssociatedPosition` je totožná s hodnotou atributu `id` přidružené pozice, která je geobodu přiřazována.

Hodnota atributu `id_GeoPoint` je totožná a hodnotou atributu `id` geobodu, jemuž je přidružená pozice přiřazována.

2.5.11 EntityLocation

Abstraktní třída `EntityLocation` představuje obecný koncept lokace. Lokace je objektem, který prostorově vymezuje výskyt síťové entity v závislosti na konkrétním použití může nabývat různých podob, významů a míry abstrakce. V rámci RTM přísluší každá lokace právě jedné lokalizované síťové entitě. VMŽI však na rozdíl od RTM zavádí koncept sdílených lokací a lokalizačních prvků, díky kterému může být jedna lokace přiřazena více síťovým entitám a zároveň se každá lokace může skládat z několika lokalizačních prvků, které mohou být společné několika lokacím.

Třída `EntityLocation` je třídou RTM, ale v kontextu VMŽI má poněkud odlišné vlastnosti. V rámci VMŽI se jedná o specializací třídy `NamedResource` (jako je tomu na rozdíl od RTM verze 1.1 v RTM verze 1.0), tudíž dědí atributy `id`, `name` a `longname` a veškeré obecné vlastnosti této třídy. Dále pro tuto třídu zaveden atribut `numberOfFeatures`. Jedná se o atribut VMŽI.

Hodnota atributu `numberOfFeatures` vyjadřuje počet lokalizačních prvků, které jsou příslušné lokaci přiřazeny. Atribut `numberOfFeatures` může nabývat hodnoty přirozeného čísla. Jedná se kontrolní atribut. Uvedenou skutečnost lze zjistit také jako počet instancí třídy `AssociatedLocationFeature` nebo `GeoLocationFeature`, hodnota jejichž atributu `id_AssociatedFeature`, resp. `id_GeoFeature` je totožná s hodnotou atributu `id` příslušné lokace.

2.5.12 AssociatedLocation

Instancemi terminální třídy `AssociatedLocation` jsou jednotlivé přidružené lokace. Přidružená lokace je takovou lokací, která se vztahuje k některému síťovému prvku nebo k některým síťovým prvkům, a to tím způsobem, že jí je přiřazen přidružený prvek vázaný k síťovému prvku nebo několik přidružených prvků vázaných k síťovým prvkům. Každé přidružené lokaci je přiřazen alespoň jeden přidružený prvek, maximálně (a obvykle) však právě tolik přidružených prvků, kolik síťových prvků přidružená lokace (zčásti nebo zcela) pokrývá. Charakter přiřazovaných přidružených prvků (tj. zda se jedná o přidružené pozice nebo sekce) a příp. podmínky jejich přiřazování přitom závisí na typu přidružené lokace.

Třída `AssociatedLocation` je zavedena VMŽI za účelem nahradit funkci tříd RTM `SpotLocation`, `LinearLocation` a `AreaLocation` (specializace třídy RTM `EntityLocation`). Tyto třídy RTM umožňují vyjádřit lokaci síťových entit vůči síťovým prvkům buď přímo (v případě třídy `SpotLocation`, resp. její specializace `SpotLocationIntrinsic`) nebo prostřednictvím třídy `AssociatedNetElement`, resp. jejích specializací `OrderedAssociatedNetElement` a `AssociatedNetElementIntrinsic` (v případě tříd `LinearLocation` a `AreaLocation`). VMŽI na rozdíl od RTM zařazuje všechny instance reprezentující lokace, mají-li charakter lokace přidružené, do společné třídy `AssociatedLocation` nezávisle na tom, zda se jedná o lokaci bodovou, liniovou nebo plošnou. Jedná se o specializaci třídy `EntityLocation`, tudíž dědí atributy `id`, `name`, `longname` a `numberOfFeatures` a veškeré obecné vlastnosti této třídy. Dále je pro tuto třídu zaveden atribut `associatedLocationType`. Atribut `associatedLocationType` je zveden VMŽI za účelem nahradit kategorizační funkci tříd RTM `SpotLocation`, `LinearLocation` a `AreaLocation`.

Hodnota atributu `associatedLocationType` vyjadřuje typ přidružené lokace. Atribut `associatedLocationType` může nabývat následujících hodnot:

- `spot` – vyjadřuje bodovou přidruženou lokaci (tento typ lokace využívá jako lokalizačních prvků přidružených pozic)
- `linear` – vyjadřuje liniovou přidruženou lokaci (tento typ lokace využívá jako lokalizačních prvků přidružených sekcí, je vyžadována jejich postupná návaznost)
- `area` – vyjadřuje plošnou přidruženou lokaci (tento typ lokace využívá jako lokalizačních prvků přidružených sekcí, není vyžadována jejich postupná návaznost)

2.5.13 AssociatedLocationFeature

Instance asociační třídy `AssociatedLocationFeature` slouží k přiřazení přidruženého prvku (tj. přidružené pozice nebo sekce) přidružené lokaci.

Pro tuto třídu jsou zavedeny atributy `id_AssociatedLocation`, `sequence`, `id_AssociatedFeature` a `featureOrientation`. Atributy `id_AssociatedLocation` a `id_AssociatedFeature` jsou atributy referenčními. Atribut `sequence` je zaveden VMŽI za účelem nahradit atribut `sequence` RTM třídy `OrderedAssociatedNetElement`, kterou VMŽI neimplementuje. Atribut `featureOrientation` je zaveden VMŽI za účelem nahradit atribut `keepsOrientation` RTM třídy `AssociatedNetElement`, kterou VMŽI neimplementuje. Každá instance třídy `AssociatedLocationFeature` je jednoznačně identifikována kombinací hodnot atributů `id_AssociatedLocation` a `sequence`.

Hodnota atributu `id_AssociatedLocation` je totožná s hodnotou atributu `id` přidružené lokace, jíž je přidružený prvek přiřazován.

Hodnota atributu `sequence` vyjadřuje pořadí přiřazení přidruženého prvku příslušné přidružené lokaci. Atribut `sequence` může nabývat hodnoty přirozeného čísla od `0` do hodnoty atributu `numberOfFeatures` příslušné přidružené lokace. Přidružený prvek, který je přidružené lokaci přiřazován za využití instance třídy `AssociatedLocationFeature`, hodnota jejíhož atributu `sequence` je rovna `1`, vystupuje ve vztahu k této přidružené lokaci v roli referenčního přidruženého prvku. Pořadí přiřazení přidruženého prvku přidružené lokaci dále nabývá na významu při konstrukci liniové přidružené lokace, u níž je vyžadováno, aby na sebe jednotlivé přidružené sekce v pořadí definovaném atributem `sequence` jednotlivých instancí třídy `AssociatedLocationFeature` navazovaly.

Hodnota atributu `id_AssociatedFeature` je totožná s hodnotou atributu `id` přidruženého prvku, jež je příslušné přidružené lokaci přiřazován.

Atribut `featureOrientation` vyjadřuje, zda přidružené lokaci přiřazovaný přidružený prvek v rámci tohoto přiřazení respektuje orientaci síťového prvku, k němuž je vázaný, nebo nikoliv. Atribut `featureOrientation` může nabývat následujících hodnot:

- `1` – vyjadřuje orientaci přidruženého prvku shodnou s orientací síťového prvku (odpovídá hodnotě atributu RTM `keepsOrientation` = `1`)
- `-1` – vyjadřuje orientaci přidruženého prvku opačnou, než je orientace síťového prvku (odpovídá hodnotě atributu RTM `keepsOrientation` = `0`)
- `0` – uvádí se v případech, kdy orientaci přidruženého prvku nemá smysl uvažovat (přidružený prvek je vázán na neliniový síťový prvek)

2.5.14 AssociatedLocationEntity

Instance asociační třídy `AssociatedLocationEntity` slouží k přiřazení síťové entity přidružené lokaci.

Pro tuto třídu jsou zavedeny atributy `id_NetEntity`, `id_AssociatedLocation`, `lateralSide`, `lateralDistance`, `verticalSide`, `verticalDistance`, `functionalLocationReference`, `functionalLocationReference`, `applicationDirection` a `locationPriority`. Atributy `id_NetEntity` a `id_AssociatedLocation` jsou atributy referenčními. Atributy `lateralSide`, `lateralDistance`, `verticalSide` a `verticalDistance` jsou zavedeny VMŽI po vzoru railML 3 nad rámec RTM. Atribut `applicationDirection` je zaveden VMŽI za účelem nahradit atribut `applicationDirection` RTM tříd `SpotLocation` a `LinearLocation`. Atributy `functionalLocationReference`, `functionalLocationReference` a `locationPriority` jsou zavedeny VMŽI nad rámec RTM. Každá instance třídy `AssociatedLocationEntity` je jednoznačně identifikována kombinací hodnot atributů `id_NetEntity` a `id_AssociatedLocation`.

Hodnota atributu `id_NetEntity` je totožná s hodnotou atributu `id` síťové entity, která je přidružené lokaci přiřazována.

Hodnota atributu `id_AssociatedLocation` je totožná s hodnotou atributu `id` přidružené lokace, jíž je síťová entita přiřazována.

Hodnota atributu `lateralSide` vyjadřuje, na které straně vůči příslušné přidružené lokaci se z horizontálního hlediska příslušná síťová entita nachází. Pro posouzení strany je přitom zásadní orientace přidružené lokace, která je dána orientací jednotlivých přidružených prvků v rámci jejich přiřazení příslušné přidružené lokaci (závisí na orientaci síťového prvku, k němuž je příslušná síťová entita připojována a hodnotě atributu `featureOrientation` instance třídy `AssociatedLocationFeature`, jejímž prostřednictvím má být toto připojení realizováno). Atribut `lateralSide` může nabývat následujících hodnot:

- `-1` – vyjadřuje skutečnost, že se síťová entita nachází vlevo od příslušné přidružené lokace
- `0` – vyjadřuje skutečnost, že je síťová entita svou lokací s příslušnou přidruženou lokací v horizontálním směru ztotožněna, příp. že se nachází jak vlevo, tak vpravo od příslušné přidružené lokace, také se uplatní, pokud nemá smysl stranu z horizontálního hlediska smysl uvažovat
- `1` – vyjadřuje skutečnost, že se síťová entita nachází vpravo od příslušné přidružené lokace

Hodnota atributu `lateralDistance` vyjadřuje boční vzdálenost lokace síťové entity od přidružené lokace, jíž je síťová entita přiřazována, v metrech. Atribut `lateralDistance` může nabývat hodnoty nezáporného reálného čísla.

Hodnota atributu `verticalSide` vyjadřuje, na které straně vůči příslušné přidružené lokaci se z vertikálního hlediska příslušná síťová entita nachází. Atribut `verticalSide` může nabývat následujících hodnot:

- `-1` – vyjadřuje skutečnost, že se síťová entita nachází pod příslušnou přidruženou lokací
- `0` – vyjadřuje skutečnost, že se síťová entita nachází buď v úrovni příslušné přidružené lokace, nebo jak pod ní, tak nad ní
- `1` – vyjadřuje skutečnost, že se síťová entita nachází nad příslušnou přidruženou lokací

Hodnota atributu `verticalDistance` vyjadřuje vertikální vzdálenost lokace síťové entity od přidružené lokace, jíž je síťová entita přiřazována, v metrech. Atribut `verticalDistance` může nabývat hodnoty nezáporného reálného čísla.

Hodnota atributu `functionalLocationReference` vyjadřuje referenci funkční lokace. Atribut `functionalLocationReference` může nabývat následujících hodnot:

- `none` – vyjadřuje skutečnost, že v rámci příslušného přiřazení síťové entity přidružené lokaci nedochází k definování funkční lokace
- `source` – vyjadřuje skutečnost, že v rámci příslušného přiřazení síťové entity přidružené lokaci dochází k definování funkční lokace ve zdroji, tedy je možné funkční lokaci síťové entity přímo ztotožnit s příslušnou přidruženou lokací
- `target` – vyjadřuje skutečnost, že v rámci příslušného přiřazení síťové entity přidružené lokaci dochází k definování funkční lokace v cíli, tedy je možné funkční lokaci síťové entity ztotožnit s lokací získanou na základě příslušné přidružené lokace po aplikaci hodnot atributů `lateralSide`, `lateralDistance`, `verticalSide` a `verticalDistance` (nejedná se o typický případ)

- `range` – vyjadřuje skutečnost, že v rámci příslušného přiřazení síťové entity přidružené lokaci dochází k definování funkční lokace v rozmezí mezi zdrojem a cílem (nejedná se o typický případ)

V případě potřeby mohou být definovány i další hodnoty atributu `functionalLocationReference`.

Hodnota atributu `functionalLocationReference` vyjadřuje referenci fyzické lokace. Atribut `functionalLocationReference` může nabývat následujících hodnot:

- `none` – vyjadřuje skutečnost, že v rámci příslušného přiřazení síťové entity přidružené lokaci nedochází k definování fyzické lokace
- `source` – vyjadřuje skutečnost, že v rámci příslušného přiřazení síťové entity přidružené lokaci dochází k definování fyzické lokace ve zdroji, tedy je možné fyzickou lokaci síťové entity přímo ztotožnit s příslušnou přidruženou lokací
- `target` – vyjadřuje skutečnost, že v rámci příslušného přiřazení síťové entity přidružené lokaci dochází k definování fyzické lokace v cíli, tedy je možné fyzickou lokaci síťové entity ztotožnit s lokací získanou na základě příslušné přidružené lokace po aplikaci hodnot atributů atributů `lateralSide`, `lateralDistance`, `verticalSide` a `verticalDistance`
- `range` – vyjadřuje skutečnost, že v rámci příslušného přiřazení síťové entity přidružené lokaci dochází k definování fyzické lokace v rozmezí mezi zdrojem a cílem

V případě potřeby mohou být definovány i další hodnoty atributu `functionalLocationReference`.

Hodnota atributu `applicationDirection` vyjadřuje směr, pro nějž jsou atributy síťové entity, která je přiřazována přidružené lokaci (resp. její samotná existence z funkčního hlediska), platné. Pro posouzení směru platnosti je přitom zásadní orientace přidružené lokace, která je dána orientací jednotlivých přidružených prvků v rámci jejich přiřazení příslušné přidružené lokaci (závisí na orientaci síťového prvku, k němuž je příslušná síťová entita připojována a hodnotě atributu `featureOrientation` instance třídy `AssociatedLocationFeature`, jejímž prostřednictvím má být toto připojení realizováno). Atribut `applicationDirection` může nabývat následujících hodnot:

- `1` – vyjadřuje platnost pouze v normálním směru (odpovídá hodnotě atributu RTM `applicationDirection` = `normal`)
- `-1` – vyjadřuje platnost pouze v opačném směru (odpovídá hodnotě atributu RTM `applicationDirection` = `reverse`)
- `0` – vyjadřuje platnost obousměrnou (odpovídá hodnotě atributu RTM `applicationDirection` = `both`), také se uplatní, pokud nemá smysl směr platnosti uvažovat

Hodnota atributu `locationPriority` vyjadřuje míru prioritizace příslušné přidružené lokace v rámci přiřazení všech lokací příslušné síťové entitě. Atribut `locationPriority` může nabývat hodnoty přirozeného čísla. Čím nižšího čísla atribut `locationPriority` nabývá, tím vyšší prioritu vyjadřuje. Prioritizaci v rámci přiřazování síťových entit přidruženým lokacím je možné uplatnit např. při vykreslování síťových entit na základě lokací.

2.5.15 GeoLocation

Instancemi terminální třídy *GeoLocation* jsou jednotlivé geolokace. Geolokace je takovou lokací, která je založena na geoprvku nebo geoprvcích, umístěných v geometrickém (resp. zeměpisném) prostoru. Každé geolokaci je přiřazen alespoň jeden geoprvek. Charakter přiřazovaných geoprvků (tj. zda se jedná o geobody, otevřené geořetězce nebo uzavřené geořetězce) a příp. podmínky jejich přiřazování přitom závisí na typu geolokace.

Třída `GeoLocation` je zavedena VMŽI za účelem nahradit funkci tříd RTM `SpotLocationCoordinate` a `LinearLocationCoordinate`, které se vztahují k lokalizaci síťových entit s využitím souřadnic souřadnicových systémů. VMŽI na rozdíl od RTM zařazuje všechny instance reprezentující lokace, mají-li charakter lokace geometrické (resp. zeměpisné), do společné třídy `GeoLocation` nezávisle na tom, zda se jedná o lokaci bodovou, liniovou nebo plošnou. Jedná se o specializaci třídy `EntityLocation`, tudíž dědí atributy `id`, `name`, `longname` a `numberOfFeatures` a veškeré obecné vlastnosti této třídy. Dále je pro tuto třídu zaveden atribut `geoLocationType`. Atribut `geoLocationType` je zaveden VMŽI za účelem nahradit kategorizační funkci tříd RTM `SpotLocationCoordinate` a `LinearLocationCoordinate`, svou podstatou odpovídajícím lokalizaci s využitím prvků GIS typu bod a linie a umožnit odlišit také lokalizaci s využitím prvků GIS typu plocha.

Hodnota atributu `geoLocationType` vyjadřuje typ geolokace. Vzhledem k tomu, že je geolokace realizována s využitím geoprvku nebo jejich skupiny, je typ geolokace určen v souladu s označením tříd geoprvků standardně využívaných v GIS. Atribut `geoLocationType` může nabývat následujících hodnot:

- `point` – vyjadřuje geolokaci s využitím geobodu nebo geobodů, odpovídajících prvkům GIS typu bod
- `line` – vyjadřuje geolokaci s využitím otevřeného řetězce geobodů nebo otevřených řetězců geobodů, odpovídajících prvkům GIS typu linie
- `area` – vyjadřuje geolokaci s využitím uzavřeného řetězce geobodů nebo uzavřených řetězců geobodů, odpovídajících prvkům GIS typu plocha

2.5.16 GeoLocationFeature

Instance asociační třídy `GeoLocationFeature` slouží k přiřazení geoprvku (tj. geobodu nebo řetězci geobodů) geolokaci.

Pro tuto třídu jsou zavedeny atributy `id_GeoLocation`, `sequence`, `id_GeoFeature` a `featureOrientation`. Atributy `id_GeoLocation` a `id_GeoFeature` jsou atributy referenčními. Atribut `sequence` je zaveden VMŽI po vzoru třídy `AssociatedLocationFeature` za účelem umožnit identifikovat geoprvek přiřazený dané geolokaci v rámci všech přiřazení geoprvků této geolokaci stejným způsobem jako přidružený prvek přiřazený dané přidružené lokaci v rámci všech přiřazení přidružených prvků této přidružené lokaci. Tento koncept umožňuje jednu geolokaci vyjádřit s využitím více geoprvků. Atribut `featureOrientation` je zaveden VMŽI po vzoru třídy `AssociatedLocationFeature` za účelem umožnit vyjádřit orientaci geoprvku v rámci jeho přiřazení geolokaci stejným způsobem jako orientaci přidruženého prvku v rámci jeho přiřazení přidružené lokaci. Každá instance třídy `GeoLocationFeature` je jednoznačně identifikována kombinací hodnot atributů `id_GeoLocation` a `sequence`.

Hodnota atributu `id_GeoLocation` je totožná s hodnotou atributu `id` geolokace, jíž je geoprvek přiřazován.

Hodnota atributu `sequence` vyjadřuje pořadí přiřazení geoprvku příslušné geolokaci. Atribut `sequence` může nabývat hodnoty přirozeného čísla od `0` do hodnoty atributu `numberOfFeatures` příslušné geolokace. Geoprvek, který je geolokaci přiřazován za využití instance třídy `GeoLocationFeature`, hodnota jejíhož atributu `sequence` je rovna `1`, vystupuje ve vztahu k této geolokaci v roli referenčního geoprvku.

Hodnota atributu `id_GeoFeature` je totožná s hodnotou atributu `id` geoprvku, jež je příslušné geolokaci přiřazován.

Atribut `featureOrientation` vyjadřuje, zda geolokaci přiřazovaný geoprvek v rámci tohoto přiřazení respektuje výchozí orientaci geoprvku, k němuž je vázaný, nebo nikoliv. Orientaci geoprvku má smysl uvažovat v případě, že je geoprvkem geořetězec. Potom je jeho výchozí orientace dána vzrůstajícím pořadím geobodů v rámci přiřazení příslušnému geořetězci. Atribut `featureOrientation` může nabývat následujících hodnot:

- `1` – vyjadřuje orientaci přidruženého prvku shodnou s výchozí orientací geoprvku
- `-1` – vyjadřuje orientaci přidruženého prvku opačnou, než je výchozí orientace geoprvku
- `0` – uvádí se v případech, kdy orientaci geoprvku nemá smysl uvažovat (geoprvkem je geobod)

2.5.17 GeoLocationEntity

Instance asociační třídy `GeoLocationEntity` slouží k přiřazení síťové entity geolokaci.

Pro tuto třídu jsou zavedeny atributy `id_NetEntity`, `id_GeoLocation`, `lateralSide`, `lateralDistance`, `verticalSide`, `verticalDistance`, `functionalLocationReference`, `functionalLocationReference`, `applicationDirectiona locationPriority`. Atributy `id_NetEntity` a `id_GeoLocationjsou` atributy referenčními. Atributy `lateralSide`, `lateralDistance`, `verticalSide`, `verticalDistance`, `functionalLocationReference`, `functionalLocationReference`, `applicationDirectiona locationPriorityjsou` zavedeny VMŽI po vzoru třídy `AssociatedLocationFeatureza` účelem umožnit vyjádřit lokalizační parametry v rámci přiřazení síťové entity geolokaci obdobným způsobem jako v rámci přiřazení síťové entity přidružené lokaci. Každá instance třídy `GeoLocationEntity` je jednoznačně identifikována kombinací hodnot atributů `id_NetEntity` a `id_GeoLocation`.

Hodnota atributu `id_NetEntity` je totožná s hodnotou atributu `id` síťové entity, která je geolokaci přiřazována.

Hodnota atributu `id_GeoLocation` je totožná s hodnotou atributu `id` geolokace, jíž je síťová entita přiřazována.

Hodnota atributu `lateralSide` vyjadřuje, na které straně vůči příslušné geolokaci se z horizontálního hlediska příslušná síťová entita nachází. Pro posouzení strany je přitom zásadní orientace přidružené lokace, která je dána orientací jednotlivých přidružených prvků v rámci jejich přiřazení příslušné přidružené lokaci (závisí na výchozí orientaci geoprvku, k němuž je příslušná síťová entita připojována a hodnotě atributu `featureOrientation` instance třídy `GeoLocationFeature`, jejímž prostřednictvím má být toto připojení realizováno). Atribut `lateralSide` může nabývat následujících hodnot:

- `-1` – vyjadřuje skutečnost, že se síťová entita nachází vlevo od příslušné geolokace
- `0` – vyjadřuje skutečnost, že je síťová entita svou lokací s příslušnou geolokací v horizontálním směru ztotožněna, příp. že se nachází jak vlevo, tak vpravo od příslušné

geolokace, také se uplatní, pokud nemá smysl stranu z horizontálního hlediska smysl uvažovat

- `1` – vyjadřuje skutečnost, že se síťová entita nachází vpravo od příslušné geolokace

Hodnota atributu `lateralDistance` vyjadřuje boční vzdálenost lokace síťové entity od geolokace, jíž je síťová entita přiřazována, v metrech. Jedná se o šířku obalové křivky geolokace vynesenou v horizontálním směru v závislosti na hodnotě atributu `lateralSide`. Atribut `lateralDistance` může nabývat hodnoty nezáporného reálného čísla.

Hodnota atributu `verticalSide` vyjadřuje, na které straně vůči příslušné geolokaci se z vertikálního hlediska příslušná síťová entita nachází. Atribut `verticalSide` může nabývat následujících hodnot:

- `-1` – vyjadřuje skutečnost, že se síťová entita nachází pod příslušnou geolokací
- `0` – vyjadřuje skutečnost, že se síťová entita nachází buď v úrovni příslušné geolokace, nebo jak pod ní, tak nad ní
- `1` – vyjadřuje skutečnost, že se síťová entita nachází nad příslušnou geolokací

Hodnota atributu `verticalDistance` vyjadřuje vertikální vzdálenost lokace síťové entity od geolokace, jíž je síťová entita přiřazována, v metrech. Jedná se o výšku vynesenou vůči geolokaci ve vertikálním směru v závislosti na hodnotě atributu `lateralSide`. Atribut `verticalSide` může nabývat hodnoty nezáporného reálného čísla.

Hodnota atributu `functionalLocationReference` vyjadřuje referenci funkční lokace. Atribut `functionalLocationReference` může nabývat následujících hodnot:

- `none` – vyjadřuje skutečnost, že v rámci příslušného přiřazení síťové entity geolokaci nedochází k definování funkční lokace
- `source` – vyjadřuje skutečnost, že v rámci příslušného přiřazení síťové entity geolokaci dochází k definování funkční lokace ve zdroji, tedy je možné funkční lokaci síťové entity přímo ztotožnit s příslušnou geolokací (nejedná se o typický případ)
- `target` – vyjadřuje skutečnost, že v rámci příslušného přiřazení síťové entity geolokaci dochází k definování funkční lokace v cíli, tedy je možné funkční lokaci síťové entity ztotožnit s lokací získanou na základě příslušné přidružené lokace po aplikaci hodnot atributů `lateralSide`, `lateralDistance`, `verticalSide` a `verticalDistance` (nejedná se o typický případ)
- `range` – vyjadřuje skutečnost, že v rámci příslušného přiřazení síťové entity geolokaci dochází k definování funkční lokace v rozmezí mezi zdrojem a cílem (nejedná se o typický případ)

V případě potřeby mohou být definovány i další hodnoty atributu `functionalLocationReference`.

Hodnota atributu `functionalLocationReference` vyjadřuje referenci fyzické lokace. Atribut `functionalLocationReference` může nabývat následujících hodnot:

- `none` – vyjadřuje skutečnost, že v rámci příslušného přiřazení síťové entity geolokaci nedochází k definování fyzické lokace (nejedná se o typický případ)
- `source` – vyjadřuje skutečnost, že v rámci příslušného přiřazení síťové entity geolokaci dochází k definování fyzické lokace ve zdroji, tedy je možné fyzickou lokaci síťové entity přímo ztotožnit s příslušnou přidruženou lokací

- `target` – vyjadřuje skutečnost, že v rámci příslušného přiřazení síťové entity geolokaci dochází k definování fyzické lokace v cíli, tedy je možné fyzickou lokaci síťové entity ztotožnit s lokací získanou na základě příslušné geolokace po aplikaci hodnot atributů `lateralSide`, `lateralDistance`, `verticalSide` a `verticalDistance` (nejedná se o typický případ)
- `range` – vyjadřuje skutečnost, že v rámci příslušného přiřazení síťové entity geolokaci dochází k definování fyzické lokace v rozmezí mezi zdrojem a cílem

V případě potřeby mohou být definovány i další hodnoty atributu `functionalLocationReference`.

Hodnota atributu `applicationDirection` vyjadřuje směr, pro nějž jsou atributy síťové entity, která je přiřazována geolokaci (resp. její samotná existence z funkčního hlediska), platné. Pro posouzení směru platnosti je přitom zásadní orientace geolokace, která je dána orientací jednotlivých geoprvků v rámci jejich přiřazení příslušné geolokaci (závisí výchozí orientaci geoprvku, k němuž je příslušná síťová entita připojována a hodnotě atributu `featureOrientation` instance třídy `GeoLocationFeature`, jejímž prostřednictvím má být toto připojení realizováno). Atribut `applicationDirection` může nabývat následujících hodnot:

- `1` – vyjadřuje platnost pouze v normálním směru
- `-1` – vyjadřuje platnost pouze v opačném směru
- `0` – vyjadřuje platnost v obou směrech, také se uplatní, pokud nemá smysl směr platnosti uvažovat

Hodnota atributu `locationPriority` vyjadřuje míru prioritizace příslušné geolokace v rámci přiřazení všech lokací příslušné síťové entitě. Atribut `locationPriority` může nabývat hodnoty přirozeného čísla. Čím nižšího čísla atribut `locationPriority` nabývá, tím vyšší prioritu vyjadřuje. Prioritizaci v rámci přiřazování síťových entit přidruženým lokacím je možné uplatnit např. při vykreslování síťových entit na základě lokací.

2.6 NetEntity

Blok `NetEntity` je blokem, jehož třídy na obecné úrovni reprezentují síťové entity, tedy objekty a vlastnosti železniční infrastruktury, lokalizovatelné vůči topologii sítě nebo s využitím souřadnicových systému, a v rámci VMŽI také vztahy mezi nimi. Blok je tvořen v rámci VMŽI abstraktní třídou `NetEntity`, u níž se předpokládá, že její terminální specializace jsou definovány v rámci konkrétního rozšíření modelu a třídou `NetEntityRelation`, která v závislosti na konkrétním rozšíření může fungovat buď jako terminální, nebo mít definované jiné terminální specializace (a sama být případně abstraktní).

Při implementaci tematického bloku `NetEntity` do VMŽI není zohledněna třída RTM `LocatedNetEntity`, jejímiž instancemi jsou v rámci RTM lokalizované síťové entity, a to z důvodu možného vzniku nejednoznačnosti přiřazení objektu terminální třídě v případě rozšíření VMŽI o specializace třídy `NetEntity`, které by vedlo ke vzniku komplikací při implementaci VMŽI v prostředí grafického editoru s vrstvami strukturalizovanými na základě dědičnosti tříd VMŽI. Informaci o tom, zda je příslušná instance specializace třídy `NetEntity` lokalizovaná, lze odvodit na základě existence odpovídající instance

třídy `AssociatedLocationEntity`, resp. `GeoLocationEntity`. Třídu RTM `NetEntity` VMŽI implementuje s tím, že ji nově klasifikuje jako abstraktní. Třídu `NetEntityRelation` zavádí VMŽI nově nad rámec RTM.

2.6.1 NetEntity

Abstraktní třída `NetEntity` vyjadřuje obecný koncept síťové entity. Síťová entita je objektem, který může vystupovat na určité síťové úrovni nebo síťových úrovních a být přidruženě nebo geometricky (resp. zeměpisně) lokalizován.

Třída `NetEntity` je třídou RTM, který ji však na rozdíl od VMŽI nepovažuje za třídu abstraktní. VMŽI nicméně nepředpokládá, že by na základě této třídy byly vytvářeny instance. Třída `NetworkResource` je specializací třídy `NetworkResource`, tudíž dědí atributy `id`, `name`, a `longname` a veškeré obecné vlastnosti této třídy. Další atributy pro ni nejsou zavedeny.

2.6.2 NetEntityRelation

Instancemi terminální třídy `NetEntityRelation` jsou jednotlivé vztahy síťových entit. Vztah síťových entit je takovým objektem, který slouží k přiřazení síťové entity jiné síťové entitě.

Třída `NetEntityRelation` je zavedena VMŽI nad rámec RTM. Jedná se o specializaci třídy `BaseObject`, tudíž dědí atribut `id`. Dále jsou pro tuto třídu zavedeny atributy `id_NetEntity_A`, `id_NetEntity_b` a `reletionType`. Atributy `id_NetEntity_A` a `id_NetEntity_b` jsou atributy referenčními. Atribut `reletionType` je zaveden VMŽI nad rámec RTM.

Hodnota atributu `id_NetEntity_A` je totožná s hodnotou atributu `id` té síťové entity, která v rámci příslušného vztahu síťových entit vystupuje v roli A.

Hodnota atributu `id_NetEntity_b` je totožná s hodnotou atributu `id` té síťové entity, která v rámci příslušného vztahu síťových entit vystupuje v roli B.

Hodnota atributu `reletionType` vyjadřuje typ vztahu mezi síťovými entitami. Atribut `reletionType` může nabývat následujících hodnot:

- `aIsPartOfB` – vyjadřuje skutečnost, že síťová entita, která v rámci příslušného vztahu mezi síťovými entitami vystupuje v roli a je součástí síťové entity, která v rámci příslušného vztahu mezi síťovými entitami vystupuje v roli B
- `aIsChildOfB` – vyjadřuje skutečnost, že síťová entita, která v rámci příslušného vztahu mezi síťovými entitami vystupuje v roli a je potomkem síťové entity, která v rámci příslušného vztahu mezi síťovými entitami vystupuje v roli B
- `aIsPropertyOfB` – vyjadřuje skutečnost, že síťová entita, která v rámci příslušného vztahu mezi síťovými entitami vystupuje v roli a je vlastnictvím síťové entity, která v rámci příslušného vztahu mezi síťovými entitami vystupuje v roli B
- `aDelimitsB` – vyjadřuje skutečnost, že síťová entita, která v rámci příslušného vztahu mezi síťovými entitami vystupuje v roli a ohraničuje síťovou entitu, která v rámci příslušného vztahu mezi síťovými entitami vystupuje v roli B

V případě potřeby mohou být definovány i další hodnoty atributu `reletionType`, které se mohou uplatnit také pro popisování instancí specializací třídy `NetEntityRelation`, zavedených pro jednotlivá specifická rozšíření modelu.

Kapitola 3
Systém topologického popisu sítě

V rámci této kapitoly budou diskutovány možnosti vyjádření topologického popisu železniční (resp. drážní) infrastruktury obecněji ze systémového hlediska. Ačkoliv je přístup RTM a VMŽI je pro tento popis cennou inspirací, je žádoucí, aby zde uvedený teoretický koncept byl uplatnitelný uplatnitelný i v širším kontextu.

Vymezenou část železniční (resp. drážní) infrastruktury, která je předmětem našeho zájmu, budeme označovat jako síť n. Skutečnosti, že na síť n nahlížíme jako na síť železniční (resp. drážní) infrastruktury, jsou poplatné i uváděné příklady a související komentáře. Obecné principy týkající topologického popisu sítě n lze nicméně uplatnit i pro potřeby popisu sítí jiného charakteru.

Síť n budeme chápat jako objekt reálného světa, na němž identifikujeme systém popisu sítě $\mathbb{R}$. Nad systémem popisu sítě $\mathbb{R}$ lze uvažovat na zcela obecné úrovni s využitím nástrojů teorie systémů. Podrobíme-li systém $\mathbb{R}$ funkční dekompozici, můžeme jako jednu z jeho částí oddělit subsystém topologického popisu sítě $\mathbb{T}$. S ohledem na charakter úloh řešených v rámci této práce se právě tento subsystém systému $\mathbb{R}$ stává hlavním předmětem našeho zájmu a nahlížíme na něj jako na samostatný systém. V okolí systému topologického popisu sítě $\mathbb{T}$ můžeme identifikovat prvky ostatních subsystémů systému $\mathbb{R}$, ale také prvky topologického popisu jiných částí železniční (resp. drážní) infrastruktury, které nejsou součástí sítě n.

3.1 Objekty

K vyjádření různých skutečností týkajících se systému topologického popisu sítě $\mathbb{T}$ a jeho okolí budeme využívat objekty.

Množinu objektů je obecně možné označit jako $\mathcal{O}$. Množinu $\mathcal{O}$ lze rozepsat jako

$$\mathcal{O} = \{o_1, o_2, \dots o_k\},$$

kde každý $o \in \mathcal{O}$, tedy o_i, $i \in \{1, 2, \dots k\}$, je objekt množiny $\mathcal{O}$ a k je celkový počet objektů množiny $\mathcal{O}$.

3.2 Veličiny

K vyjádření různých kvantitativních i kvalitativních skutečností týkajících se objektů budou používány veličiny. Veličiny budeme chápat na vysoce obecné úrovni jako vlastnosti, které mohou nabývat hodnot z předem definovaných oborů. Nemusí se nutně jednat o číselné hodnoty, ale také např. o textové hodnoty definované výčtem přípustných hodnot. Veličiny budeme vyjadřovat ve stanovených jednotkách, které zahrneme do jejich definice. Veličinu aplikovanou v prostředí softwarového produktu tak bude možné vyjádřit jako atribut.

Z hlediska proměnlivosti hodnoty veličiny je možné rozlišit následující typy veličin:

- neměnná veličina (konstanta)
- proměnná veličina

Podle vztahu k objektům je možné rozlišit následující typy veličin:

- vstupní veličina
- výstupní veličina
- stavová veličina
- parametr

Veličiny budou využívány zejména k vyjádření různých polohopisných a popisných charakteristik objektů systému. Objekty stejné třídy budou popisovány těmi samými veličinami. Za předpokladu, že je množina $\mathcal{O}$ množinou objektů stejné třídy, které mají být popisovány stejnými veličinami, můžeme pro množinu $\mathcal{O}$ zavést množinu těchto veličin, kterou označíme jako $\mathcal{M}(\mathcal{O})$.

Množinu $\mathcal{M}(\mathcal{O})$ lze rozepsat jako

$$\mathcal{M}(\mathcal{O}) = \{m_1(\mathcal{O}), m_2(\mathcal{O}), \dots m_h(\mathcal{O})\}\,,$$

kde každá $m(\mathcal{O}) \in \mathcal{M}(\mathcal{O})$, tedy $m_j(\mathcal{O})$, $j \in \{1, 2, \dots h\}$, je veličina, kterou mají být popsány objekty množiny $\mathcal{O}$ a h je celkový počet veličin, kterými mají být objekty množiny $\mathcal{O}$ popsány.

Budeme-li pro množinu objektů $\mathcal{O}$ zavádět množinu veličin $\mathcal{M}(\mathcal{O})$, jako jednu z veličin $m \in \mathcal{O}$ zavedeme takovou veličinu, na základě jejíž hodnoty může být každý objekt $o \in \mathcal{O}$ jednoznačně identifikován. Veličinu vystupující v této roli budeme nazývat identifikátor a označovat jako ι. V praxi mohou hodnoty této veličiny pro jednotlivé objekty vyjadřovat index, podle něhož je možné tyto objekty množiny $\mathcal{O}$, resp. třídy jejímiž jsou instancemi, řadit, nebo se může jednat např. o hodnoty univerzálně jedinečného identifikátoru UUID. Pro účely operací prováděných s objekty množiny $\mathcal{O}$ bude každý objekt $o \in \mathcal{O}$ reprezentován právě hodnotou svého identifikátoru ι.

Jako další z veličin zavedeme takovou veličinu, na základě jejíž hodnoty může být pro každý objekt $o \in \mathcal{O}$ jednoznačně určena třída objektu, jíž je objekt o instancí. Tuto veličinu budeme nazývat příslušnost k třídě a označovat jako ϵ. Další veličiny budou zaváděny na základě příslušnosti objektu k třídě a v závislosti na konkrétním aplikačním případu w.

3.3 Hodnoty veličin

Pro každou veličinu m je nutné jednoznačně vymezit, kterých hodnot může nabývat. Tyto hodnoty budeme nazývat přípustné hodnoty veličiny m. Uvedené vymezení může být buď obecně platné, nebo může záviset na konkrétním aplikačním případů w. Hodnotu, kterou veličina m nabývá pro objekt o zapíšeme jako $m(o)$. Přiřazení hodnoty $m(o)$, kterou veličina m nabývá pro objekt o, veličině m potom zapíšeme následujícím způsobem:

$$v = v(m, o) = (m, m(o))\,.$$

Množinu přiřazení hodnot, kterých konkrétní veličina $m \in \mathcal{M}(\mathcal{O})$ nabývá pro objekty množiny $\mathcal{O}$, těmto objektům budeme označovat jako $\mathcal{V}(m, \mathcal{O})$.

Množinu $\mathcal{V}(m, \mathcal{O})$ lze rozepsat jako

$$\mathcal{V}(m, \mathcal{O}) = \{v(m, o_1), v(m, o_2), \dots v(m, o_k)\}\,,$$

kde každé $v(m,o) \in \mathcal{V}(m,\mathcal{O})$, tedy $v(m,o_i)$, $i \in \{1,2,\dots k\}$, je přiřazení hodnoty, které veličina m nabývá pro objekt $o \in \mathcal{O}$, tedy o_i, a k je celkový počet objektů množiny $\mathcal{O}$.

Množinu přiřazení hodnot, které veličiny množiny $\mathcal{M}(\mathcal{O})$ nabývají pro konkrétní objekt $o \in \mathcal{O}$, těmto veličinám budeme označovat jako $\mathcal{V}(\mathcal{M}(\mathcal{O}),o)$.

Množinu $\mathcal{V}(\mathcal{M}(\mathcal{O}),o)$ lze rozepsat jako

$$\mathcal{V}(\mathcal{M}(\mathcal{O}),o) = \{v(m_1(\mathcal{O}),o), v(m_2(\mathcal{O}),o), \dots v(m_h(\mathcal{O}),o)\},$$

kde každé $v(m,o) \in \mathcal{V}(\mathcal{M}(\mathcal{O}),o)$, tedy $v\big(m_j,o\big)$, $j \in \{1,2,\dots h\}$, je přiřazení hodnoty, které veličina $m \in \mathcal{M}(\mathcal{O})$, tedy m_j, nabývá pro objekt $o \in \mathcal{O}$, a h je celkový počet veličin, jimiž jsou objekty množiny $\mathcal{O}$ popsány. Pro účely operací prováděných s objekty množiny $\mathcal{O}$ budeme předpokládat, že pro každý objekt $o \in \mathcal{O}$ je množina přiřazení $\mathcal{V}(\mathcal{M}(\mathcal{O}),o)$ dostupná na základě hodnoty $\iota(o)$.

S využitím přiřazení hodnot veličinám, která jsou prvky množiny $\mathcal{V}(m,\mathcal{O})$ a $\mathcal{V}(\mathcal{M}(\mathcal{O}),o)$ je možné sestavit matici $\mathfrak{O}$. Matice $\mathfrak{O}$ je matice hodnot, které nabývají veličiny množiny $\mathcal{M}(\mathcal{O})$ pro jednotlivé objekty množiny $\mathcal{O}$. Jedná se o matici $k \times h$, jejímž řádkům odpovídají jednotlivé objekty množiny $\mathcal{O}$ a jejímž sloupcům odpovídají jednotlivé veličiny množiny $\mathcal{M}(\mathcal{O})$. K jednotlivým řádcích matice $\mathfrak{O}$ je možné nalézt informaci o tom, který objekt $o \in \mathcal{O}$ příslušný řádek reprezentuje. Jedná se o hodnotu $\iota(o)$. K jednotlivým sloupcům matice je nutné mít připojenou informaci o tom, kterou veličinu $m(\mathcal{O}) \in \mathcal{M}(\mathcal{O})$ příslušný sloupec reprezentuje.

3.4 Síť

Budeme předpokládat, že jednotlivé (zpravidla rozsáhlejší) oblasti železniční infrastruktury reálného světa je možné reprezentovat jako sítě. Množinu sítí budeme obecně označovat jako $\mathcal{N}$. Množinu $\mathcal{N}$ lze rozepsat jako

$$\mathcal{N} = \{n_1, n_2, \dots n_k\},$$

kde každá síť $n \in \mathcal{N}$, tedy n_i, $i \in \{1,2,\dots k\}$, je síť množiny $\mathcal{N}$ a k je celkový počet sítí množiny $\mathcal{N}$.

Pro účely každé úlohy řešené v rámci konkrétního aplikačního případe w budeme pracovat s jednou konkrétní sítí n, pro níž zavádíme systém topologického popisu sítě $\mathbb{T}$, který je subsystémem systému popisu sítě $\mathbb{R}$. Pro potřeby identifikace systému topologického popisu sítě $\mathbb{T}$ (resp. též jeho nadsystému $\mathbb{R}$) na objektu reálného světa, je nezbytné jednoznačně vymezit tu část železniční (resp. drážní) infrastruktury, která má být sítí n reprezentována. O tomto vymezení rozhodujeme v roli pozorovatele na základě účelu, vzhledem k němuž se příslušný výsek železniční (resp. drážní) infrastruktury stává předmětem našeho zájmu. Tento účel závisí na konkrétním aplikačním případu w, v jehož rámci hodláme úlohy nad systémem $\mathbb{T}$ řešit.

Považujeme-li železniční (resp. drážní) infrastrukturu za neměnnou v čase, resp. nahlížíme-li na ni v jednom konkrétním časovém okamžiku relevantním v kontextu sledovaného účelu, jedná se o zejména prostorové vymezení. S ohledem na charakter úloh řešených v rámci této práce jsou pro toto vymezení zásadní zejména možnosti rozčlenění popisu infrastruktury na prvky jejího topologického popisu. Vlastní realizace prostorového vymezení oblasti reprezentované sítí n může být ale poplatná i různým dalším aspektům souvisejícím se sledovaným účelem v kontextu aplikačního případu w. Mohou být zohledněny např. technické parametry, provozní určení, administrativní členění apod.

Již při prostorovém vymezování této oblasti, jakožto objektu reálného světa, na němž má být identifikován systém topologického popisu sítě $\mathbb{T}$ (resp. též jeho nadsystém $\mathbb{R}$), je vhodné mít rámcovou představu o rozlišovacích úrovních, které mají být pro systém topologického popisu sítě $\mathbb{T}$ zavedeny, a o třídách a funkčních typech prvků, které mají na jednotlivých rozlišovacích úrovních figurovat. Určení tříd a funkčních typů těchto prvků je důležité také pro specifikaci typu přípustné struktury systému topologického popisu sítě $\mathbb{T}$, kterou chceme na jednotlivých zavedených rozlišovacích úrovních rozpoznat.

Se zavedením několika různých rozlišovacích úrovní systému $\mathbb{T}$ (které je možné zároveň považovat za rozlišovací úrovně nadsystému $\mathbb{R}$), uvažujeme z toho důvodu, že se míra podrobnosti, s jakou na systém topologického popisu sítě $\mathbb{T}$ (resp. na systém popisu sítě $\mathbb{R}$), za různých okolností nahlížíme, liší. Zavedení několika rozlišovacích úrovní je zároveň nutným předpokladem k tomu, aby bylo možné realizovat transformaci struktury systému $\mathbb{T}$ napříč těmito rozlišovacími úrovněmi, což je cílem této práce.

Výše uvedené představy se opět odvíjejí od účelového pohledu závislého na konkrétním aplikačním případu w. Jsou důležité mimo jiné pro to, aby bylo možné z prostorového hlediska jednoznačně učit hranici mezi oblastí reprezentovanou sítí n a navazujícími částmi železniční (resp. drážní) infrastruktury, které již nejsou bezprostředním předmětem našeho zájmu, a definovat tak také rozhraní mezi systémem popisu sítě $\mathbb{T}$ a jeho topologickým okolím, které je zároveň okolím systému popisu sítě $\mathbb{R}$.

3.5 Okolí

Okolí systému popisu sítě $\mathbb{R}$, který je nadsystémem systému topologického popisu sítě $\mathbb{T}$, reprezentuje ty části železniční, resp. drážní, infrastruktury, které sice sousedí s oblastí reprezentovanou sítí n, ale nejsou její přímou součástí. Je možné předpokládat, že je toto okolí strukturované podobným způsobem jako samostatný systém $\mathbb{R}$, ale již nespadá do oblasti našeho přímého zájmu. Může produkovat výstupy, které jsou vstupy systému $\mathbb{R}$ a naopak přijímat vstupy, které jsou výstupy systému $\mathbb{R}$.

Z pohledu vlastního systému topologického popisu sítě $\mathbb{T}$ lze rozlišit okolí dvojího charakteru. Topologické okolí systému topologického popisu sítě $\mathbb{T}$ reprezentuje ty části železniční, resp. drážní, infrastruktury, které sice sousedí s oblastí reprezentovanou sítí n, ale nejsou její přímou součástí, z topologického hlediska. Je tvořeno prvky topologického popisu železniční (resp. drážní infrastruktury). Je tudíž možné uvažovat, že se zároveň jedná o tu část okolí systému $\mathbb{R}$, která je strukturovaná podobným způsobem jako systém topologického popisu sítě $\mathbb{T}$. Může produkovat některé výstupy, které jsou vstupy systému $\mathbb{T}$ a naopak přijímat některé vstupy, které jsou výstupy systému $\mathbb{T}$. Jedná se o takové výstupy a vstupy, které mohou být vzájemně předávány právě mezi prvky topologického popisu železniční (resp. drážní) infrastruktury.

Oproti tomu netopologické okolí systému topologického popisu sítě $\mathbb{T}$ reprezentuje ty části železniční, resp. drážní, infrastruktury, které sice jsou součástí sítě n, ale není možné nebo vhodné je reprezentovat přímo s využitím systému topologického popisu sítě $\mathbb{T}$. Je tvořeno jinými prvky, než prvky topologického popisu železniční (resp. drážní) infrastruktury. Jedná se o tu část systému $\mathbb{R}$, která je strukturovaná rozdílným způsobem než systém $\mathbb{T}$. Může opět produkovat některé výstupy, které jsou vstupy systému $\mathbb{T}$ a naopak přijímat některé vstupy, které jsou výstupy systému $\mathbb{T}$, jedná se však o podstatně rozdílné vstupy a výstupy než v případě topologického okolí systému topologického popisu sítě $\mathbb{T}$, které obecně mezi prvky topologického popisu železniční (resp. drážní) infrastruktury být vzájemně předávány nemohou.

3.6 Rozlišovací úrovně

Na rozlišovací úroveň nahlížíme jako na stupeň podrobnosti, na němž je vyjádřena struktura systému. S ohledem na charakter úloh řešených v rámci této práce budou jednotlivé rozlišovací úrovně navrhovány tak, aby byl jejich návrh v rámci uvažovaného aplikačního případu w poplatný potřebám vyjádření struktury s systému topologického popisu sítě $\mathbb{T}$. Může být přitom přihlédnuto také k potřebám vyjádření struktury systému popisu sítě $\mathbb{R}$, který je nadsystémem systému $\mathbb{T}$. Pro různé aplikační případy se mohou požadavky na stupeň podrobnosti jednotlivých zaváděných rozlišovacích úrovní a jejich celkový počet lišit, stejně jako způsob, jakým je na nich síť n vyjádřena, na čemž může záviset i typ přípustné struktury na jednotlivých rozlišovacích úrovních.

Množinu rozlišovacích úrovní budeme obecně označovat jako $\mathcal{L}$. Množinu $\mathcal{L}$ lze rozepsat jako

$$\mathcal{L} = \{l_1, l_2, \dots l_k\},$$

kde každá rozlišovací úroveň $l \in \mathcal{L}$, tedy l_i, $i \in \{1, 2, \dots k\}$, je rozlišovací úroveň množiny $\mathcal{L}$ a k je celkový počet rozlišovacích úrovní množiny $\mathcal{L}$.

Množinu všech zavedených rozlišovacích úrovní systému popisu sítě $\mathbb{R}$ budeme označovat jako $\mathcal{L}_{\mathbb{R}}$. Množinu všech zavedených rozlišovacích úrovní systému topologického popisu sítě $\mathbb{T}$ budeme označovat jako $\mathcal{L}_{\mathbb{T}}$. Pro systém $\mathbb{T}$ budeme zpravidla zavádět stejné rozlišovací úrovně jako pro jeho nadsystém $\mathbb{R}$. Tuto skutečnost můžeme zapsat jako

$$\mathcal{L}_{\mathbb{T}} = \mathcal{L}_{\mathbb{R}}.$$

Budeme-li o zavedené rozlišovací úrovni systému $\mathbb{T}$ pojednávat v kontextu určité role, ve které v rámci uvažovaného aplikačního případu w vystupuje, použijeme pro ni odpovídající označení, s jehož pomocí je možné identifikovat také příslušnou strukturu systému $\mathbb{T}$, její komponenty a případně další související objekty. Za zavedenou rozlišovací úroveň systému $\mathbb{T}$ budeme považovat každou rozlišovací úroveň l^{E}, na níž je známá struktura s^{E} systému $\mathbb{T}$, nebo na níž je záměrem strukturu s^{E} systému $\mathbb{T}$ získat. Pro každý aplikační případ w je při zavádění rozlišovací úrovně l^{E} třeba strukturu s^{E} buď přímo identifikovat, nebo alespoň určit, jaký typ struktury je pro rozlišovací úroveň l^{E} přípustný. Jsou-li tyto předpoklady pro rozlišovací úroveň l^{E} splněny, lze zapsat

$$l^{\mathrm{E}} \in \mathcal{L}_{\mathbb{T}}.$$

V kontextu každého aplikačního případu w, v rámci něhož jsou řešeny jednotlivé úlohy integrace a dezintegrace prvků systému topologického popisu sítě $\mathbb{T}$, předpokládáme, že ve vztahu k tomuto aplikačnímu případu w jedna ze zavedených rozlišovacích úrovní vystupuje v roli referenční rozlišovací úrovně. Rozlišovací úroveň ve vztahu k danému aplikačnímu případu w vystupující v roli referenční rozlišovací úrovně budeme označovat jako l^{R}. Pro referenční rozlišovací úroveň $l^{\mathrm{R}} \in \mathcal{L}_{\mathbb{T}}$ platí, že je na této rozlišovací úrovni známá struktura s^{R}, a to již před zahájením příslušné úlohy integrace nebo dezintegrace prvků systému topologického popisu sítě $\mathbb{T}$. Struktura s^{R} se stává jedním ze vstupů pro tuto úlohu.

Takové zavedené rozlišovací úrovně, na nichž před zahájením úlohy integrace nebo dezintegrace prvků systému topologického popisu sítě $\mathbb{T}$ není známa struktura s^{D} a pro něž má být tato struktura získána jako výstup příslušné integrační nebo dezintegrační úlohy řešené v rámci konkrétního aplikačního případu w, budou ve vztahu k tomuto aplikačnímu případu vystupovat v roli odvozených rozlišovacích úrovní. Pro rozlišovací

úroveň vystupující obecně v roli odvozené rozlišovací úrovně budeme používat označení l^{D}. Pro každou odvozenou rozlišovací úroveň l^{D} opět platí $l^{\mathrm{D}} \in \mathcal{L}_{\mathbb{T}}$.

Takové odvozené rozlišovací úrovně, na nichž má být vyjádřena struktura s^{G} jako výsledek úlohy integrace prvků systému topologického popisu sítě, budou ve vztahu k příslušnému aplikačnímu případu w (resp. k příslušné referenční rozlišovací úrovni l^{R}) vystupovat v roli méně podrobných odvozených rozlišovacích úrovní. Pro rozlišovací úroveň vystupující v roli méně podrobné odvozené rozlišovací úrovně budeme používat označení l^{G}. Pro každou méně podrobnou rozlišovací úroveň l^{G} opět platí $l^{\mathrm{G}} \in \mathcal{L}_{\mathbb{T}}$.

Takové odvozené rozlišovací úrovně, na nichž má být vyjádřena struktura s^{F} jako výsledek úlohy dezintegrace prvků systému topologického popisu sítě, budou ve vztahu k příslušnému aplikačnímu případu w (resp. k příslušné referenční rozlišovací úrovni l^{R}) vystupovat v roli podrobnějších odvozených rozlišovacích úrovní. Pro rozlišovací úroveň vystupující v roli podrobnější odvozené rozlišovací úrovně budeme používat označení l^{F}. Pro každou podrobnější rozlišovací úroveň l^{F} opět platí $l^{\mathrm{F}} \in \mathcal{L}_{\mathbb{T}}$.

3.7 Prvky

Prvek je oddělitelnou částí systému, která nemá na dané rozlišovací úrovni zavedenou vnitřní strukturu. V rámci této práce se budeme primárně zabývat prvky systému topologického popisu sítě $\mathbb{T}$, čemuž také podřídíme způsob jejich označování. Za prvky systému topologického popisu sítě $\mathbb{T}$ budeme považovat prvky topologického popisu železniční (resp. drážní) infrastruktury, které reprezentují objekty reálného světa, nacházející se v té oblasti železniční (resp. drážní) infrastruktury reprezentovanou sítí n.

Množinu prvků systému topologického popisu systému sítě $\mathbb{T}$ budeme obecně označovat jako $\mathcal{A}$. Množinu $\mathcal{A}$ lze rozepsat jako

$$\mathcal{A} = \{a_1, a_2, \dots a_k\},$$

kde každý prvek $a \in \mathcal{A}$, tedy a_i, $i \in \{1, 2, \dots k\}$, je prvek systému $\mathbb{T}$ a k je celkový počet prvků systému $\mathbb{T}$ množiny $\mathcal{A}$.

To, jak rozsáhlou část oblasti, na níž je identifikován systém topologického popisu sítě $\mathbb{T}$ (resp. jeho nadsystém $\mathbb{R}$) prvek reprezentuje, závisí na míře podrobnosti zavedené rozlišovací úrovně l^{E}, na níž tento prvek figuruje. Vzhledem k tomu, že pro každý aplikační případ w uvažujeme s několika různými rozlišovacími úrovněmi zavedenými pro systém $\mathbb{T}$, je nutné tuto skutečnost reflektovat také při označování prvků a jejich množin. Množinu prvků systému topologického popisu systému sítě $\mathbb{T}$ figurujících na rozlišovací úrovni l^{E} budeme označovat jako $\mathcal{A}^{\mathrm{E}}$. Množinu $\mathcal{A}^{\mathrm{E}}$ lze rozepsat jako

$$\mathcal{A}^{\mathrm{E}} = \{a_1^{\mathrm{E}}, a_2^{\mathrm{E}}, \dots a_k^{\mathrm{E}}\},$$

kde každý prvek $a^{\mathrm{E}} \in \mathcal{A}^{\mathrm{E}}$, tedy a_i^{E}, $i \in \{1, 2, \dots k\}$, je prvek systému $\mathbb{T}$ figurující na rozlišovací úrovni l^{E} a k je celkový počet prvků systému $\mathbb{T}$ figurujících na rozlišovací úrovni l^{E}.

3.7.1 Třídy prvků

Za účelem kategorizace prvků např. podle toho, jakým způsobem je jejich prostřednictvím vyjádřen objekt reálného světa, jejž reprezentují, jakým způsobem je možné začlenit je do struktury systému, jehož jsou součástí, podle veličin, jimž lze tyto prvky popsat, nebo funkcí, které u nich lze sledovat mohou být pro prvky, jakožto objekty

sloužící k popisu železniční (resp. drážní) infrastruktury, zavedeny specializované třídy. Tyto třídy je možné v závislosti na konkrétním aplikačním příkladu w zavést v podstatě libovolně. Toho je možné s výhodou využít např. pro účely kategorizace prvků netopologického okolí systému $\mathbb{T}$. Pro prvky topologického popisu železniční (resp. drážní) infrastruktury, které jsou prvky systému topologického popisu sítě $\mathbb{T}$ a jeho topologického okolí, budeme nicméně při dalších úvahách využívat kategorizací, která je zavedena na vysoce obecné úrovni v souladu s přístupem metodiky RTM, již v této oblasti implementuje také VMŽI. Jedná se o kategorizaci na prvky neliniové a na prvky liniové.

Pro prvky topologického popisu železniční (resp. drážní) infrastruktury zavedeme v souladu s touto skutečností následující hodnoty veličiny ϵ vyjadřující příslušnost objektu k třídě:

- `N` – vyjadřuje skutečnost, že příslušný prvek topologického popisu železniční (resp. drážní) infrastruktury patří do třídy prvků neliniových
- `L` – vyjadřuje skutečnost, že příslušný prvek topologického popisu železniční (resp. drážní) infrastruktury patří do třídy prvků liniových

Kategorizace prvků topologického popisu železniční (resp. drážní) infrastruktury je obecně nezávislá na uvažované rozlišovací úrovni, resp. se předpokládá, že bude řešena pro jednotlivé zavedené rozlišovací úrovně systému $\mathbb{T}$ individuálně. Neliniový prvek systému $\mathbb{T}$ budeme obecně označovat jako a^{N}, zatímco pro liniový prvek systému $\mathbb{T}$ použijeme obecné označení a^{L}. V případě potřeby vyjádřit skutečnost, že množina prvků $\mathcal{A}$ obsahuje pouze neliniové prvky, resp. pouze prvky liniové, využijeme pro ni označení $\mathcal{A}^{\mathrm{N}}$, resp. $\mathcal{A}^{\mathrm{L}}$. Obdobně můžeme tuto skutečnost vyjádřit pro jakoukoliv množinu $\mathcal{A}_{\mathrm{S}}$, pro níž platí $\mathcal{A}_{\mathrm{S}} \subset \mathcal{A}$, a to s využitím označení $\mathcal{A}_{\mathrm{S}}^{\mathrm{L}}$, resp. $\mathcal{A}_{\mathrm{S}}^{\mathrm{N}}$. Abychom vyjádřili příslušnost prvků k jednotlivým z právě zavedených množin, budeme pro ně používat označení a_{S}, $a_{\mathrm{S}}^{\mathrm{L}}$, resp. $a_{\mathrm{S}}^{\mathrm{N}}$.

Konkrétní prvky topologického popisu železniční (resp. drážní) infrastruktury kategorizované na neliniové a neliniové je nicméně nutné nadále nutné vnímat v kontextu rozlišovací úrovně l^{E}, na níž figurují. Neliniový prvek systému $\mathbb{T}$ figurující na rozlišovací úrovni l^{E} budeme označovat jako a^{EN}, zatímco pro liniový prvek systému $\mathbb{T}$ figurující na rozlišovací úrovni l^{E} použijeme označení a^{EL}. V případě potřeby vyjádřit skutečnost, že množina $\mathcal{A}^{\mathrm{E}}$ obsahuje pouze neliniové prvky, resp. pouze prvky liniové, využijeme pro ni označení $\mathcal{A}^{\mathrm{EN}}$, resp. $\mathcal{A}^{\mathrm{EL}}$. Obdobně můžeme tuto skutečnost vyjádřit pro jakoukoliv množinu $\mathcal{A}_{\mathrm{S}}^{\mathrm{E}}$, pro níž platí $\mathcal{A}_{\mathrm{S}}^{\mathrm{E}} \subset \mathcal{A}^{\mathrm{E}}$, a to s využitím označení $\mathcal{A}_{\mathrm{S}}^{\mathrm{EL}}$, resp. $\mathcal{A}_{\mathrm{S}}^{\mathrm{EN}}$. Abychom vyjádřili příslušnost prvků k jednotlivým z právě zavedených množin, budeme pro ně používat označení $a_{\mathrm{S}}^{\mathrm{E}}$, $a_{\mathrm{S}}^{\mathrm{EL}}$, resp. $a_{\mathrm{S}}^{\mathrm{EN}}$.

Neliniové prvky je možné uplatnit zejména při reprezentaci objektů reálného světa bodového nebo plošného charakteru. Liniové prvky je možné uplatnit při reprezentaci objektů reálného světa liniového charakteru. Prostřednictvím liniových prvků je možné výstižněji (podrobněji) vyjádřit některé funkční vlastnosti těchto objektů. Pro různé aplikační případy, které se navzájem liší např. způsobem, jakým je vyjádřena síť n a mírou podrobnosti uvažované zavedené rozlišovací úrovně l^{E}, může být ten samý objekt reálného světa reprezentován jednou jako prvek neliniový, podruhé jako prvek liniový. Připouští se zároveň, že objekt reálného světa reprezentovaný v rámci daného aplikačního případu w na podrobnější rozlišovací úrovni l^{F} liniovým prvkem a^{FL}, může být v rámci toho samého aplikačního případu w na méně podrobné rozlišovací úrovni l^{G} reprezentován neliniovým prvkem a^{GN}.

Příkladem typu objektu reálného světa, na němž je možné výše uvedené tvrzení demonstrovat, je výhybka. Pokud nám postačuje na výhybku nahlížet s vyšší mírou

abstrakce a uvažovat s ní jako s bodem (resp. oblastí), kde dochází k větvení kolejových tras (např. v případě, že na úrovni struktury s^{E} nepotřebujeme podrobněji postihnout její funkci) postačuje nám reprezentovat ji jako neliniový prvek. Pokud na výhybku nahlížíme ze stavebního hlediska jako na technické zařízení, u nějž chceme zdůraznit jeho převládající rozměr, můžeme ji reprezentovat jako liniový prvek. Tento přístup nám také na úrovní struktury s^{E} umožní snáze postihnout některé funkční vlastnosti, které by v případě reprezentace výhybky jako prvku neliniového bylo nutné vyjádřit alternativním způsobem.

3.7.2 Funkční typy prvků

Zatímco třídy prvků topologického popisu železniční (resp. drážní) infrastruktury, zejména pokud jejich prostřednictvím rozlišujeme pouze mezi prvky neliniovými a liniovými, vnímáme na vysoce obecné a abstraktní úrovni, pro některé aplikační případy může být zejména v kontextu jednotlivých zavedených rozlišovacích úrovní vhodné tyto prvky dále kategorizovat na základě funkčních typů. Obdobnou kategorizaci lze použít i pro prvky netopologického okolí systému $\mathbb{T}$, byť s nutností definovat jiné kategorie, vyhovující účelu vyplývajícímu z konkrétního aplikačního případu w.

Funkční typ prvku můžeme do jisté míry vnímat jako typ objektu reálného světa, který příslušný prvek reprezentuje, neboť objekty reálného světa stejného typu vykonávají podobnou funkci. Vzhledem ke skutečnosti, že systém topologického popisu sítě $\mathbb{T}$ je subsystémem systému popisu sítě $\mathbb{R}$ získaným jako jeho dekomponovaná část na základě jeho funkční dekompozice, lze očekávat, že funkce jeho prvků, zvláště pak těch, které společně figurují na jedné zavedené rozlišovací úrovni l^{E}, budou spíše homogenního charakteru. Rámcové funkce systému topologického popisu sítě $\mathbb{T}$ jsou skutečné stejné. Funkci těchto prvků je poskytovat společnou vztažnou bázi pro ukotvení souřadnicových systémů nutných vyjádření lokalizace železničních vozidel, a to včetně definování podmínek jejich přípustného pohybu (nesení a vedení), a zařízení a vlastností železniční (resp. drážní) infrastruktury reprezentovaných prvky netopologického okolí systému $\mathbb{T}$ v rámci oblasti reprezentované sítí n, případně jejího okolí. Způsob realizace těchto funkcí se nicméně může pro jednotlivé prvky systému topologického popisu železniční (resp. drážní) infrastruktury lišit, a to zejména na základě třídy těchto prvků, ale v mnohých případech také v závislosti na podrobnějším určení objektu reálného světa, který příslušný prvek reprezentuje.

Seznam typů objektů reálného světa, které mohou být reprezentovány prvky systému topologického popisu sítě $\mathbb{T}$ a jeho topologického okolí, ani prvky netopologického okolí systému $\mathbb{T}$, pro něž lze rovněž různé funkční typy definovat, není pevně stanoven. Tyto typy je nutné definovat vždy v závislosti na konkrétním aplikačním případu w.

Dále budeme uvažovat o takových typech objektů reálného světa, které mohou být reprezentovány prvky topologického popisu železniční (resp. drážní) infrastruktury a o jejich reprezentaci těmito prvky. S vědomím jisté míry abstrakce a rizika výskytu určitých sémantických nesrovnalostí přeneseme označení typů objektů reálného světa na funkční typy prvků, které objekty reálného světa těchto typů reprezentují. Budeme tak hovořit např. o prvcích funkčních typu dopravna, traťový úsek, větev výhybky, kolej ve stavebním smyslu apod. Na rozdíl od typu objektu reálného světa, který je v rámci systému $\mathbb{T}$ reprezentován prvkem a^{E} reprezentuje, je funkční typ prvku a^{E} nutné vnímat více v kontextu konkrétního aplikačního případu w a navíc také v kontextu konkrétní zavedené rozlišovací úrovně l^{E}, na níž je v rámci aplikačního případu w vyjádřena struktura s^{E}, jíž je prvek a^{E} součástí.

funkční typ prvku	N	L
síť	•	
oblast v rámci sítě (např. tarifní zóna)	•	
trať	(•)	•
dopravna	•	•
traťový (mezistaniční) úsek	(•)	•
přepravní stanoviště	•	•
zhlaví	•	•
kolejová skupina	•	•
napájecí stopa	•	(•)
oblast kontroly volnosti koleje	•	(•)
kolejová trasa	(•)	•
výhybka (výhybková konstrukce) v dopravním smyslu	•	•
kolej v dopravním smyslu	(•)	•
výhybka (výhybková konstrukce) ve stavebním smyslu	•	•
kolej ve stavebním smyslu	•	(•)
větev výhybky	(•)	•
rameno výhybky	(•)	•
úsek koleje konstantních (např. geometrických) parametrů	(•)	•
kolejnice	(•)	•
srdcovka	•	•

Tabulka 3.1. Příklady možných funkčních typů prvků systému.

Seznam příkladů funkčních typů prvků, které mohou v rámci některých aplikačních případů figurovat na některých zavedených rozlišovacích úrovních, je uveden v tabulce 3.1. Tabulka byla sestavena tak, aby funkční typy prvků uvedené výše nalézaly uplatnění spíše ve strukturách vyjádřených na méně podrobných rozlišovacích úrovních, zatímco typy prvků uvedené níže nalézaly uplatnění spíše ve strukturách vyjádřených na podrobnějších rozlišovacích úrovních. Toto pravidlo však chápejme spíše orientačně. V některých případech mohou být rozlišovací úrovně, na nichž figurují prvky některých z uvedených funkčních typů, z hlediska míry podrobnosti dokonce neporovnatelné, a to např. z důvodu výrazně rozdílných způsobů, jakými je síť n v rámci jednotlivých aplikačních případů na jednotlivých zavedených rozlišovacích úrovních, např. při specifikaci přípustného typu struktury, vyjadřována.

Předpokládejme, že objekty reálného světa stejného typu budou v rámci daného aplikačního případu w na dané rozlišovací úrovni l^{E} reprezentovány prvky stejné třídy a stejného funkčního typu. To mimo jiné usnadní standardizaci jejich popisu s využitím veličin vyjadřujících parametry těchto objektů reálného světa přenášených na tyto prvky. Za těchto okolností (nikoliv však absolutně) je možné prohlásit, že je třída prvku závislá na jeho funkčním typu. Vhodnost reprezentace objektů reálného světa v závislosti na jejich typu prostřednictvím neliniových a liniových prvků může být do značné

míry závislá také na způsobu, jakým je v rámci příslušného aplikačního případu w vyjadřována síť n a na míře podrobnosti uvažované zavedené rozlišovací úrovně l^{E}.

Z tabulky 3.1 je možné vysledovat, že objekty reálného světa reprezentované prvky některých z uvedených funkčních typů mají tendenci se jevit spíše jako objekty bodového (resp. plošného) charakteru. U jiných je naopak zřejmý převážně liniový charakter. Ačkoliv tato vlastnost může být jedním z kritérií, na jejichž základě lze rozhodnout, prvky které třídy mají být objekty reálného světa daného funkčního typu reprezentovány, v některých případech nemusí být příslušnost prvku a^{E} k třídě neliniových nebo liniových prvků pouze na základě jeho funkčního typu zřejmá. Tato souvislost může vyplynout např. až v závislosti na podrobnější určení způsobu, jakým má být v rámci konkrétního aplikačního případu w síť n vyjadřována, a míře podrobnosti rozlišovací úrovně l^{E}, na níž má prvek a^{E} figurovat, tedy při specifikaci typu přípustné struktury s^{E}. Při ní může být cílem některé aspekty objektů reálného světa modifikovat, potlačit nebo naopak vyzdvihnout, což závisí mimo jiné na konkrétním účelu, za nímž je systém $\mathbb{T}$ v rámci aplikačního případu w navrhován.

V tabulce 3.1 je dále (s ohledem na výše zmíněné skutečnosti opět pouze orientačně) specifikováno, do jaké míry může být pro daný funkční typ prvku uvažováno se skutečností, že v rámci některé rozlišovací úrovně l^{E} některého aplikačního případu w budou prvky toto funkčního typu vystupovat jako prvky neliniové, resp. liniové. Ve sloupci `N` (vhodnost reprezentace prostřednictvím neliniového prvku), resp. `L` (vhodnost reprezentace prostřednictvím liniového prvku), je tato skutečnost vyjádřena buď symbolem • (objekt reálného světa typu, jemuž odpovídá daný funkční typ prvku, je vhodné reprezentovat prvkem příslušné třídy), symbolem (•) (objekt reálného světa typu, jemuž odpovídá daný funkční typ prvku, je vhodné reprezentovat prvkem příslušné třídy pouze ve specifických případech), nebo prázdnou buňkou (objekt reálného světa typu, jemuž odpovídá daný funkční typ prvku, není vhodné reprezentovat prvkem příslušné třídy). Ačkoliv v mnohých případech připadá v úvahu reprezentace objektu reálného světa určitého typu s využitím neliniových i liniových prvků, nepředpokládá se, že by objekty reálného světa stejného typu byly na jedné rozlišovací úrovni l^{E} reprezentovány prvky různých tříd.

Funkční typ prvku není vždy nutné výslovně vyjádřit, zvlášť pokud jedna třída prvků odpovídá pouze jednomu funkčnímu typu prvku. Na typ prvku a^{E} je možné usuzovat také na základě způsobu možného začlenění tohoto prvku do struktury s^{E}, který jednak reflektuje příslušnost prvku a^{E} k třídě, ale může zároveň přinášet ještě podrobnější informaci o funkci prvku. Funkční typ prvku a^{E} tak můžeme odvodit např. na základě jeho třídy a počtu dalších prvků topologického popisu železniční infrastruktury (resp. drážní), které s ním incidují, na základě charakteristik incidujících prvků netopologického okolí systému $\mathbb{T}$, nebo na základě hodnot jiných parametrů, jimiž je prvek a^{E} popsán. V některých případech je však vhodné vyjádřit funkční typ prvků přímo v podobě hodnoty veličiny obdobným způsobem, jakým vyjadřujeme např. jeho třídu nebo různé popisné parametry.

V rámci konkrétního aplikačního případu w můžeme pro prvky zavést veličinu τ vyjadřující funkční typ prvku. Vzhledem ke skutečnosti, že funkční typ prvku je nutné vnímat v kontextu rozlišovací úrovně w a příslušnosti k třídě ϵ, přizpůsobíme tomu i podrobnější označování této veličiny. Funkční typ neliniového prvku figurujícího na rozlišovací úrovni w tak budeme označovat jako $\tau^{\mathrm{E}}_{\mathrm{N}}$ a funkční typ liniového prvku figurujícího na této rozlišovací úrovni budeme označovat jako $\tau^{\mathrm{E}}_{\mathrm{L}}$. Hodnoty, kterých mohou tyto veličiny nabývat, budou stanovovány individuálně pro každý aplikační případ w a rozlišovací úroveň l^{E}.

3.7.3 Popis prvků

Na základě příslušnosti prvku k rozlišovací úrovni l^{E}, na níž teto prvek figuruje (případně k jiným způsobem definované množině objektů), a k třídě ϵ, jíž je instancí, lze v rámci daného aplikačního případu w účelově zavést i další veličiny, které prvek popisují. V případě prvků systému topologického popisu sítě $\mathbb{T}$ uvažujeme s třídou prvků liniových a s třídou prvků neliniových. Neliniový prvek $a^{\mathrm{EN}} \in \mathcal{A}^{\mathrm{EN}}$ tak budeme popisovat veličinami množiny $\mathcal{M}(\mathcal{A}^{\mathrm{EN}})$ a liniový prvek $a^{\mathrm{EL}} \in \mathcal{A}^{\mathrm{EL}}$ veličinami množiny $\mathcal{M}(\mathcal{A}^{\mathrm{EL}})$. V případě, že neliniové i liniové prvky systému $\mathbb{T}$ figurující na rozlišovací úrovni l^{E} mohou být popsány stejnými veličinami, příp. chceme pracovat s množinou takových veličin, které jsou pro všechny prvky množiny $\mathcal{A}^{\mathrm{E}}$ společné, lze tyto veličiny vyjádřit pomocí množiny $\mathcal{M}(\mathcal{A}^{\mathrm{E}})$.

Veličinu $m^{\mathrm{E}} \in \mathcal{M}(\mathcal{A}^{\mathrm{E}})$ lze chápat zcela obecně. Může se jednat o proměnnou stavovou veličinu, jejíž hodnota se mění v závislosti na vstupech prvku a^{E}, ale také o popisný parametr prvku a^{E}, jehož hodnota zůstává v průběhu realizace procesů nad strukturou s^{E} neměnná. Stejně jako všechny v případě všech ostatních objektů zařadíme do množiny $\mathcal{M}(\mathcal{A}^{\mathrm{E}})$ identifikátor ι a, jak již bylo řečeno, k vyjádření příslušnosti prvku a^{E} ke třídě budeme využívat veličinu ϵ. Jako jednu z poměrně obvyklých veličin pro popis prvků systému $\mathbb{T}$ figurujících na rozlišovací úrovni e můžeme zavést také již zmíněný funkční typ prvku v modifikacích $\tau_{\mathrm{N}}^{\mathrm{E}}$ a $\tau_{\mathrm{L}}^{\mathrm{E}}$ definujících přípustné hodnoty této veličiny pro neliniové a liniové prvky.

Další veličiny je možné zavádět účelově, s ohledem na potřeby daného aplikačního případe w. Jednou z poměrně typických veličin, která může být uplatněna zejména pro popis liniových prvků, je délka liniového objektu reálného světa, který příslušný prvek reprezentuje, měřená v ose. Tuto veličinu budeme označovat jako λ a obvykle vyjadřovat v metrech. Opět můžeme definovat různé modifikace této veličiny v závislosti na podrobnějších okolnostech jejího užití.

Jelikož prvky netopologického okolí systému $\mathbb{T}$ mohou být podstatně heterogennějšího charakteru než prvky topologického popisu železniční infrastruktury, je možné při jejich implementaci očekávat zavedení většího počtu tříd a s tím související rozmanitější popis s využitím širší škály veličin a jejich přípustných hodnot.

3.7.4 Prvky okolí systému topologického popisu sítě

Prvky okolí systému topologického popisu sítě $\mathbb{T}$ je možné rozdělit na prvky topologického okolí systému $\mathbb{T}$ a prvky netopologického okolí systému $\mathbb{T}$.

Za prvky topologického okolí systému $\mathbb{T}$ budeme považovat prvky topologického popisu železniční (resp. drážní) infrastruktury, které reprezentují objekty reálného světa nacházející se mimo oblast železniční (resp. drážní) infrastruktury reprezentovanou sítí n, ale pro něž existují s objekty reálného světa reprezentovanými prvky systému topologického popisu sítě $\mathbb{T}$ přímé funkční nebo fyzické relace. Množinu prvků topologického okolí systému topologického popisu sítě $\mathbb{T}$ budeme obecně označovat jako $\mathcal{U}$.

Množinu $\mathcal{U}$ lze rozepsat jako

$$\mathcal{U} = \{u_1, u_2, \dots u_k\},$$

kde každý prvek $u \in \mathcal{U}$, tedy u_i, $i \in \{1, 2, \dots k\}$, je prvek topologického okolí systému topologického popisu sítě $\mathbb{T}$, a k je celkový počet prvků topologického okolí systému topologického popisu sítě $\mathbb{T}$ množiny $\mathcal{U}$.

Obecná pravidla, včetně způsobu podrobnějšího označování, např. na základě příslušnosti k rozlišovací úrovni nebo k třídě prvků, která jsou zavedena pro prvky systému $\mathbb{T}$,

jsou aplikovatelná také na prvky jeho topologického okolí, neboť se jedná o prvky stejného charakteru. Množinu prvků topologického okolí systému $\mathbb{T}$, které figurují na rozlišovací úrovni l^{E}, budeme označovat jako $\mathcal{U}^{\mathrm{E}}$, a prvek topologického okolí systému $\mathbb{T}$, který je prvkem této množiny budeme obecně označovat jako u^{E}. Obdobným způsobem můžeme na prvky topologického okolí systému $\mathbb{T}$ aplikovat také označování jejich množin a prvků samotných na základě jejich příslušnosti ke třídě prvků neliniových nebo liniových, které jsme zavedli pro prvky systému topologického popisu sítě $\mathbb{T}$.

Za prvky netopologického okolí systému $\mathbb{T}$ budeme považovat prvky systému sítě $\mathbb{R}$, které nejsou součástí systému $\mathbb{T}$, ale které reprezentují objekty reálného světa, pro něž existují s objekty reálného světa reprezentovanými prvky systému topologického popisu sítě $\mathbb{T}$ přímé funkční nebo fyzické relace. Nejedná se tedy o prvky topologického popisu železniční (resp. drážní) infrastruktury. Množinu prvků netopologického okolí systému topologického popisu sítě $\mathbb{T}$ budeme obecně označovat jako $\mathcal{E}$. Množinu $\mathcal{E}$ lze rozepsat jako

$$\mathcal{E} = \{e_1, e_2, \dots e_k\},$$

kde každý prvek $e \in \mathcal{E}$, tedy e_i, $i \in \{1, 2, \dots k\}$, je prvek netopologického okolí systému topologického popisu sítě $\mathbb{T}$, a k je celkový počet prvků netopologického okolí systému topologického popisu sítě $\mathbb{T}$ množiny $\mathcal{E}$.

Obecná pravidla, která jsou zavedena pro prvky systému $\mathbb{T}$, nejsou sice na prvky jeho netopologického okolí ve všech případech aplikovatelná, neboť se jedná o prvky rozdílného charakteru, v případě, že je to účelné, však na ně můžeme aplikovat např. pravidla týkající se způsobu podrobnějšího označování na základě jejich příslušnosti k rozlišovací úrovni, které budeme využívat u prvků systému $\mathbb{T}$. Množinu prvků netopologického okolí systému $\mathbb{T}$, které figurují na rozlišovací úrovni l^{E}, budeme označovat jako $\mathcal{E}^{\mathrm{E}}$, a prvek netopologického okolí systému $\mathbb{T}$, který je prvkem této množiny budeme obecně označovat jako e^{E}. Označování množin prvků a prvků samotných na základě jejich příslušnosti ke třídě prvků neliniových nebo liniových, které jsme zavedli pro prvky systému topologického popisu sítě $\mathbb{T}$ na prvky netopologického okolí systému $\mathbb{T}$ aplikovat nemůžeme, neboť tyto třídy nejsou pro tyto prvky zavedeny, v závislosti na konkrétním aplikačním případu w můžeme ale libovolné třídy prvků netopologického okolí systému $\mathbb{T}$ zavést a následně na ně označování na základě těchto tříd aplikovat.

3.7.5 Konektory

Za účelem možnosti propojování oddělených struktur zavedeme speciální třídu prvků, které budeme nazývat konektory. Konektor budeme vnímat jako abstraktní prvek, který nemá předobraz v žádném objektu reálného světa. S konektorem může být v rámci dané o konektory a konektorové vazby rozšířené struktury, propojen konektorovou vazbou libovolný prvek této struktury. Ztotožněním dvou konektorů různých struktur (což může být realizováno za předpokladu splnění podmínky propojitelnosti) dojde k nahrazení konektorových vazeb propojujících každý z těchto konektorů s prvkem příslušné struktury jedinou vazbou propojující tyto prvky, a tím i k propojení těchto struktur.

V rámci této práce budeme předpokládat, že konektory mohou být vazbou propojeny pouze s prvky topologického popisu železniční (resp. drážní) infrastruktury. Na úrovni systému topologického popisu sítě $\mathbb{T}$ je možné uvažovat konektory na rozhraní struktury tohoto systému a (potenciálně neznámé) struktury topologického okolí systému $\mathbb{T}$. Konektory nebudeme považovat za součást struktury, k jejímuž propojení s jinými strukturami mohou být použity, ale za součást rozšířené struktury, kterou pro tento účel zavedeme. Množinu konektorů rozšířené struktury, která vznikla rozšířením struktury

s, jejímiž prvky jsou prvky systému topologického popisu sítě $\mathbb{T}$ množiny $\mathcal{A}$, budeme obecně označovat jako $\mathcal{B}$. Množinu $\mathcal{B}$ lze rozepsat jako

$$\mathcal{B} = \{b_1, b_2, \dots b_k\},$$

kde každý konektor $b \in \mathcal{B}$, tedy b_i, $i \in \{1, 2, \dots k\}$, je konektorem rozšířené struktury $\tilde{s}$, sloužícím k propojením některého prvku množiny $\mathcal{A}$ s jiným prvkem (zpravidla mimo strukturu s) a k je celkový počet konektorů rozšířené struktury $\tilde{s}$.

Obdobným způsobem můžeme vyjádřit množinu konektorů rozšířené struktury $\tilde{s}^{\mathrm{E}}$ systému $\mathbb{T}$, vyjádřené na zavedené rozlišovací úrovni l^{E}, kterou označíme jako $\mathcal{B}^{\mathrm{E}}$, resp. množinu konektorů libovolné rozšířené struktury $\tilde{s}_{\mathrm{S}}$, kterou označíme jako $\mathcal{B}_{\mathrm{S}}$.

Konektory je možné, stejně jako ostatní prvky, popsat hodnotami veličin. Na jejich základě budeme s využitím definovaných podmínek propojitelnosti posuzovat, zda jsou dva konektory b_{A} a b_{B} různých rozšířených struktur vzájemně ztotožnitelné. V případě, že se podaří dva takovéto konektory vzájemně ztotožnit, je možné vytvořit uspořádanou dvojici těchto konektorů

$$p = (b_{\mathrm{A}}, b_{\mathrm{B}}),$$

kterou nazveme přiřazením konektorů, k propojování oddělených struktur. Množinu přiřazení konektorů budeme obecně označovat jako $\mathcal{P}$. Množinu $\mathcal{P}$ lze rozepsat jako

$$\mathcal{P} = \{p_1, p_2, \dots p_k\},$$

kde každé přiřazení konektorů $p \in \mathcal{P}$, tedy p_i, $i \in \{1, 2, \dots k\}$, je přiřazením konektorů množiny $\mathcal{P}$ a k je celkový počet přiřazení konektorů množiny $\mathcal{P}$.

Konektory mohou být uspořádávány do skupin. Konektorovou skupinou budeme mít na mysli libovolnou neprázdnou podmnožinu dané množiny konektorů $\mathcal{B}$. Konfigurací konektorových skupin ve vztahu k množině $\mathcal{B}$ potom nazveme každou takovou množinu konektorových skupin, jejichž sjednocením získáme množinu $\mathcal{B}$. Množinu konfigurací konektorových skupin budeme obecně označovat jako $\mathcal{D}$. Množinu $\mathcal{D}$ lze rozepsat jako

$$\mathcal{D} = \{d_1, d_2, \dots d_k\},$$

kde každá konfigurace konektorových skupin $d \in \mathcal{D}$, tedy d_i, $i \in \{1, 2, \dots k\}$, je konfigurace konektorových skupin množiny $\mathcal{D}$ a k je celkový počet konfigurací konektorových skupin množiny $\mathcal{D}$. Množinu konfigurací konektorových skupin budeme využívat v roli množiny přípustných konfigurací konektorových skupin.

3.7.6 Póly prvků

Za účelem možnosti podrobnějšího vyjádření způsobu začlenění prvku v závislosti na jeho třídě do struktury, v níž vystupuje, zavedeme pojem pól prvku. Pólem prvku budeme rozumět místo možného zaústění vazeb do prvku. Pro prvky topologického popisu železniční (resp. drážní) infrastruktury jsme zavedli třídu prvků neliniových a třídu prvků liniových. V případě prvků neliniových nejsme schopni místo zaústění vazeb do prvku blíže rozlišit, uvažujeme, že do neliniového prvku jsou všechny vazby zaústěny ve stejném místě. Pro každý neliniový prvek bude definován právě jeden pól. Pro neliniový prvek a^{N} zapíšeme tento pól jako

$$q_0(a^{\mathrm{N}}).$$

Oproti tomu prvky liniové se z hlediska způsobu jejich začlenění do struktury chovají jako dvojprvky. Vazba může být do liniového prvku zaústěna buď na jeho začátku nebo na jeho konci. Pro každý liniový prvek budou definovány právě dva póly, z nichž jeden bude nazýván pólem počátečním a druhý pólem koncovým. Počáteční pól liniového prvku a^{L} zapíšeme jako

$$q_0(a^{\mathrm{L}}).$$

Koncový pól liniového prvku a^{L} zapíšeme jako

$$q_1(a^{\mathrm{L}}).$$

Množinu pólů prvků množiny $\mathcal{A}$ budeme obecně označovat jako $\mathcal{Q}(\mathcal{A})$. Množinu $\mathcal{Q}(\mathcal{A})$ lze rozepsat jako

$$\mathcal{Q}(\mathcal{A}) = \left\{q_0(a_1^{\mathrm{N}}), q_0(a_2^{\mathrm{N}}), \ldots q_0(a_k^{\mathrm{N}})\right\} \cup$$

$$\cup \left\{q_0(a_1^{\mathrm{L}}), q_1(a_1^{\mathrm{L}}), q_0(a_2^{\mathrm{L}}), q_1(a_1^{\mathrm{L}}), \ldots q_0(a_h^{\mathrm{L}}), q_1(a_h^{\mathrm{L}})\right\},$$

kde každý $q_0(a_i^{\mathrm{N}})$, $i \in \{1, 2, \ldots k\}$ je pól neliniového prvku množiny $\mathcal{A}^{\mathrm{N}} \subset \mathcal{A}$, každý $q_0(a_i^{\mathrm{L}})$, $i \in \{1, 2, \ldots h\}$ je počáteční pól liniového prvku množiny $\mathcal{A}^{\mathrm{L}} \subset \mathcal{A}$, každý $q_1(a_i^{\mathrm{L}})$, $i \in \{1, 2, \ldots h\}$ je koncový pól liniového prvku množiny $\mathcal{A}^{\mathrm{L}} \subset \mathcal{A}$, k je celkový počet neliniových prvků množiny $\mathcal{A}^{\mathrm{N}}$ a h je celkový počet liniových prvků množiny $\mathcal{A}^{\mathrm{L}}$.

Obdobným způsobem můžeme vyjádřit množinu pólů prvků struktury s^{E}, kterou označíme jako $\mathcal{Q}(\mathcal{A}^{\mathrm{E}})$, resp. množinu pólů prvků libovolné množiny $\mathcal{A}_{\mathrm{S}}$, $\mathcal{A}_{\mathrm{S}} \subset \mathcal{A}$, kterou označíme jako $\mathcal{Q}(\mathcal{A}_{\mathrm{S}})$. Množinu pólů je možné vyjádřit také pro jakoukoliv množinu ostatních prvků topologického popisu železniční (resp. drážní) infrastruktury a pro jakoukoliv množinu konektorů zavedených za účelem propojování těchto prvků.

3.8 Vazby

Mezi oddělitelnými částmi celku, na němž identifikujeme systém, existují relace. Relace vyjádříme jako vazby mezi prvky reprezentujícími na určité zavedené rozlišovací úrovni l^{E} tyto oddělitelné části.

Vazby je možné obecně chápat jako orientované. Orientovanou vazbu mezi prvky a_{x} a a_{y}, pro niž je prvek a_{x} zdrojovým prvkem a prvek a_{y} prvkem cílovým, zapíšeme jako uspořádanou dvojici těchto prvků:

$$\vec{r}_{\mathrm{xy}} = \left(a_{\mathrm{x}}, a_{\mathrm{y}}\right).$$

Obdobně vazbu mezi prvky a_{y} a a_{x}, pro niž je prvek a_{y} zdrojovým prvkem a prvek a_{x} prvkem cílovým zapíšeme jako uspořádanou dvojici těchto prvků:

$$\vec{r}_{\mathrm{yx}} = \left(a_{\mathrm{y}}, a_{\mathrm{x}}\right).$$

Vyjadřujeme-li s využitím vazeb fyzickou propojenost mezi dvěma částmi celku reprezentovanými prvky a_{x} a a_{y}, je možné vycházet z předpokladu, že vazbu $\left(a_{\mathrm{y}}, a_{\mathrm{x}}\right)$ zavedeme právě tehdy, když zavedeme vazbu $\left(a_{\mathrm{x}}, a_{\mathrm{y}}\right)$. Vazbu mezi těmito prvky je potom možné chápat jako neorientovanou a zapsat ji jako neuspořádanou dvojici těchto prvků, tedy jako jejich množinu:

$$\overline{r} = \overline{r}_{\mathrm{xy}} = \overline{r}_{\mathrm{yx}} = \left\{a_{\mathrm{x}}, a_{\mathrm{y}}\right\} = \left\{a_{\mathrm{y}}, a_{\mathrm{x}}\right\}.$$

Pro účely této práce budeme uvažovat, že je tento předpoklad splněn a vazby budeme chápat jako neorientované. Pokud bychom se rozhodli s využitím vazeb vyjadřovat pouze funkční propojenost mezi dvěma částmi celku, tento předpoklad by za určitých okolností nemusel být správný, a to zejména v případě, že bychom chtěli tyto úvahy aplikovat na síť jiného typu, než je síť železniční (resp. drážní) infrastruktury. Příkladem takové sítě může být síť vodních cest. V rámci této práce budeme s využitím vazeb funkční propojenost mezi dvěma částmi celku sice také vyjadřovat, ale pouze takovou, která existuje obousměrně, navíc pouze s využitím takových vazeb, které zároveň vyjadřují fyzickou propojenost těchto částí.

Na vzdory této skutečnosti bude v některých případech (např. při konstrukci struktury systému $\mathbb{T}$) nutné formálně rozlišovat role jednotlivých prvků, které příslušná vazba propojuje. V případě vazeb vzájemně propojujících prvky topologického popisu železniční (resp. drážní) infrastruktury (přičemž takovýto prvek může být též konektorem) bude dále nutné (zvlášť v případě prvků liniových) rozlišit, ke kterému pólu prvku je tato vazba připojena. Takovouto vazbu formálně vyjádříme jako

$$r_{\mathrm{x}_u y_v} = \left(q_u(a_\mathrm{x}), q_v(a_\mathrm{y})\right),$$

kde $q_u(a_\mathrm{x})$ vyjadřuje pól prvku a_x, k němuž je vazba $r_{\mathrm{x}_u\mathrm{y}_v}$ navázána a $q_v(a_\mathrm{y})$ vyjadřuje pól prvku a_y, k němuž je vazba $r_{\mathrm{x}_u y_v}$ navázána.

Množinu vazeb existujících mezi prvky množiny $\mathcal{A}$ systému topologického popisu systému sítě $\mathbb{T}$ budeme obecně označovat jako $\mathcal{R}$. Množinu $\mathcal{R}$ lze rozepsat jako

$$\mathcal{R} = \{r_1, r_2, \ldots r_k\},$$

kde každá vazba $r \in \mathcal{R}$, tedy r_i, $i \in \{1, 2, \ldots k\}$, je vazba mezi dvěma prvky množiny $\mathcal{A}$ a k je celkový počet takovýchto vazeb existujících v mezi prvky množiny $\mathcal{A}$.

Vzhledem ke skutečnosti tomu, že pro každý aplikační případ w uvažujeme s několika různými rozlišovacími úrovněmi zavedenými pro systém $\mathbb{T}$, je nutné tuto skutečnost reflektovat také při označování vazeb a jejich množin. Množinu vazeb existujících mezi prvky systému topologického popisu systému sítě $\mathbb{T}$ figurujícími na rozlišovací úrovni l^E budeme označovat jako $\mathcal{R}^\mathrm{E}$. Množinu $\mathcal{R}^\mathrm{E}$ lze rozepsat jako

$$\mathcal{R}^\mathrm{E} = \{r_1^\mathrm{E}, r_2^\mathrm{E}, \ldots r_k^\mathrm{E}\},$$

kde každá vazba $r^\mathrm{E} \in \mathcal{R}^\mathrm{E}$, tedy r_i^E, $i \in \{1, 2, \ldots k\}$, je vazbou mezi dvěma prvky systému $\mathbb{T}$ figurujícími na rozlišovací úrovni l^E a k je je celkový počet takovýchto vazeb existujících v rámci systému $\mathbb{T}$.

Při označování vazeb je vhodné zohlednit také případy, kdy vazba propojuje prvek systému $\mathbb{T}$ s prvkem topologického okolí systému $\mathbb{T}$, s prvkem netopologického okolí systému $\mathbb{T}$, nebo s konektorem rozšířené struktury systému $\mathbb{T}$. V rámci této práce budeme využívat vazeb propojujících prvky systému $\mathbb{T}$ s konektory. Tyto vazby budeme označovat jako konektorové vazby. Konektorovou vazbu propojující prvek množiny $\mathcal{A}$ s konektorem množiny $\mathcal{B}$ budeme obecně označovat jako $\hat{r}$. Množinu konektorových vazeb zapíšeme obecně jako $\hat{\mathcal{R}}$

3.8.1 Průchodnost vazeb

Prostřednictvím vazeb mezi prvky topologického popisu železniční (resp. drážní) infrastruktury budeme vyjadřovat prostorovou návaznost mezi částmi infrastruktury, které tyto prvky reprezentují. Tato prostorová návaznost může být vnímána na dvou základních úrovních:

- fyzická návaznost – existuje za předpokladu, že se příslušné objekty reálného světa stýkají v prostoru
- funkční návaznost – existuje za předpokladu existence fyzické návaznosti příslušných objektů reálného světa a současného splnění podmínek průchodnosti

Vazby mezi prvky topologického popisu železniční (resp. drážní) infrastruktury budeme konstruovat již v případě splnění podmínky fyzické návaznosti mezi těmito prvky reprezentovanými objekty reálného světa. Skutečnost, zda je zároveň splněna také podmínka funkční návaznosti, vyjádříme s využitím příznaku průchodnosti příslušné vazbě přiděleného. Budeme předpokládat, že příznak průchodnosti platí současně pro oba směry vazby. Průchodnost budeme chápat jako veličinu, kterou budeme popisovat všechny vazby mezi prvky topologického popisu železniční (resp. drážní) infrastruktury a také všechny konektorové vazby. Tuto veličinu označíme jako η. Zavedeme následující hodnoty veličiny η vyjadřující průchodnost vazby:

- `1` – vyjadřuje skutečnost, že je příslušná vazba mezi prvky topologického popisu železniční (resp. drážní) infrastruktury průchodná
- `0` – vyjadřuje skutečnost, že příslušná vazba mezi prvky topologického popisu železniční (resp. drážní) infrastruktury není průchodná

Dvě konektorové vazby navázané ke konektoru (který vznikl např. ztotožněním dvou konektorů různých substruktur) můžeme při odstranění konektoru nahradit vazbou mezi prvky (resp. jejich příslušnými póly) s tímto konektorem původně propojených těmito konektorovými vazbami. Nově vzniklou vazbu je možné považovat za průchodnou pouze v případě, že byly průchodné obě jí nahrazené konektorové vazby.

3.9 Struktury

S využitím množiny prvků a množiny vazeb můžeme vyjádřit strukturu. Strukturu systému topologického popisu sítě $\mathbb{T}$ budeme obecně označovat jako s. Vyjádříme ji jako uspořádanou dvojici množiny prvků $\mathcal{A}$ a množiny vazeb $\mathcal{R}$ systému $\mathbb{T}$. Tuto skutečnost zapíšeme jako

$$s = (\mathcal{A}, \mathcal{R})\,.$$

Obdobným způsobem můžeme vyjádřit také libovolnou substrukturu struktury s.

Množinu struktur, které jsou substrukturami struktury s budeme obecně označovat jako $\mathcal{S}$. Množinu $\mathcal{S}$ lze rozepsat jako

$$\mathcal{S} = \{s_1, s_2, \ldots s_k\}\,,$$

kde každá struktura $\dot{s} \in \mathcal{S}$, tedy s_i, $i \in \{1, 2, \ldots k\}$, je substrukturou struktury s a k je celkový počet struktur množiny $\mathcal{S}$.

Pro potřeby úloh řešených v rámci této práce budeme předpokládat, že v rámci každého aplikačního případu w má systém topologického popisu sítě $\mathbb{T}$ zavedenou strukturu na každé ze zavedených rozlišovacích úrovních $l^{\mathrm{E}} \in \mathcal{L}_{\mathbb{T}}$. O každé zavedené struktuře s^{E}, vyjádřené na zavedené rozlišovací úrovni l^{E}, předpokládáme, že jsme schopni specifikovat typ, který je pro ni přípustný. Ne každá zavedená struktura musí být od samého počátku známá. V některých případech ji získáme až jako výsledek úlohy dezintegrace nebo integrace prvků systému topologického popisu sítě $\mathbb{T}$. Strukturu systému topologického popisu sítě $\mathbb{T}$ budeme tedy chápat vždy v kontextu některé z těchto zavedených rozlišovacích úrovní.

Pro každou zavedenou rozlišovací úroveň l^{E} je možné odpovídající zavedenou strukturu s^{E} systému topologického popisu sítě $\mathbb{T}$ zapsat jako

$$s^{\mathrm{E}} = (\mathcal{A}^{\mathrm{E}}, \mathcal{R}^{\mathrm{E}}),$$

kde $\mathcal{A}^{\mathrm{E}}$ je množina prvků systému $\mathbb{T}$ figurujících na rozlišovací úrovni l^{E} a $\mathcal{R}^{\mathrm{E}}$ je množina vazeb systému $\mathbb{T}$ figurujících na rozlišovací úrovni l^{E}.

Obdobným způsobem můžeme vyjádřit také libovolnou substrukturu struktury s^{E}.

Množinu struktur, které jsou substrukturami struktury s^{E} budeme obecně označovat jako $\mathcal{S}^{\mathrm{E}}$. Množinu $\mathcal{S}^{\mathrm{E}}$ lze rozepsat jako

$$\mathcal{S}^{\mathrm{E}} = \{s_1^{\mathrm{E}}, s_2^{\mathrm{E}}, \dots s_k^{\mathrm{E}}\},$$

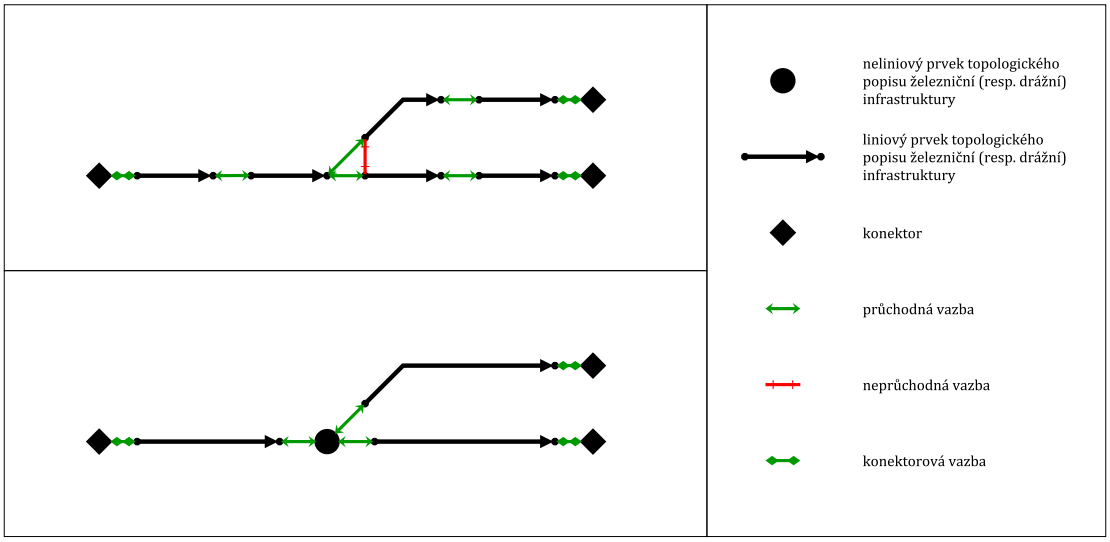
kde každá struktura $\dot{s}^{\mathrm{E}} \in \mathcal{S}^{\mathrm{E}}$, tedy s_i^{E}, $i \in \{1, 2, \dots k\}$, je substrukturou struktury s^{E} a k je celkový počet struktur množiny $\mathcal{S}^{\mathrm{E}}$.

Zavedená struktura systému topologického popisu sítě $\mathbb{T}$, která je v kontextu úlohy integrace nebo dezintegrace prvků systému topologického sítě $\mathbb{T}$ řešené v rámci určitého aplikačního případu w známá již před zahájením řešení této úlohy a je vyjádřena na referenční rozlišovací úrovni l^{R}, bude v tomto kontextu vystupovat jako referenční struktura a budeme pro ni používat označení s^{R}.

Takové zavedené struktury, které nejsou před zahájením úlohy integrace nebo dezintegrace prvků systému topologického popisu $\mathbb{T}$ známy a které mají být získány a vyjádřeny na odpovídajících odvozených rozlišovacích úrovních l^{D} až jako výstup příslušné integrační nebo dezintegrační úlohy řešené v rámci určitého aplikačního případu w, budou ve vztahu k této úloze vystupovat v roli odvozené struktury a budeme pro ně používat označení s^{D}.

Takové odvozené struktury, které mají být získány jako výstup úlohy integrace prvků systému topologického popisu sítě a vyjádřeny na odpovídající méně podrobné rozlišovací úrovni l^{G}, budou ve vztahu k této úloze (resp. k příslušné referenční struktuře s^{R}) vystupovat v roli méně podrobné struktury a budeme pro ně používat označení s^{G}.

Takové odvozené struktury, které mají být získány jako výstup úlohy integrace prvků systému topologického popisu sítě a vyjádřeny na odpovídající méně podrobné rozlišovací úrovni l^{F}, budou ve vztahu k této úloze (resp. k příslušné referenční struktuře s^{R}) vystupovat v roli méně podrobné struktury a budeme pro ně používat označení s^{F}.



Obrázek 3.1. Příklad různých způsobů modelování výhybky v rámci konektory oddělené části železniční infrastruktury.

3.9.1 Typ přípustné struktury

V rámci daného aplikačního případu w, je pro každou zavedenou rozlišovací úroveň l^{E} nutné specifikovat typ přípustné struktury, který musí struktura s^{E} respektovat. Typ přípustné struktury specifikovaný pro rozlišovací úroveň l^{E} budeme označovat jako $\tau_{\mathrm{S}}^{\mathrm{E}}$. Specifikace typu přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{E}}$ musí být v souladu se způsobem vyjádření sítě n uvažovaným pro zavedenou rozlišovací úroveň l^{E} v rámci daného aplikačního případu w a reflektovat stupeň podrobnosti odpovídající této zavedené rozlišovací úrovně. Specifikace zahrnuje následující kroky:

- stanovení tříd prvků, které mohou být součástí struktury s^{E}
- pro každou z uvažovaných tříd prvků struktury s^{E} rozhodnutí, zda mají být v rámci této třídy rozlišovány funkční typy prvků a případně které
- stanovení typů vazeb z hlediska průchodnosti, které mohou být součástí struktury s^{E}
- stanovení dalších veličin a jejich přípustných hodnot, které mohou být použity pro popis prvků struktury s^{E} (obvykle v závislosti na jejich třídě)
- stanovení okolností, za jakých má být objekt reálného světa reprezentován prvek struktury s^{E} dané třídy, funkčního typu a hodnot dalších veličin
- stanovení pravidel, která musí být respektována při vytváření vazeb mezi prvky struktury s^{E} (také s ohledem na typ vazeb z hlediska průchodnosti a obvykle v závislosti třídě a funkčním typu propojovaných prvků, případně také na hodnotách dalších veličin)

V rámci této práce rozlišujeme dvě třídy prvků topologického popisu železniční (resp. drážní) infrastruktury, které mohou být součástí struktury s^{E}. Jedná se o třídy prvků neliniových a liniových. Na základě přípustného zastoupení prvků těchto tříd v zavedené struktuře s^{E} budeme rozlišovat mezi následujícími typy přípustné struktury:

- stejnorodá struktura neliniových prvků
- stejnorodá struktura liniových prvků
- různorodá struktura

Součástí stejnorodé struktury neliniových prvků mohou být pouze takové prvky, které jsou prvky třídy neliniových prvků. Strukturu tohoto typu vyjádřenou na zavedené rozlišovací úrovni l^{E} můžeme označit jako s^{EN}. Součástí stejnorodé struktury liniových prvků mohou být pouze takové prvky, které jsou prvky třídy liniových prvků. Strukturu tohoto typu vyjádřenou na zavedené rozlišovací úrovni l^{E} můžeme označit jako s^{EL}. Součástí různorodé struktury mohou být jak prvky třídy neliniových prvků, tak prvky třídy liniových prvků. Pro strukturu tohoto typu vyjádřenou na zavedené rozlišovací úrovni l^{E} budeme používat obecné označení s^{E}.

Prvky různých funkčních typů mohou společně vystupovat ve struktuře s^{E} pouze za specifických okolností. Nutnou podmínkou je, aby se jednalo vzájemně komplementární funkční typy prvků. Pomocí všech prvků vzájemně komplementárních funkčních typů, které jsou součástí struktury s^{E}, je nezbytné na rozlišovací úrovni l^{E} dokázat popsat síť n z topologického hlediska zcela, souvisle a jednoznačně.

Příklady funkčních typů prvků, které mohou společně utvářet zavedenou strukturu s^{E}, shrnuje tabulka 3.2. Tabulka byla sestavena tak, aby kombinace funkčních typů

funkční typy neliniových prvků	**funkční typy liniových prvků**
síť	–
oblast v rámci sítě (např. tarifní zóna)	–
dopravna v uzlovém bodě sítě (např. stanice odbočné, křižovatkové, . . .)	traťový úsek mezi dopravnami v uzlových bodech sítě
dopravna s kolejovým rozvětvením (např. stanice)	traťový úsek mezi dopravnami s kolejovým rozvětvením
dopravna s kolejovým rozvětvením (např. stanice)	traťová kolej (v dopravním smyslu) mezi dopravnami
dopravna (ne nutně s kolejovým rozvětvením)	traťový úsek mezi dopravnami
zhlaví v rámci dopravny	kolejová skupina, traťový úsek mezi dopravnami
napájecí stopa	–
oblast kontroly volnosti koleje	–
–	kolejová trasa
výhybka (výhybková konstrukce) v dopravním smyslu	kolej v dopravním smyslu
výhybka (výhybková konstrukce) ve stavebním smyslu	kolej ve stavebním smyslu
–	výhybka (výhybková konstrukce) ve stavebním smyslu, kolej ve stavebním smyslu
–	větev výhybky, kolej ve stavebním smyslu
–	rameno výhybky, kolej ve stavebním smyslu
–	kolejnice
srdcovka	kolejnice

Tabulka 3.2. Příklady komplementárních funkčních typů prvků.

prvků uvedené na společných řádcích výše nalézaly uplatnění spíše ve strukturách vyjádřených na méně podrobných rozlišovacích úrovních, zatímco kombinace funkčních typů prvků uvedené na společných řádcích níže nalézaly uplatnění spíše ve strukturách vyjádřených na podrobnějších rozlišovacích úrovních. Toto pravidlo opět chápejme spíše orientačně. Tabulka je rozdělena do dvou sloupců, zařazení jednotlivých funkčních typů prvků do nichž pro každý řádek vyjadřuje, jaký vztah se při příslušné specifikaci typu přípustné struktury předpokládá mezi třídou a funkčním typem prvku. V některých případech řádek obsahuje pouze jeden funkční typ prvku. V takovém případě jsou všechny prvky uvažované struktury s^{E} prvky tohoto funkčního typu a zároveň přísluší třídě, v rámci jejíhož sloupce je daný funkční typ v tomto řádku zařazen. V těchto případech by struktura s^{E} byla stejnorodá (buď liniových nebo neliniových prvků) a navíc složená z prvků stejného funkčního typu. Ne zřídkakdy jsou uvedeny dva funkční typy prvků,

z nichž jeden přísluší třídě prvků neliniových a druhý třídě prvků liniových. Je zřejmé, že je možné si představit takové příklady, kdy jsou jednou prvky různých funkčních typů prvky různých tříd a podruhé prvky těch samých různých funkčních typů prvky stejné třídy. V některých případech jedna ze tříd (při podrobnějším členění funkčního typů prvků by k tomu mohlo dojít i u obou ze tříd) může zahrnovat prvky více funkčních typů. Neměla by však nastat situace, že by v rámci té samé specifikace typu přípustné struktury byl jeden funkční typ prvku příslušel jak prvkům neliniovým, tak prvkům liniovým.

3.9.2 Rozšířená struktura

V některých případech budeme potřebovat strukturu s_i^{E} (např. v roli substruktury struktury s^{E} odpovídající vnitřní struktuře prvku a_i^{G} figurujícího na méně podrobné rozlišovací úrovni) rozšířit o množinu konektorů $\mathcal{B}_i^{\mathrm{E}}$, s jejichž pomocí je možné tuto strukturu propojit s jinými strukturami, a o množinu konektorových vazeb $\hat{\mathcal{R}}_i^{\mathrm{E}}$, propojujících tyto konektory s prvky jiných substruktur struktury s^{E}.

Proto zavedeme rozšířenou strukturu $\tilde{s}_i^{\mathrm{E}}$ struktury s_i^{E}. Rozšířenou strukturu $\tilde{s}_i^{\mathrm{E}}$ zapíšeme jako

$$\tilde{s}_i^{\mathrm{E}} = \left(\mathcal{A}_i^{\mathrm{E}}, \tilde{\mathcal{R}}_i^{\mathrm{E}}, \mathcal{B}_i^{\mathrm{E}}\right),$$

kde $\mathcal{A}_i^{\mathrm{E}}$ je množina prvků struktury s_i^{E}, $\tilde{\mathcal{R}}_i^{\mathrm{E}}$ je rozšířená množina vazeb struktury struktury s_i^{E}, kterou můžeme vyjádřit jako

$$\tilde{\mathcal{R}}_i^{\mathrm{E}} = \mathcal{R}_i^{\mathrm{E}} \cup \hat{\mathcal{R}}_i^{\mathrm{E}},$$

kde $\mathcal{R}_i^{\mathrm{E}}$ je množina vazeb struktury s_i^{E} a $\hat{\mathcal{R}}_i^{\mathrm{E}}$ je množina konektorových vazeb rozšířené struktury $\tilde{s}_i^{\mathrm{E}}$, a $\mathcal{B}_i^{\mathrm{E}}$ je množina konektorů rozšířené struktury $\tilde{s}_i^{\mathrm{E}}$.

3.9.3 Vzory vnitřní struktury

Při hledání vnitřní struktury prvků budeme využívat předdefinovaných vzorů vnitřní struktury (resp. rozšířených vzorů vnitřní struktury). Množinu vzorů vnitřní struktury (resp. rozšířených vzorů vnitřní struktury), které mohou být použity k vytvoření substruktury (resp. rozšířené substruktury) struktury systému $\mathbb{T}$ vyjádřené na zavedené rozlišovací úrovni l^{E}, budeme označovat jako ${}^*\mathcal{S}^{\mathrm{E}}$ (resp. ${}^*\tilde{\mathcal{S}}^{\mathrm{E}}$). Konkrétní vzor vnitřní struktury použitý pro vytvoření struktury s_i^{E} (resp. rozšířené struktury $\tilde{s}_i^{\mathrm{E}}$) pak bude označen jako ${}^*s_i^{\mathrm{E}}$ (resp. ${}^*\tilde{s}_i^{\mathrm{E}}$).

Vzory (rozšířené) vnitřní struktury určené k vyjádření (rozšířené) vnitřní struktury prvků figurujících na referenční rozlišovací úrovni l^{R} také na podrobnější rozlišovací úrovni l^{F} budou popisovány veličinami, na základě jejichž hodnot bude možné posuzovat regularitu těchto vzorů ve věci uvedeného záměru vůči jednotlivým těmto prvkům. Tyto veličiny budou definovány takovým způsobem, aby na základě porovnání jejich hodnot s hodnotami veličin, jimiž jsou popsány prvky figurující na referenční rozlišovací úrovni l^{R} bylo možné posoudit, zda je příslušný vzor (rozšířené) vnitřní struktury možné použít k vyjádření (rozšířené) vnitřní struktury jednotlivých z těchto prvků. Speciální veličinou zaváděnou za tímto účelem pro vzory rozšířené vnitřní struktury je veličina udávající přípustné konfigurace konektorových skupin příslušného vzoru rozšířené vnitřní struktury, které vyjadřují, jakým způsoby by tyto konektory mohly být integrovány v konektory rozlišovací úrovně, na níž figurují prvky, vůči nimž má být regularita těchto vzorů posuzována. Tuto veličinu budeme nazývat přípustné konfigurace konektorových skupin a označovat jako δ. Bude mít charakter množiny přípustných konfigurací konektorových skupin.

3.10 Kolekce

Za účelem vyjádřit, z jakých prvků podrobnější struktury se skládá určitý prvek méně podrobné struktury, budeme využívat kolekce. Každou kolekci je nutné chápat ve vztahu ke dvěma rozlišovacím úrovním (resp. odpovídajícím strukturám systému topologického popisu sítě $\mathbb{T}$), z nichž jedna je podrobnější a druhá méně podrobná. Kolekce budeme prakticky využívat k vyjádření skutečnosti, v které prvky méně podrobné struktury s^{G} byly integrovány prvky referenční rozlišovací úrovně s^{R} nebo které prvky podrobnější struktury s^{F} vznikly dezintegrací kterých prvků referenční struktury s^{R}. Budeme rozlišovat dvě třídy kolekcí, a sice kolekce neuspořádané a uspořádané.

Neuspořádanou kolekci, vůči níž vystupuje prvek a_i^{G} méně podrobné rozlišovací úrovně l^{G} v roli celku a prvky podrobnější rozlišovací úrovně l^{F}, které jsou prvky matice $\mathcal{A}_i^{\mathrm{F}}$, v roli částí, zapíšeme jako uspořádanou dvojici tohoto prvku a této množiny:

$$c_i^{\mathrm{U}} = (a_i^{\mathrm{G}}, \mathcal{A}_i^{\mathrm{F}})\,.$$

Uspořádanou kolekci, vůči níž vystupuje prvek a_i^{G} méně podrobné rozlišovací úrovně l^{G} v roli celku a prvky podrobnější rozlišovací úrovně l^{F}, které jsou prvky uspořádané h-tice $^{\circ}\mathcal{A}_i^{\mathrm{F}}$, v roli částí, zapíšeme jako uspořádanou dvojici tohoto prvku a této uspořádané h-tice:

$$c_i^{\mathrm{O}} = (a_i^{\mathrm{G}}, {}^{\circ}\mathcal{A}_i^{\mathrm{F}})\,.$$

Množinu kolekcí definovaných v rámci systému topologického popisu sítě $\mathbb{T}$ budeme obecně označovat jako $\mathcal{C}$. Množinu $\mathcal{C}$ lze rozepsat jako

$$\mathcal{C} = \{c_1, c_2, \ldots c_k\}\,,$$

kde každá kolekce $c \in \mathcal{C}$, tedy c_i, $i \in \{1, 2, \ldots k\}$, je kolekce definovaná v rámci systému $\mathbb{T}$ a k je celkový počet kolekcí definovaných v rámci systému $\mathbb{T}$ množiny $\mathcal{C}$.

V případě potřeby vyjádřit skutečnost, že množina kolekcí $\mathcal{C}$ obsahuje pouze neuspořádané kolekce, resp. pouze kolekce uspořádané, využijeme pro ni označení $\mathcal{C}^{\mathrm{U}}$, resp. $\mathcal{C}^{\mathrm{O}}$. Obdobně můžeme tuto skutečnost vyjádřit pro jakoukoliv množinu $\mathcal{C}_{\mathrm{S}}$, pro níž platí $\mathcal{C}_{\mathrm{S}} \subset \mathcal{C}$, a to s využitím označení $\mathcal{C}_{\mathrm{S}}^{\mathrm{U}}$, resp. $\mathcal{C}_{\mathrm{S}}^{\mathrm{O}}$.

Kapitola 4
Nástroje transformace struktury

Cílem této práce je navrhnout metodiku integrace a dezintegrace prvků topologického popisu železniční infrastruktury. Tato metodika má být aplikovatelná na systém topologického popisu sítě $\mathbb{T}$, který byl na vysoce obecné úrovni popsán v kapitole 3. Součástí návrhu metodiky bude definování rámcových pravidel pro systematický návrh transformačních algoritmů, v souladu s nimiž budou realizovány transformační procesy. Při návrhu transformačních algoritmů bude jako stavebních bloků využíváno univerzálněji navržených operací a kroků, v souladu s nimiž budou realizovány subprocesy a události transformačních procesů.

4.1 Transformační algoritmy

Transformačním algoritmem t_{RD} budeme v kontextu této práce nazývat algoritmus navržený pro daný aplikační případ w_{RD}, jehož cílem je zajistit, aby na základě referenční struktury s^{R} vznikla odvozená struktura s^{D}. Transformační algoritmus t_{RD} vyžijeme v případě, že je žádoucí strukturu strukturu systému topologického popisu sítě $\mathbb{T}$ vyjádřenou na referenční rozlišovací úrovni l^{R}, vyjádřit také na méně odvozené rozlišovací úrovni l^{D}. Realizaci transformačního algoritmu za definovaných vstupů budeme nazývat transformačním procesem, příp. též transformací.

Takový transformační algoritmus, jehož cílem je zajistit, aby na základě referenční struktury s^{R} vznikla méně podrobná struktura s^{G}, budeme nazývat integračním algoritmem. Integrační algoritmus t_{RG} vyžijeme v případě, že je žádoucí strukturu strukturu systému topologického popisu sítě $\mathbb{T}$ vyjádřenou na referenční rozlišovací úrovni l^{R}, vyjádřit také na méně podrobné rozlišovací úrovni l^{G}. Realizaci integračního algoritmu za definovaných vstupů budeme nazývat integračním procesem, příp. též integrací.

Při integraci jsou prvky struktury s^{R} slučovány v prvky struktury s^{G}. Uvažujme, že integrací prvků struktury s^{R}, pro které platí, že náleží do množiny $\mathcal{A}_i^{\mathrm{R}} \subset \mathcal{A}^{\mathrm{R}}$, získáváme prvek a_i^{G} struktury s^{G}. Jelikož je splněna podmínka, že je každému prvku množiny $\mathcal{A}^{\mathrm{R}}$ přiřazen nejvýše jeden prvek množiny $\mathcal{A}^{\mathrm{G}}$, je možné vztah mezi prvky množiny $\mathcal{A}^{\mathrm{R}}$ a prvky množiny $\mathcal{A}^{\mathrm{G}}$ popsat jako zobrazení z množiny $\mathcal{A}^{\mathrm{R}}$ do množiny $\mathcal{A}^{\mathrm{G}}$. Při integraci zároveň dochází ke vzniku vazeb struktury s^{G}. Vazby struktury s^{G} vznikají na základě sloučení některých vazeb struktury s^{R}. Některé vazby struktury s^{R} se při vzniku vazeb struktury s^{G} nemusejí navenek vůbec projevit, protože jsou integrovány do prvků této struktury.

Takový transformační algoritmus, jehož cílem je zajistit, aby na základě referenční struktury s^{R} vznikla podrobnější struktura s^{F}, budeme nazývat dezintegračním algoritmem. Dezintegrační algoritmus t_{RG} vyžijeme v případě, že je žádoucí strukturu strukturu systému topologického popisu sítě $\mathbb{T}$ vyjádřenou na referenční rozlišovací úrovni l^{R}, vyjádřit také na méně podrobné rozlišovací úrovni l^{F}. Realizaci dezintegračního algoritmu za definovaných vstupů budeme nazývat dezintegračním procesem, příp. též dezintegrací.

Při dezintegraci jsou prvky struktury s^{R} rozčleňovány do prvků struktury s^{F}. Uvažujme, že dezintegrací prvku $\mathcal{A}_i^{\mathrm{R}}$ struktury s^{R}, získáváme prvky struktury s^{F}, pro které platí, že náleží do množiny $\mathcal{A}_i^{\mathrm{F}} \subset \mathcal{A}^{\mathrm{F}}$. Jelikož je splněna podmínka, že je každému prvku množiny $\mathcal{A}^{\mathrm{F}}$ přiřazen nejvýše jeden prvek množiny $\mathcal{A}^{\mathrm{R}}$, je možné vztah mezi prvky množiny $\mathcal{A}^{\mathrm{F}}$ a prvky množiny $\mathcal{A}^{\mathrm{R}}$ popsat jako zobrazení z množiny $\mathcal{A}^{\mathrm{F}}$ do množiny $\mathcal{A}^{\mathrm{R}}$. Při dezintegraci zároveň dochází ke vzniku vazeb struktury s^{F}. Některé vazby struktury s^{F} vznikají rozčleněním vazeb struktury s^{R}. Jiné vazby struktury s^{F} ale mohou vzniknout i na základě prvků struktury s^{R}, do nichž jsou v této struktuře integrovány.

Ne pro každý transformační algoritmus musíme být nutně schopni (zejména úrovni jeho obecného návrhu) rozlišit, zda se jedná o algoritmus integrační nebo dezintegrační. Skutečnost, zda je odvozená struktura s^{D} vyjádřena na méně podrobné nebo na podrobnější rozlišovací úrovni, než referenční struktura s^{R}, může vyplynout až na základě aplikace daného transformačního algoritmu na konkrétní datový vzorek představující vstupy, za kterých dochází k realizaci konkrétního transformačního procesu. Míry podrobnosti rozlišovacích úrovní l^{R} a l^{D} mohou nicméně zůstat neporovnatelné i po aplikaci transformačního algoritmu na konkrétní datový vzorek a realizaci konkrétního transformačního procesu. V takových případech zůstaneme u používání obecných pojmů transformační algoritmus a transformační proces.

4.2 Operace

Aby došlo k částečné homogenizaci a zpřehlednění zápisu transformačních algoritmů navrhovaných pro jednotlivé aplikační případy, budeme při jejich návrhu využívat standardizovaných operací. Operaci budeme chápat jako subalgoritmus určený k provedení dílčího úkolu, který může být využit v rámci transformačního algoritmu využit k transformaci uspořádané x-tice vstupů na uspořádanou y-tici výstupů. Realizace operace v rámci transformačního algoritmu t_{RD} je podprocesem procesu realizováno tímto transformačním algoritmem t_{RD}.

Operaci, jejímž cílem je zajistit, aby na základě struktury vyjádřené na referenční rozlišovací úrovni l^{R} vznikla struktura vyjádřená na odvozené rozlišovací úrovni l^{D}, budeme nazývat transformační operací. Pokud je míra podrobnosti odvozené rozlišovací úrovně l^{D} menší než míra podrobnosti referenční rozlišovací úrovně l^{R}, a lze ji tedy označit jako méně podrobnou rozlišovací úroveň l^{F}, budeme hovořit o integrační operaci. Pokud je míra podrobnosti odvozené rozlišovací úrovně l^{D} větší než míra podrobnosti referenční rozlišovací úrovně l^{R}, a lze ji tedy označit jako podrobnější rozlišovací úroveň l^{F}, budeme hovořit o integrační operaci. Nemusí se (a zpravidla nebude) jednat o referenční strukturu s^{R} a odvozenou strukturu s^{D}, čili buď méně podrobnou strukturu s^{G}, nebo podrobnější strukturu s^{F}, tedy kompletní struktury systému $\mathbb{T}$, vyjádřené na příslušných rozlišovacích úrovních. Zpravidla se bude jednat o určitým způsobem definované substruktury těchto struktur, resp. o rozšířené substruktury těchto struktur.

Také označení rozlišovacích úrovní je nutné chápat pouze lokálně. Ne vždy bude možné tato označení ztotožnit s označeními používanými na úrovni transformačního algoritmu, který příslušné transformační operace využívá. V rámci některých transformačních algoritmů může být využíváno pomocných rozlišovacích úrovní a struktur systému $\mathbb{T}$ na nich vyjádřených, které mohou vzhledem k některým transformačním operacím vystupovat v roli odvozené rozlišovací úrovně a vůči jiným naopak v roli referenční rozlišovací úrovně. Právě to je důvod, proč u takovýchto transformačních algoritmů nemusíme být vždy schopni rozlišit, zda se jedná o algoritmus integrační nebo

o algoritmus dezintegrační. Transformační operace budou nicméně navrhovány tak, aby bylo vždy možné rozlišit, zda se jedná o operaci integrační nebo dezintegrační. U odvozené rozlišovací úrovně l^{D} bude v rámci dané transformační operace vždy zřejmé, zda se jedná o rozlišovací úroveň méně podobnou než referenční rozlišovací úroveň l^{R}, nebo o rozlišovací úroveň podrobnější.

Každý transformační algoritmus bude navržen tak, aby bylo možné jej rozčlenit na dále nedělitelné kroky. Operaci, která je vyjádřena jedním krokem transformačního algoritmu, budeme nazývat elementární operací. Elementární operace může být začleněna buď přímo v těle vlastního transformačního algoritmu t_{RD}, nebo být součástí některé z operací, která není elementární operací. Nedělitelnost elementární transformace je nutné chápat relativně. Závisí na rozlišovací úrovni, se kterou transformační algoritmus navrhujeme. V rámci této práce budeme při návrhu transformačních algoritmů využívat takovou rozlišovací úroveň, na níž se jednotlivé transformační operace zpravidla vztahují k jednotlivým objektům systému $\mathbb{T}$ (např. vytvoření nového objektu). Stejně jako jakákoliv jiná operace bude i elementární operace využívána k transformaci uspořádané x-tice vstupů na uspořádanou y-tici výstupů, na rozdíl od operace, které není operací elementární, už však v rámci elementární operace nebude blíže specifikováno, jakým způsobem k této transformaci dochází. Realizace elementární operace v rámci transformačního algoritmu t_{RD} je událostí procesu realizováno tímto transformačním algoritmem t_{RD}. Realizace kroku v rámci transformační operace je událostí podprocesu realizováno touto transformační operací t_{RD}.

Vyjádříme-li transformační algoritmus t_{RD} s jako aktivitu s využitím diagramu aktivit UML, budeme transformační operaci, která není elementární operací, reprezentovat jako vnořenou aktivitu a elementární operaci jako akci.

Kapitola 5
Kategorizace a popis operací

V rámci této kapitoly budou představeny některé operace, které je možné využít jako stavební bloky pro návrh integračních a dezintegračních algoritmů. V případě elementárních transformací bude uváděn pouze stručný popis operace, přehled vstupů a výstupů a zápis, pomocí kterého může být vnořena do jiné operace nebo do transformačního algoritmu.

5.1 Operace vytváření objektů

Operace vytváření objektů je operací, při které dochází k vytvoření nového objektu o, jehož třída, která mu je při této operaci přidělena, je specifikována v rámci konkretní operace vytváření objektů. Spolu s objektem o mohou být za účelem zajištění konzistence dat na meziobjektové úrovni vytvořeny i další související objekty. Objekt, jemuž je při vytváření přidělena hodnota identifikátoru, může být přiřazen síti a rozlišovací úrovni. V závislosti na konkrétní operaci vytváření objektů mohou být objektu přiděleny i hodnoty dalších veličin.

5.1.1 Vytvořit neliniový prvek

Elementární operace `Vytvořit neliniový prvek` je operací vytváření objektů, která slouží k vytvoření nového neliniového prvku a^{EN} sítě n figurujícího na rozlišovací úrovni l^{E} a k jeho přiřazení síti n a rozlišovací úrovni l^{E}. Prvku a^{EN} je při této operaci zároveň přidělen identifikátor a příslušnost k třídě neliniových prvků.

Vstupy:

- n – síť, jíž má být prvek a^{EN} přiřazen
- l^{E} – zavedená rozlišovací úroveň, jíž má být prvek a^{EN} přiřazen

Výstupy:

- a^{EN} – nově vytvořený neliniový prvek

Zápis:

- $(a^{\mathrm{EN}}) \leftarrow \texttt{VytvoritNeliniovyPrvek}(n, l^{\mathrm{E}})$

5.1.2 Vytvořit liniový prvek

Elementární operace `Vytvořit liniový prvek` je operací vytváření objektů, která slouží k vytvoření nového liniového prvku a^{EL} sítě n figurujícího na rozlišovací úrovni l^{E} a k jeho přiřazení síti n a rozlišovací úrovni l^{E}. Prvku a^{EL} je při této operaci zároveň přidělen identifikátor a příslušnost k třídě liniových prvků.

Vstupy:

- n – síť, jíž má být prvek a^{EL} přiřazen
- l^{E} – zavedená rozlišovací úroveň, jíž má být prvek a^{EL} přiřazen

Výstupy:

- a^{EL} – nově vytvořený liniový prvek

Zápis:

- $(a^{\mathrm{EL}}) \leftarrow \texttt{VytvoritLiniovyPrvek}(n, l^{\mathrm{E}})$

5.1.3 Vytvořit konektor

Elementární operace `Vytvořit konektor` je operací vytváření objektů, která slouží k vytvoření nového konektoru b^{E} sítě n figurujícího na rozlišovací úrovni l^{E} a k jeho přiřazení síti n a rozlišovací úrovni l^{E}. Konektoru b^{E} je při této operaci zároveň přidělen identifikátor a příslušnost k třídě konektorů.

Vstupy:

- n – síť, jíž má být konektor b^{E} přiřazen
- l^{E} – zavedená rozlišovací úroveň, jíž má být konektor b^{E} přiřazen

Výstupy:

- b^{E} – nově vytvořený konektor

Zápis:

- $(b^{\mathrm{E}}) \leftarrow \texttt{VytvoritKonektor}(n, l^{\mathrm{E}})$

5.1.4 Vytvořit vazbu mezi póly prvku

Elementární operace `Vytvořit vazbu mezi póly prvku` je operací vytváření objektů, která slouží k vytvoření nové vazby r^{E} o průchodnosti η sítě n figurující na rozlišující úrovni l^{E} propojující prvky $a_{\mathrm{x}}^{\mathrm{E}}$ a $a_{\mathrm{y}}^{\mathrm{E}}$, a to s využitím pólů těchto prvků $q_a(a_{\mathrm{x}}^{\mathrm{E}})$ a $q_b(a_{\mathrm{y}}^{\mathrm{E}})$, a k jejímu přiřazení síti n a rozlišovací úrovni l^{E}. Prvku a^{R} je při této operaci zároveň přidělen identifikátor a příslušnost k třídě vazeb.

Vstupy:

- $q_a(a_{\mathrm{x}}^{\mathrm{E}})$ – pól prvku $a_{\mathrm{x}}^{\mathrm{E}}$, který má být propojen vazbou r^{E} s pólem prvku $a_{\mathrm{x}}^{\mathrm{E}}$
- $q_b(a_{\mathrm{y}}^{\mathrm{E}})$ – pól prvku $a_{\mathrm{y}}^{\mathrm{E}}$, který má být propojen vazbou s pólem prvku $a_{\mathrm{x}}^{\mathrm{E}}$
- η – průchodnost vazby
- n – síť, jíž má být vazba r^{E} přiřazena
- l^{E} – zavedená rozlišovací úroveň, jíž má být vazba r^{E} přiřazena

Výstupy:

- r^{E} – nově vytvořená vazba

Zápis:

- $(r^{\mathrm{E}}) \leftarrow \texttt{VytvoritVazbuMeziPolyPrvku}\left(q_a(a_{\mathrm{x}}^{\mathrm{E}}), q_b(a_{\mathrm{y}}^{\mathrm{E}}), \eta, n, l^{\mathrm{E}}\right)$

5.1.5 Vytvořit neuspořádanou kolekci

Elementární operace `Vytvořit neuspořádanou kolekci` je operací vytváření objektů, která slouží k vytvoření nové neuspořádané kolekce c^{U}, vůči níž vystupuje v roli celku prvek méně podrobné rozlišovací úrovně a^{G} a v roli částí prvky podrobnější rozlišovací úrovně množiny $\mathcal{A}^{\mathrm{F}}$. Neuspořádané kolekci c^{U} je při této operaci zároveň přidělen identifikátor a příslušnost k třídě neuspořádaných kolekcí.

Vstupy:

- a^{G} – prvek méně podrobné rozlišovací úrovně, vůči neuspořádané kolekci c^{U} vystupuje v roli celku
- $\mathcal{A}^{\mathrm{F}}$ - množina prvků podrobnější rozlišovací úrovně, které vůči neuspořádané kolekci c^{U} vystupují v roli roli částí

Výstupy:

- c^{U} – nově vytvořená neuspořádaná kolekce

Zápis:

- $(c^{\mathrm{U}}) \leftarrow \texttt{VytvoritNeusporadanouKolekci}(a^{\mathrm{G}}, \mathcal{A}^{\mathrm{F}})$

5.1.6 Vytvořit uspořádanou kolekci

Elementární operace `Vytvořit uspořádanou kolekci` je operací vytváření objektů, která slouží k vytvoření nové uspořádané kolekce c^{O}, vůči vystupuje v roli celku prvek méně podrobné rozlišovací úrovně a^{G} a v roli částí prvky podrobnější rozlišovací úrovně uspořádané k-tice $^{\circ}\mathcal{A}^{\mathrm{F}}$. Uspořádané kolekci c^{O} je při této operaci zároveň přidělen identifikátor a příslušnost k třídě uspořádaných kolekcí.

Vstupy:

- a^{G} – prvek méně podrobné rozlišovací úrovně, který vůči uspořádané kolekci c^{O} vystupuje v roli celku
- $^{\circ}\mathcal{A}^{\mathrm{F}}$ – uspořádaná k-tice prvků podobnější rozlišovací úrovně, které vůči uspořádané kolekci c^{O} vystupují v roli částí

Výstupy:

- c^{O} – nově vytvořená neuspořádaná kolekce

Zápis:

- $(c^{\mathrm{O}}) \leftarrow \texttt{VytvoritUsporadanouKolekci}(a^{\mathrm{G}}, {}^{\circ}\mathcal{A}^{\mathrm{F}})$

5.1.7 Vytvořit substrukturu na základě vzoru

Elementární operace `Vytvořit substrukturu na základě vzoru` je operací vytváření objektů, která slouží k vytvoření nové substruktury $\tilde{s}_i^{\mathrm{E}}$ (obecně se může jednat o rozšířenou substrukturu) rozlišovací úrovně l^{E} sítě *eln* na základě vzoru vnitřní struktury ${}^{*}\tilde{s}_i^{\mathrm{E}}$ (obecně se může jednat o vzor rozšířené struktury), pro kterou platí, že je s tímto vzorem isomorfní. Na jednotlivé komponenty substruktury $\tilde{s}_i^{\mathrm{E}}$ jsou přeneseny také veškeré hodnoty veličin příslušných komponent vzoru vnitřní struktury ${}^{*}\tilde{s}_i^{\mathrm{E}}$ kromě hodnot identifikátorů, a hodnot veličin, které se na tyto identifikátory odkazují. Hodnoty identifikátorů jsou přiděleny nové a hodnoty veličin, které se na ně odkazují, jsou přiděleny v souladu s nimi tak, aby byl dodržen isomorfismus i na této úrovni.

Vstupy:

- ${}^{*}\tilde{s}_i^{\mathrm{E}}$ – vzor vnitřní struktury, na jejímž základě má být vytvořena substruktura $\tilde{s}_i$
- n – síť, jíž mají být komponenty substruktury $\tilde{s}_i$ přiřazeny
- l^{E} – zavedená rozlišovací úroveň, jíž mají být komponenty substruktury $\tilde{s}_i^{\mathrm{E}}$ přiřazeny

Výstupy:

- $\tilde{s}_i^{\mathrm{E}}$ – nově vytvořená substruktura

Zápis:

- $(\tilde{s}_i^{\mathrm{E}}) \leftarrow$ `VytvoritSubstrukturuNaZakladeVzoru`$({}^{*}\tilde{s}_i^{\mathrm{E}}, n, l^{\mathrm{E}})$

5.2 Operace odstraňování objektů

Operace odstraňování objektů je operací, při které dochází k odstranění stávajícího objektu o. Společně s objektem o zaniká i přidělení hodnot jednotlivých veličin objektu o a mohou s ním být odstraněny i další objekty, které jsou na něj vázané. Hromadná operace odstraňování objektů je operací, při které dochází k odstranění stávajících objektů množiny $\mathcal{O}$.

5.2.1 Odstranit objekt

Elementární operace `Odstranit objekt` je operací odstraňování objektů, která slouží k odstranění objektu o.

Vstupy:

- o – objekt, který má být odstraněn

Zápis:

`OdstranitObjekt`(o)

5.2.2 Odstranit objekty

Operace `Odstranit objekty` je hromadnou operací odstraňování objektů, která slouží k odstranění objektů množiny $\mathcal{O}$.

Vstupy:

- $\mathcal{O}$ – množina, jejíž objekty mají být odstraněny

Zápis:

- `OdstranitObjekty`$(\mathcal{O})$

Operace : `OdstranitObjekty`$(\mathcal{O})$

---

pro všechny $o \in \mathcal{O}$ **proveď**

`OdstranitObjekty`$(\mathcal{O})$

ukonči pro všechny

5.3 Operace s hodnotami veličin

Operace s hodnotami veličin je operací, při které dochází ke generování, přiřazování a dalším manipulacím s hodnotami veličin.

5.3.1 Přiřadit hodnotu veličiny objektu

Elementární operace `Přiřadit hodnotu veličiny objektu` je operací s hodnotami veličin, která slouží k přiřazení hodnoty v veličiny m objektu o. Objekt o je operací navrácen jako objekt o' s přiřazenou hodnotou v veličiny m.

Vstupy:

- m – veličina, jejíž hodnota má být přiřazena objektu o
- v – hodnota veličiny m, která má být přiřazena objektu o
- o – objekt, jemuž má být přiřazena hodnota v veličiny m

Výstupy:

- o' – objekt o po přiřazení hodnoty v veličiny m

Zápis:

- $(o') \leftarrow$ `PriraditHodnotuVelicinyObjektu`(m, v, o)

5.3.2 Přiřadit hodnotu veličiny objektům

Operace `Přiřadit hodnotu veličiny objektům` je hromadnou operací s hodnotami veličin, která slouží k přiřazení hodnoty v veličiny m objektům množiny $\mathcal{O}$. Množina $\mathcal{O}$ je operací navrácena jako množina $\mathcal{O}'$ objektů s přiřazenou hodnotou v veličiny m.

Vstupy:

- m – veličina, jejíž hodnota má být přiřazena objektům množiny $\mathcal{O}$
- v – hodnota veličiny m, která má být přiřazena objektům množiny $\mathcal{O}$
- $\mathcal{O}$ – množina, jejímž objektům má být přiřazena hodnota v veličiny m

Výstupy:

- $\mathcal{O}'$ – množina objektů množiny $\mathcal{O}$ po přiřazení hodnoty v veličiny m těmto objektům

Zápis:

- $(\mathcal{O}') \leftarrow$ `PriraditHodnotuVelicinyObjektum`$(m, v, \mathcal{O})$

Operace : `PriraditHodnotuVelicinyObjektum`$(m, v, \mathcal{O})$

$\mathcal{O}' \leftarrow \emptyset$

pro všechny $o \in \mathcal{O}$ **proveď**

$(o') \leftarrow$ `PriraditHodnotuVelicinyObjektum`(m, v, o)

$\mathcal{O}' \leftarrow \mathcal{O}' \cup \{o'\}$

ukonči pro všechny

vrať: $(\mathcal{O}')$

5.3.3 Generovat příslušnost k třídě podle pravidla

Elementární operace `Generovat příslušnost k třídě podle pravidla` je operací s hodnotami veličin, která slouží ke generování příslušnosti k třídě ϵ na základě pravidla přidělování příslušnosti k třídě χ_{C} a parametru generování příslušnosti k třídě ρ_{C}

Vstupy:

- χ_{C} – pravidlo přidělování příslušnosti k třídě
- ρ_{C} – parametr generování příslušnosti k třídě

Výstupy:

- ϵ – příslušnost k třídě

Zápis:

- $(\epsilon) \leftarrow$ `GenerovatPrislusnostKTridePodlePravidla`$(\chi_{\mathrm{C}}, \rho_{\mathrm{C}})$

5.4 Vyhledávací operace

Vyhledávací operace je operací, při které dochází na základě vstupního objektu o k definovaným způsobem realizovanému vyhledávání dalších objektů stanovené třídy v rámci dané struktury nebo napříč strukturami. Hromadná vyhledávací operace je operací, při které dochází na základě vstupní množiny objektu $\mathcal{O}$ k definovaným způsobem realizovanému vyhledávání dalších objektů stanovené třídy v rámci dané struktury nebo napříč strukturami.

5.4.1 Sestavit množinu prvků incidujících s objektem

Elementární operace `Sestavit množinu prvků incidujících s objektem` je vyhledávací operací, která slouží k sestavení množiny prvků $\mathcal{A}_\Gamma$ incidujících se vstupním objektem o. Tímto objektem může být prvek (neliniový nebo liniový), konektor, pól prvku (příp. konektoru) nebo vazba. Konkrétní způsob realizace je závislý na třídě objektu o. Kromě vstupního objektu o je nutné znát (rozšířenou) strukturu s, v rámci níž mají být incidující prvky vyhledávány.

Vstupy:

- o – referenční objekt pro sestavování množiny incidujících prvků
- s – struktura (nebo rozšířená struktura), v rámci níž mají být incidujícící prvky vyhledávány

Výstupy:

- $\mathcal{A}_\Gamma$ – množina prvků incidujících s objektem o

Zápis:

- $(\mathcal{A}_\Gamma) \leftarrow$ `SestavitMnozinuPrvkuIncidujicichSObjektem`(o, s)

5.4.2 Sestavit množinu prvků incidujících s objekty

Operace `Sestavit množinu prvků inicidujících s objekty` je hromadnou vyhledávací operací, která slouží k sestavení množiny prvků $\mathcal{A}_\Gamma$ incidujících s objekty vstupní množiny $\mathcal{O}$. Objektem množiny $\mathcal{O}$ může být prvek (neliniový nebo liniový), konektor, pól prvku (příp. konektoru) nebo vazba. Kromě vstupní množiny objektů $\mathcal{O}$ je nutné znát (rozšířenou) strukturu s , v rámci níž mají být incidující prvky vyhledávány.

Vstupy:

- $\mathcal{O}$ – množina referenčních objektů pro sestavování množiny incidujících prvků
- s – struktura (nebo rozšířená struktura), v rámci níž mají být incidující prvky vyhledávány

Výstupy:

- $\mathcal{A}_\Gamma$ – množina prvků incidujících s objekty množiny $\mathcal{O}$

Zápis:

- $(\mathcal{A}_\Gamma) \leftarrow$ `SestavitMnozinuPrvkuIncidujicichSObjekty`$(\mathcal{O}, s)$

Operace : `SestavitMnozinuPrvkuIncidujicichSObjekty`$(\mathcal{O}, s)$

$\mathcal{A}_\Gamma \leftarrow \emptyset$

pro všechny $o \in \mathcal{O}$ **proveď**

$(\mathcal{A}_{A\Gamma}) \leftarrow$ `SestavitMnozinuPrvkuIncidujicichSObjektem`(o, s)

$\mathcal{A}_\Gamma \leftarrow \mathcal{A}_\Gamma \cup \mathcal{A}_{A\Gamma}$

ukonči pro všechny

vrať: $(\mathcal{A}_\Gamma)$

5.4.3 Sestavit množinu konektorů incidujících s objektem

Elementární operace `Sestavit množinu konektorů incidujících konektorů` je vyhledávací operací, která slouží k sestavení množiny konektorů $\mathcal{B}_\Gamma$ incidujících se vstupním objektem o. Tímto objektem může být prvek (neliniový nebo liniový), pól prvku nebo vazba. Konkrétní způsob realizace je závislý na třídě objektu o. Kromě vstupního objektu o je nutné znát rozšířenou strukturu $\tilde{s}$, v rámci níž mají být incidující konektory vyhledávány.

Vstupy:

- o – referenční objekt pro sestavování množiny incidujících konektorů
- $\tilde{s}$ – rozšířená struktura, v rámci níž mají být incidující konektory vyhledávány

Výstupy:

- $\mathcal{B}_\Gamma$ – množina konektorů incidujících s objektem o

Zápis:

- $(\mathcal{B}_\Gamma) \leftarrow$ `SestavitMnozinuKonektoruIncidujicichSObjektem`$(o, \tilde{s})$

5.4.4 Sestavit množinu konektorů incidujících s objekty

Operace `Sestavit množinu konektorů inicidujících s objekty` je hromadnou vyhledávací operací, která slouží k sestavení množiny konektorů $\mathcal{B}_\Gamma$ incidujících s objekty vstupní množiny $\mathcal{O}$. Objektem množiny $\mathcal{O}$ může být prvek (neliniový nebo liniový), pól prvku nebo vazba. Kromě vstupní množiny objektů $\mathcal{O}$ je nutné znát rozšířenou strukturu $\tilde{s}$, v rámci níž mají být incidující prvky vyhledávány.

Vstupy:

- $\mathcal{O}$ – množina referenčních objektů pro sestavování množiny incidujících konektorů
- $\tilde{s}$ – rozšířená struktura, v rámci níž mají být incidující konektory vyhledávány

Výstupy:

- $\mathcal{B}_\Gamma$ – množina konektorů incidujících s objekty množiny $\mathcal{O}$

Zápis:

- $(\mathcal{B}_\Gamma) \leftarrow$ `SestavitMnozinuKonektoruIncidujicichSObjekty`$(\mathcal{O}, s)$

Operace : `SestavitMnozinuKonektoruIncidujicichSObjekty`$(\mathcal{O}, s)$

$\mathcal{B}_\Gamma \leftarrow \emptyset$

pro všechny $o \in \mathcal{O}$ **proveď**

$(\mathcal{B}_{A\Gamma}) \leftarrow$ `SestavitMnozinuKonektoruIncidujicichSObjektem`(o, s)

$\mathcal{B}_\Gamma \leftarrow \mathcal{B}_\Gamma \cup \mathcal{B}_{A\Gamma}$

ukonči pro všechny

vrať: $(\mathcal{B}_\Gamma)$

5.4.5 Sestavit množinu pólů prvku incidujících s objektem

Elementární operace `Sestavit množinu pólů prvků incidujících s objektem` je vyhledávací operací, která slouží k sestavení množiny pólů prvků $\mathcal{Q}_\Gamma$ incidujících se vstupním objektem o. Tímto objektem může být prvek (neliniový nebo liniový), konektor nebo vazba. Konkrétní způsob realizace je závislý na třídě objektu o. Kromě vstupního objektu o je nutné znát (rozšířenou) strukturu s, v rámci níž mají být incidující póly prvků vyhledávány.

Vstupy:

- o – referenční objekt pro sestavování množiny incidujících pólů prvků
- s – struktura (nebo rozšířená struktura), v rámci níž mají být incidující póly prvků vyhledávány

Výstupy:

- $\mathcal{Q}_\Gamma$ – množina pólů prvků incidujících s objektem o

Zápis:

- $(\mathcal{Q}_\Gamma) \leftarrow$ `SestavitMnozinuPoluIncidujicichSObjektem`(o, s)

5.4.6 Sestavit množinu pólů prvků incidujících s objekty

Operace `Sestavit množinu pólů prvků inicidujících s objekty` je hromadnou vyhledávací operací, která slouží k sestavení množiny pólů prvků $\mathcal{Q}_\Gamma$ incidujících s objekty vstupní množiny $\mathcal{O}$. Objektem množiny $\mathcal{O}$ může být prvek (neliniový nebo liniový), konektor nebo vazba. Kromě vstupní množiny objektů $\mathcal{O}$ je nutné znát (rozšířenou) strukturu s , v rámci níž mají být incidující póly prvků vyhledávány.

Vstupy:

- $\mathcal{O}$ – množina referenčních objektů pro sestavování množiny incidujících pólů prvků
- s – struktura (nebo rozšířená struktura), v rámci níž mají být incidující póly prvků vyhledávány

Výstupy:

- $\mathcal{Q}_\Gamma$ – množina pólů prvků incidujících s objekty množiny $\mathcal{O}$

Zápis:

- $(\mathcal{Q}_\Gamma) \leftarrow$ `SestavitMnozinuPoluIncidujicichSObjekty`$(\mathcal{O}, s)$

Operace : `SestavitMnozinuPoluIncidujicichSObjekty`$(\mathcal{O}, s)$

$\mathcal{Q}_\Gamma \leftarrow \emptyset$

pro všechny $o \in \mathcal{O}$ **proveď**

$(\mathcal{Q}_{A\Gamma}) \leftarrow$ `SestavitMnozinuPoluIncidujicichSObjektem`(o, s)

$\mathcal{Q}_\Gamma \leftarrow \mathcal{Q}_\Gamma \cup \mathcal{Q}_{A\Gamma}$

ukonči pro všechny

vrať: $(\mathcal{Q}_\Gamma)$

5.4.7 Sestavit množinu vazeb incidujících s objektem

Elementární operace `Sestavit množinu vazeb incidujících s objektem` je vyhledávací operací, která slouží k sestavení množiny vazeb $\mathcal{R}_\Gamma$ incidujících se vstupním objektem o. Tímto objektem může být prvek (neliniový nebo liniový), konektor nebo pól prvku (příp. konektoru). Konkrétní způsob realizace je závislý na třídě objektu o. Kromě vstupního objektu o je nutné znát (rozšířenou) strukturu s, v rámci níž mají být incidující prvky vyhledávány.

Vstupy:

- o – referenční objekt pro sestavování množiny incidujících vazeb
- s – struktura (nebo rozšířená struktura), v rámci níž mají být incidující vazby vyhledávány

Výstupy:

- $\mathcal{R}_\Gamma$ – množina vazeb incidujících s objektem o

Zápis:

- $(\mathcal{R}_\Gamma) \leftarrow$ `SestavitMnozinuVazebIncidujicichSObjektem`(o, s)

5.4.8 Sestavit množinu vazeb incidujících s objekty

Operace `Sestavit množinu vazeb inicidujících s objekty` je hromadnou vyhledávací operací, která slouží k sestavení množiny vazeb $\mathcal{R}_\Gamma$ incidujících s objekty vstupní množiny $\mathcal{O}$. Objektem množiny $\mathcal{O}$ může být prvek (neliniový nebo liniový), konektor nebo pól prvku (příp. konektoru). Kromě vstupní množiny objektů $\mathcal{O}$ je nutné znát (rozšířenou) strukturu s , v rámci níž mají být incidující vazby vyhledávány.

Vstupy:

- $\mathcal{O}$ – množina referenčních objektů pro sestavování množiny incidujících vazeb
- s – struktura (nebo rozšířená struktura), v rámci níž mají být incidující vazby vyhledávány

Výstupy:

- $\mathcal{R}_\Gamma$ – množina vazeb incidujících s objekty množiny $\mathcal{O}$

Zápis:

- $(\mathcal{R}_\Gamma) \leftarrow$ `SestavitMnozinuVazebIncidujicichSObjekty`$(\mathcal{O}, s)$

Operace : `SestavitMnozinuVazebIncidujicichSObjekty`$(\mathcal{O}, s)$

$\mathcal{R}_\Gamma \leftarrow \emptyset$

pro všechny $o \in \mathcal{O}$ **proveď**

$(\mathcal{R}_{A\Gamma}) \leftarrow$ `SestavitMnozinuVazebIncidujicichSObjektem`(o, s)

$\mathcal{R}_\Gamma \leftarrow \mathcal{R}_\Gamma \cup \mathcal{R}_{A\Gamma}$

ukonči pro všechny

vrať: $(\mathcal{R}_\Gamma)$

5.4.9 Sestavit množinu kompozičních prvků

Elementární operace `Sestavit množinu kompozičních prvků` je vyhledávací operací, která slouží k sestavení množiny $\mathcal{A}_\Omega$, jejímiž prvky jsou prvky, pro každý z nichž platí, že vůči některé z takových kolekcí množiny $\mathcal{C}_\mathrm{S}$, vůči níž vystupuje vstupní prvek a v roli části, vystupuje v roli celku.

Vstupy:

- a – referenční prvek pro sestavování množiny kompozičních prvků
- $\mathcal{C}_\mathrm{S}$ – množina kolekcí v rámci nichž mají být kompoziční prvky vyhledávány

Výstupy:

- $\mathcal{A}_\Omega$ – množina kompozičních prvků

Zápis:

- $(\mathcal{A}_\Omega) \leftarrow \texttt{SestavitMnozinuKompozicnichPrvku}(a, \mathcal{C}_\mathrm{S})$

5.4.10 Sestavit množinu komponentních prvků

Elementární operace `Sestavit množinu komponentních prvků` je vyhledávací operací, která slouží k sestavení množiny, jejímiž prvky jsou množiny $(\mathcal{A}_\omega)_i$, pro každou z nichž platí, že vůči některé z takových kolekcí množiny $\mathcal{C}_\mathrm{S}$, vůči níž vystupuje vstupní prvek a v roli celku, její prvky vystupují v roli částí.

Vstupy:

- a – referenční prvek pro sestavování množiny množin komponentních prvků
- $\mathcal{C}_\mathrm{S}$ – množina kolekcí v rámci nichž mají být množiny komponentních prvků vyhledávány

Výstupy:

- $\left((\mathcal{A}_\omega)_1, (\mathcal{A}_\omega)_2, \ldots (\mathcal{A}_\omega)_k\right)$ – množina množin komponentních prvků

Zápis:

- $(\mathcal{A}_\Omega) \leftarrow \texttt{SestavitMnozinuKompozicnichPrvku}(a, \mathcal{C}_\mathrm{S})$

5.5 Operace za spoluúčasti zpracovatele

Operace za spoluúčasti zpracovatele je operací, při je nutná spoluúčast zpracovatele, od kterého je vyžadováno provedení definované akce.

5.5.1 Vybrat vyhovující objekt zpracovatelem

Elementární operace `Vybrat vyhovující objekt zpracovatelem` je operací za spoluúčasti zpracovatele, při které zpracovatel z množiny objektů $\mathcal{O}$ vybírá jeden konkrétní objekt o.

Vstupy:

- $\mathcal{O}$ – množina objektů, ze které zpracovatel vybírá objekt o

Výstupy:

- o – objekt vybraný zpracovatelem z množiny $\mathcal{O}$

Zápis:

- $(o) \leftarrow$ `VybratVyhovujiciObjektZpracovatelem`$(\mathcal{O})$

5.5.2 Vytvořit nový vzor vnitřní struktury zpracovatelem

Elementární operace `Vytvořit nový vzor vnitřní struktury zpracovatelem` je operací za spoluúčasti zpracovatele, při které zpracovatel vytváří nový vzor (rozšířené) vnitřní struktury ${}^{*}\tilde{s}^{\mathrm{F}}_{\mathrm{R}}$ pro rozlišovací úroveň l^{F}, který může reprezentovat (rozšířenou) vnitřní strukturu některých prvků figurujících na rozlišovací úrovni l^{R}. Současně vzorům na základě těchto rozlišovacích úrovní požaduje přiřazuje požadované atributy.

Vstupy:

- l^{F} – podrobnější rozlišovací úroveň, na níž má být podle vzoru $\tilde{s}^{\mathrm{F}}_{\mathrm{G}}$ vytvářena (rozšířená) struktura
- l^{G} – méně podrobná rozlišovací úroveň, na níž figurují prvky, na jejichž dezintegraci může být vzor $\tilde{s}^{\mathrm{F}}_{\mathrm{G}}$ aplikován

Výstupy:

- ${}^{*}\tilde{s}^{\mathrm{F}}_{\mathrm{G}}$ – nově vytvořený vzor (rozšířené) vnitřní struktury

Zápis:

- $({}^{*}\tilde{s}^{\mathrm{F}}_{\mathrm{G}}) \leftarrow$ `VytvoritNovyVzorVnitrniStrukturyZpracovatelem`$(l^{\mathrm{F}}, l^{\mathrm{G}})$

5.6 Ověřovací operace

Ověřovací operace je operací, při které dochází k ověření splnění vstupního kritéria nebo podmínky regularity aplikované na vstupní objekt nebo objekty. Splnění kritéria nebo podmínky regularity je indikováno návratovou hodnou, která je `1`, pokud došlo ke splnění podmínky nebo kritéria, v opačném případě je tato hodnota `0`.

5.6.1 Ověřit splnění kritéria

Elementární operace `Ověřit splnění kritéria` je ověřovací operací, která slouží k vyhodnocení splnění kritéria ψ aplikovaného na objekt o. Splnění tohoto kritéria je indikováno návratovou hodnotou Ψ.

Vstupy:

- o – objekt, na nějž je aplikováno kritérium ψ
- ψ – kritérium

Výstupy:

- Ψ – návratová hodnota kritéria ψ pro objekt o

Zápis:

- $(\Psi) \leftarrow$ `OveritSplneniKriteria`(o, ψ)

5.6.2 Ověřit splnění podmínky regularity

Elementární operace `Ověřit splnění podmínky regularity` je ověřovací operací, která slouží k vyhodnocení splnění podmínky regularity φ aplikované na objekty o_A a o_B. Splnění této podmínky regularity je indikováno návratovou hodnotou Φ.

Vstupy:

- o_A – první z objektů, na rozhraní mezi nimiž má být posuzována regularita podle podmínky regularity φ
- o_B – druhý z objektů, na rozhraní mezi nimiž má být posuzována regularita podle podmínky regularity φ
- φ – podmínka regularity

Výstupy:

- Φ – návratová hodnota podmínky regularity na rozhraní mezi objekty o_A a o_B

Zápis:

- $(\Phi) \leftarrow$ `OveritSplneniPodminkyRegularity`$(o_\mathrm{A}, o_\mathrm{B}, \varphi)$

5.7 Klasifikační operace

Klasifikační operace je operací, při které dochází k rozdělování vstupní množiny objektů do výstupních množin na základě definovaného klasifikačního kritéria. Klasifikace prvků systému $\mathbb{T}$ může být podkladem pro dekompozici tohoto systému.

5.7.1 Binárně klasifikovat prvky

Operace `Binárně klasifikovat prvky` je klasifikační operací, která slouží k rozdělení vstupní množiny prvků $\mathcal{A}$ na dvě disjunktní podmnožiny $\mathcal{A}_{O+}$ a $\mathcal{A}_{O-}$ na základě klasifikačního kritéria ψ.

Vstupy:

- $\mathcal{A}$ – množina prvků určená k rozdělení
- ψ – binární klasifikační kritérium

Výstupy:

- $\mathcal{A}_{O+}$ – množina prvků vyhovujících binárnímu klasifikačnímu kritériu ψ
- $\mathcal{A}_{O-}$ – množina prvků nevyhovujících binárnímu klasifikačnímu kritériu ψ

Zápis:

- $(\mathcal{A}_{O+}, \mathcal{A}_{O-}) \leftarrow$ `BinarneKlasifikovatPrvky`$(\mathcal{A}, \psi)$

Operace : `BinarneKlasifikovatPrvky`$(\mathcal{A}, \psi)$

$\mathcal{A}_{O+} \leftarrow \emptyset$

$\mathcal{A}_{O-} \leftarrow \emptyset$

pro všechny $a \in \mathcal{A}$ **proveď**

$(\Psi) \leftarrow$ `OveritSplneniKriteria`(a, ψ)

jestliže $\Psi = 1$ **proveď**

$\mathcal{A}_{O+} \leftarrow \mathcal{A}_{O+} \cup \{a\}$

jinak

$\mathcal{A}_{O-} \leftarrow \mathcal{A}_{O-} \cup \{a\}$

ukonči jestliže

ukonči pro všechny

vrať: $(\mathcal{A}_{O+}, \mathcal{A}_{O-})$

5.7.2 Klasifikovat prvky podle komponent souvislosti

Operace `Klasifikovat prvky podle komponent souvislosti` je klasifikační operací, která slouží k rozdělení množiny prvků $\mathcal{A}$, která je součástí vstupní struktury $s = (\mathcal{A}, \mathcal{R})$ podle komponent souvislosti grafu, jehož vrcholy reprezentují prvky množiny $\mathcal{A}$ a jehož hrany reprezentují hrany množiny $\mathcal{R}$. Operace vrací množinu množin prvků $\mathcal{A}_i$, $i \in \{1, 2, \ldots k\}$, do nichž jsou prvky množiny $\mathcal{A}$ rozděleny podle komponent souvislosti tohoto grafu o celkovém počtu k.

Vstupy:

- s – struktura, jejíž prvky jsou určeny k rozdělení

Výstupy:

- $\{\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_k\}$ – množina množin prvků rozdělených podle komponent souvislosti

Zápis:

- $(\{\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_k\}) \leftarrow$ `KlasifikovatPrvkyPodleKomponentSouvislosti`(s)

Operace : `KlasifikovatPrvkyPodleKomponentSouvislosti`(s)

$(\mathcal{A}, \mathcal{R}) \leftarrow s$

$\mathcal{A}_\Gamma \leftarrow \emptyset$

$i \leftarrow 0$

dokud není $\mathcal{A} = \emptyset$ **prováděj**

$i \leftarrow i + 1$

$\mathcal{A}_i \leftarrow \emptyset$

$a_\mathrm{A} \leftarrow a \in \mathcal{A}$

opakuj

jestliže $\mathcal{A}_i \neq \emptyset$ **proveď**

$a_\mathrm{A} \leftarrow a \in \mathcal{A}_\Gamma$

$\mathcal{A}_\Gamma \leftarrow \mathcal{A}_\Gamma \setminus \{a_\mathrm{A}\}$

ukonči jestliže

$\mathcal{A}_i \leftarrow \mathcal{A}_i \cup \{a_\mathrm{A}\}$

$\mathcal{A} \leftarrow \mathcal{A} \setminus \{a_\mathrm{A}\}$

$(\mathcal{A}_{\mathrm{A}\Gamma}) \leftarrow$ `SestavitMnozinuPrvkuIncidujicichSObjektem`(a_{A}, s)

$\mathcal{A}_{\Gamma} \leftarrow \mathcal{A}_{\Gamma} \cup \mathcal{A}_{\mathrm{A}\Gamma}$

dokud není $\mathcal{A}_{\Gamma} = \emptyset$

ukonči dokud není

$k \leftarrow i$

vrať: $(\{\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_k\})$

5.7.3 Sestavit množinu přípustných vzorů vnitřní struktury

Operace `Sestavit množinu přípustných vzorů vnitřní struktury` je klasifikační operací, která slouží k výběru přípustných vzorů (rozšířené) vnitřní struktury ${}^{*}\tilde{\mathcal{S}}_{\mathrm{O}+}$ z množiny vzorů (rozšířené) vnitřní struktury ${}^{*}\tilde{\mathcal{S}}$, které v souladu s podmínkou regularity φ mohou představovat vnitřní strukturu prvku a.

Vstupy:

- a – prvek, vůči němuž je regularita vzorů (rozšířené) vnitřní struktury posuzována
- ${}^{*}\tilde{\mathcal{S}}$ – množina vzorů (rozšířené) vnitřní struktury, z nichž mají být vybrány ty přípustné
- φ – podmínka regularity

Výstupy:

- ${}^{*}\tilde{\mathcal{S}}_{\mathrm{O}+}$ – množina vzorů (rozšířené) vnitřní struktury, které vyhovují podmínce regularity φ

Zápis:

- $\left({}^{*}\tilde{\mathcal{S}}_{\mathrm{O}+}\right) \leftarrow$ `SestavitMnozinuPripustnychVzoruVnitrniStruktury`$\left(a, {}^{*}\tilde{\mathcal{S}},\right)$

Operace : `SestavitMnozinuPripustnychVzoruVnitrniStruktury`$(a, {}^{*}\mathcal{S}, \varphi)$

${}^{*}\tilde{\mathcal{S}}_{\mathrm{O}+} \leftarrow \emptyset$

pro všechny ${}^{*}\tilde{s} \in {}^{*}\tilde{\mathcal{S}}$ **proveď**

$(\Phi) \leftarrow$ `OveritSplneniPodminkyRegularity`$(a, {}^{*}\tilde{s}, \varphi)$

jestliže $\Phi = 1$ **proveď**

${}^{*}\tilde{\mathcal{S}}_{\mathrm{O}+} \leftarrow {}^{*}\tilde{\mathcal{S}}_{\mathrm{O}+} \cup \{{}^{*}\tilde{s}\}$

ukonči jestliže

ukonči pro všechny

vrať: $\left({}^{*}\tilde{\mathcal{S}}_{\mathrm{O}+}\right)$

5.8 Přiřazovací operace

Přiřazovací operace je operací, při které dochází k přiřazování objektů jedné množiny objektům druhé množiny.

5.8.1 Sestavit množinu přiřazení konektorů

Operace `Sestavit množinu přiřazení konektorů` je přiřazovací operací, která slouží k vytvoření množiny uspořádaných dvojic konektorů $\mathcal{P}$, jejíž jednotlivé uspořádané dvojice jsou vytvářeny na základě postupného přiřazování konektorů množiny $\mathcal{B}_{\mathrm{B}}$ konektorům množiny $\mathcal{B}_{\mathrm{A}}$. Přiřazování probíhá v souladu s definovanou podmínkou φ_{S}. V případě, že je v souladu s podmínkou φ_{S} možné konektoru $b_{\mathrm{A}} \in \mathcal{B}_{\mathrm{A}}$ přiřadit větší počet doposud nepřiřazených konektorů z množiny $\mathcal{B}_{\mathrm{A}}$, rozhodne o výsledném přiřazení uživatel.

Vstupy:

- $\mathcal{B}_{\mathrm{A}}$ – množina konektorů, jimž jsou přiřazovány konektory z množiny $\mathcal{B}_{\mathrm{B}}$
- $\mathcal{B}_{\mathrm{B}}$ – množina konektorů, které jsou přiřazovány konektorům z množiny $\mathcal{B}_{\mathrm{A}}$
- φ_{S} – podmínka propojitelnosti

Výstupy:

- $\mathcal{P}$ – množina uspořádaných dvojic konektorů, která vyjadřuje přiřazení konektoru z množiny $\mathcal{B}_{\mathrm{B}}$ konektoru z množiny $\mathcal{B}_{\mathrm{A}}$

Zápis:

- $(\mathcal{P}) \leftarrow$ `SestavitMnozinuPrirazeniKonektoru`$(\mathcal{B}_{\mathrm{A}}, \mathcal{B}_{\mathrm{B}}, \varphi_{\mathrm{S}})$

Operace : `SestavitMnozinuPrirazeniKonektoru`$(\mathcal{B}_{\mathrm{A}}, \mathcal{B}_{\mathrm{B}}, \varphi_{\mathrm{S}})$

$\mathcal{P} \leftarrow \emptyset$

dokud není $\mathcal{B}_{\mathrm{A}} = \emptyset$ **nebo** $\mathcal{B}_{\mathrm{B}} = \emptyset$ **prováděj**

$\mathcal{B}_{\mathrm{C}} \leftarrow \emptyset$

$b_{\mathrm{A}} \leftarrow b \in \mathcal{B}_{\mathrm{A}}$

pro všechny $b_{\mathrm{B}} \in \mathcal{B}_{\mathrm{B}}$ **proveď**

$(\Phi) \leftarrow$ `OveritSplneniPodminkyRegularity`$(b_{\mathrm{A}}, b_{\mathrm{B}}, \varphi_{\mathrm{S}})$

jestliže $\Phi = 1$ **proveď**

$\mathcal{B}_{\mathrm{C}} \leftarrow \mathcal{B}_{\mathrm{C}} \cup \{b_{\mathrm{B}}\}$

ukonči jestliže

ukonči pro všechny

$(b_{\mathrm{C}}) \leftarrow$ `VybratVyhovujiciObjektZpracovatelem`$(\mathcal{B}_{\mathrm{C}})$

$p \leftarrow (b_{\mathrm{A}}, b_{\mathrm{C}})$

$\mathcal{P} \leftarrow \mathcal{P} \cup \{p\}$

$\mathcal{B}_{\mathrm{A}} \leftarrow \mathcal{B}_{\mathrm{A}} \setminus \{b_{\mathrm{A}}\}$

$\mathcal{B}_{\mathrm{B}} \leftarrow \mathcal{B}_{\mathrm{B}} \setminus \{b_{\mathrm{C}}\}$

ukonči dokud není

vrať: $(\mathcal{P})$

5.9 Integrační operace

Integrační operace je transformační operací aplikovanou na množinu prvků $\mathcal{A}_i^{\mathrm{R}} \subset \mathcal{A}^{\mathrm{R}}$ nebo na uspořádanou k-tici těchto prvků $^{\circ}\mathcal{A}_i^{\mathrm{R}}$, na jejímž základě je vytvořen prvek a_i^{G}. Během integrační operace je vytvořena také uspořádaná kolekce c_i^{U} nebo neuspořádaná kolekce c_i^{O}, která vyjadřuje, že prvky množiny $\mathcal{A}_i^{\mathrm{R}}$ nebo uspořádané k-tice $^{\circ}\mathcal{A}_i^{\mathrm{R}}$ jsou částmi nově vzniklého prvku a_i^{G}. V závislosti na třídě prvku a_i^{R} vytvořeného integrací lze hovořit o integraci do neliniového prvku nebo o integraci do liniového prvku.

5.9.1 Neuspořádaně integrovat prvky

Operace `Neuspořádaně integrovat prvky` je integrační operací, která slouží k vytvoření prvku a_i^{G} třídy ϵ_{A} sítě n figurujícího na méně podrobné rozlišovací úrovni l^{G} a neuspořádané kolekce c_i^{U}, vůči níž vystupuje prvek a^{G} v roli celku a prvky množiny $\mathcal{A}_i^{\mathrm{R}}$, jejichž integrací prvek a_i^{G} vzniká, v roli částí.

Vstupy:

- $\mathcal{A}_i^{\mathrm{R}}$ – množina prvků figurujících na referenční rozlišovací úrovni určených k integraci v prvek a_i^{G}
- ϵ_{A} – příslušnost prvku a_i^{G} k třídě
- n – síť, ve které má být vytvořen prvek a_i^{G}
- l^{G} – méně podrobná rozlišovací úroveň, na níž má být vytvořen prvek a_i^{G}

Výstupy:

- a_i^{G} – nově vytvořený prvek vzniklý integrací prvků množiny $\mathcal{A}_i^{\mathrm{R}}$
- c_i^{U} – neuspořádaná kolekce, vůči niž vystupuje v roli celku prvek a_i^{G} a v roli částí prvky množiny $\mathcal{A}_i^{\mathrm{R}}$

Zápis:

- $(a_i^{\mathrm{G}}, c_i^{\mathrm{U}}) \leftarrow \texttt{NeusporadaneIntegrovatPrvky}(\mathcal{A}_i^{\mathrm{R}}, \epsilon_{\mathrm{A}}, n, l^{\mathrm{G}})$

Operace : $\texttt{NeusporadaneIntegrovatPrvky}(\mathcal{A}_i^{\mathrm{R}}, \epsilon_{\mathrm{A}}, n, l^{\mathrm{G}})$

---

vyber případ ϵ_{A}

případ `N`

$(a_i^{\mathrm{G}}) \leftarrow \texttt{VytvoritNeliniovyPrvek}(l^{\mathrm{G}}, n)$

případ `L`

$(a_i{}^{\mathrm{G}}) \leftarrow \texttt{VytvoritLiniovyPrvek}(l^{\mathrm{G}}, n)$

ukonči vyber

$(c_i^{\mathrm{U}}) \leftarrow \texttt{VytvoritNeusporadanouKolekci}(a_i^{\mathrm{G}}, \mathcal{A}_i^{\mathrm{R}})$

vrať: $(a_i^{\mathrm{G}}, c_i^{\mathrm{U}})$

5.9.2 Uspořádaně integrovat prvky

Operace `Uspořádaně integrovat prvky` je integrační operací, která slouží k vytvoření prvku a_i^{G} třídy ϵ_{A} sítě n figurujícího na méně podrobné rozlišovací úrovni a uspořádané kolekce c_i^{O}, vůči níž vystupuje prvek a_i^{G} v roli celku a prvky uspořádané k-tice $^{\circ}\mathcal{A}_i^{\mathrm{R}}$, jejichž integrací prvek a_i^{G} vzniká, v roli částí.

Vstupy:

- $^{\circ}\mathcal{A}_i^{\mathrm{R}}$ – uspořádaná k-tice prvků figurujících na referenční rozlišovací úrovni určených k integraci v prvek a_i^{G}
- ϵ_{A} – příslušnost prvku a_i^{G} k třídě
- n – síť, ve které má být vytvořen prvek a_i^{G}
- l^{G} – méně podrobná rozlišovací úroveň, na níž má být vytvořen prvek a_i^{G}

Výstupy:

- a_i^{G} – nově vytvořený prvek vzniklý integrací prvků množiny $\mathcal{A}_i^{\mathrm{R}}$
- c_i^{O} – uspořádaná kolekce, vůči niž vystupuje v roli celku prvek a_i^{G} a v roli částí prvky uspořádané k-tice $^{\circ}\mathcal{A}_i^{\mathrm{R}}$

Zápis:

- $(a_i^{\mathrm{G}}, c_i^{\mathrm{O}}) \leftarrow \texttt{UsporadaneIntegrovatPrvky}(\mathring{\mathcal{A}}_i^{\mathrm{R}}, \epsilon_{\mathrm{A}}, n, l^{\mathrm{G}})$

Operace : $\texttt{UsporadaneIntegrovatPrvky}(\mathring{\mathcal{A}}_i^{\mathrm{R}}, \epsilon_{\mathrm{A}}, n, l^{\mathrm{G}})$

---

vyber případ ϵ_{A}

případ `N`

$(a_i^{\mathrm{G}}) \leftarrow \texttt{VytvoritNeliniovyPrvek}(l^{\mathrm{G}}, n)$

případ `L`

$(a_i^{\mathrm{G}}) \leftarrow \texttt{VytvoritLiniovyPrvek}(l^{\mathrm{G}}, n)$

ukonči vyber

$(c_i^{\mathrm{O}}) \leftarrow \texttt{VytvoritUsporadanouKolekci}(a_i^{\mathrm{G}}, \mathring{\mathcal{A}}_i^{\mathrm{R}})$

vrať: $(a_i^{\mathrm{G}}, c_i^{\mathrm{O}})$

5.10 Dezintegrační operace

Dezintegrační operace je transformační operací aplikovanou prvek a_i^{R}, na jehož základě je vytvořena substruktura s_i^{F}, která odpovídá vnitřní struktuře prvku a_i^{R} vyjádřené rozlišovací úrovni l^{F}, nebo odpovídající rozšířená substruktura $\tilde{s}_i^{\mathrm{F}}$. Během dezintegrační operace je vytvořena také uspořádaná kolekce c_i^{U} nebo neuspořádaná kolekce c_i^{O}, která vyjadřuje, že nově vytvořené prvky množiny $\mathcal{A}_i^{\mathrm{F}}$ jsou částmi prvku a_i^{R}. V závislosti na způsobu, kterým probíhá nalezení vnitřní struktury prvku a_i^{R}, je možné rozlišovat mezi dezintegrací prvku a_i^{R} přiřazením vnitřní struktury a dezintegrací prvku a^{R} generováním vnitřní struktury. Mezi speciální případy dezintegrace prvku a_i^{R} generováním vnitřní struktury patří sériová dezintegrace prvku a_i^{R} a paralelní dezintegraci prvku a_i^{R}. V závislosti na třídě dezintegrovaného prvku a_i^{R} lze hovořit o dezintegraci neliniového prvku nebo o dezintegraci liniového prvku.

Obecně je možné za typičtější pokládat dezintegraci neliniového prvku, protože u neliniového prvku je možné spíše předpokládat, že bude možné na některé z podrobnějších rozlišovacích úrovní nalézt jeho vnitřní strukturu. Sériovou a paralelní dezintegraci je nicméně možné považovat za typičtější pro liniové prvky. Je možné ji aplikovat zejména na takové prvky, které reprezentují reálné objekty železniční infrastruktury liniového charakteru, tedy např. prvky typu kolej nebo traťový úsek.

5.10.1 Dezintegrovat prvek přiřazením struktury

Operace `Dezintegrovat prvek přiřazením struktury` je dezintegrační operací, která slouží k vytvoření (rozšířené) substruktury $\tilde{s}_i^{\mathrm{F}}$ sítě n vyjádřené na podrobnější rozlišovací úrovni l^{F}, a to na základě nalezené (rozšířené) vnitřní struktury prvku a_i^{R} figurujícího na referenční rozlišovací úrovni l^{R} určeného k dezintegraci, a neuspořádané kolekce c_i^{U}, která vyjadřuje, že je prvek a_i^{R} složen z prvků (rozšířené) substruktury s_i^{F}.

Výběr vhodných vzorů (rozšířených) vnitřních struktur probíhá z množiny vzorů (rozšířených) vnitřních struktur ${}^{*}\tilde{\mathcal{S}}^{\mathrm{F}}_{\mathrm{R}}$ na základě podmínky topologické regularity φ_{T} a podmínky parametrické regularity φ_{P} vůči prvku a_i^{R}. Výslednou volbu vyhovujícího vzoru (rozšířené) vnitřní struktury provede uživatel z omezeného výběru topologicky i parametricky regulárních vzorů (rozšířených) vnitřních struktur. V případě, že žádný topologicky i parametricky regulární vzor (rozšířené) vnitřní struktury struktury neexistuje nebo není dle úsudku uživatele vyhovující, vytvoří uživatel nový (rozšířený) vzor vnitřní struktury (pokud možno se záměrem, aby již byl pro daný účel dezintegrace prvku a_i^{R} po všech stránkách vyhovující), který je zařazen do množiny (rozšířených) vzorů vnitřních struktur ${}^{*}\tilde{\mathcal{S}}^{\mathrm{F}}_{\mathrm{R}}$ a také podroben posouzení topologické a parametrické regularity, což se opakuje až do nalezení po všech stránkách vyhovujícího vzoru (rozšířené) vnitřní struktury ${}^{*}\tilde{s}_i^{\mathrm{F}}$, na jehož základě je (rozšířená) substruktura $\tilde{s}_i^{\mathrm{F}}$ vytvořena.

Vstupy:

- a_i^{R} – prvek figurující na referenční rozlišovací úrovni určený k dezintegraci v (rozšířenou) substrukturu $\tilde{s}_i^{\mathrm{F}}$
- ${}^{*}\tilde{\mathcal{S}}^{\mathrm{F}}_{\mathrm{R}}$ – množina vzorů (rozšířených) vnitřních struktur
- n – síť, ve které má být vytvořena (rozšířená) substruktura $\tilde{s}_i^{\mathrm{F}}$
- l^{F} – podrobnější rozlišovací úroveň, na níž má být vytvořena (rozšířená) substruktura $\tilde{s}_i^{\mathrm{F}}$
- φ_{T} – podmínka topologické regularity
- φ_{P} – podmínka parametrické regularity

Výstupy:

- $\tilde{s}_i^{\mathrm{F}}$ – nově vytvořená (rozšířená) substruktura
- c_i^{U} – neuspořádaná kolekce, vůči niž vystupuje v roli celku prvek a_i^{R} a v roli částí prvky (rozšířené) substruktury $\tilde{s}_i^{\mathrm{F}}$

Zápis:

- $(\tilde{s}_i^{\mathrm{F}}, c_i^{\mathrm{U}}) \leftarrow \texttt{DezintegrovatPrvekPrirazenimStruktury}\left(a_i^{\mathrm{R}}, {}^{*}\tilde{\mathcal{S}}^{\mathrm{F}}_{\mathrm{R}}, l^{\mathrm{F}}, n, \varphi_{\mathrm{T}}, \varphi_{\mathrm{P}}\right)$

Operace : $\texttt{DezintegrovatPrvekPrirazenimStruktury}\left(a_i^{\mathrm{R}}, {}^{*}\tilde{\mathcal{S}}^{\mathrm{F}}_{\mathrm{R}}, l^{\mathrm{F}}, n, \varphi_{\mathrm{T}}, \varphi_{\mathrm{P}}\right)$

${}^{*}\tilde{s}_i^{\mathrm{F}} \leftarrow \texttt{Null}$

${}^{*}\tilde{\mathcal{S}}^{\mathrm{F}}_{\mathrm{S}} \leftarrow {}^{*}\tilde{\mathcal{S}}^{\mathrm{F}}_{\mathrm{R}}$

dokud není $\tilde{s}_i^{\mathrm{F}} \neq \texttt{Null}$ **prováděj**

$\left({}^{*}\tilde{\mathcal{S}}^{\mathrm{F}}_{\mathrm{T}}\right) \leftarrow \texttt{SestavitMnozinuPripustnychVzoruVnitrniStruktury}\left(a_i^{\mathrm{R}}, {}^{*}\tilde{\mathcal{S}}^{\mathrm{F}}_{\mathrm{S}}, \varphi_{\mathrm{T}}\right)$

$\left({}^{*}\tilde{\mathcal{S}}^{\mathrm{F}}_{\mathrm{P}}\right) \leftarrow \texttt{SestavitMnozinuPripustnychVzoruVnitrniStruktury}\left(a_i^{\mathrm{R}}, {}^{*}\tilde{\mathcal{S}}^{\mathrm{F}}_{\mathrm{T}}, \varphi_{\mathrm{P}}\right)$

$\left({}^{*}\tilde{s}_i^{\mathrm{F}}\right) \leftarrow \texttt{VybratVyhovujiciObjektZpracovatelem}\left(a_i^{\mathrm{R}}, {}^{*}\tilde{\mathcal{S}}^{\mathrm{F}}_{\mathrm{P}}\right)$

jestliže ${}^*\tilde{s}_i^{\mathrm{F}} = \texttt{Null}$ **proveď**

$\left({}^*\tilde{s}_u^{\mathrm{F}}\right) \leftarrow \texttt{VytvoritNovyVzorVnitrniStrukturyZpracovatelem}\left(l^{\mathrm{F}}, l^{\mathrm{R}}\right)$

${}^*\tilde{\mathcal{S}}_{\mathrm{R}}^{\mathrm{F}} \leftarrow {}^*\tilde{\mathcal{S}}_{\mathrm{R}}^{\mathrm{F}} \cup \left\{{}^*\tilde{s}_u^{\mathrm{F}}\right\}$

${}^*\tilde{\mathcal{S}}_{\mathrm{S}}^{\mathrm{F}} \leftarrow \left\{{}^*\tilde{s}_u^{\mathrm{F}}\right\}$

ukonči jestliže

ukonči dokud není

$\left(\tilde{s}_i^{\mathrm{F}}\right) \leftarrow \texttt{VytvoritSubstrukturuNaZakladeVzoru}\left({}^*\tilde{s}_i^{\mathrm{F}}, n, l^{\mathrm{F}}\right)$

$\left(\mathcal{A}_i, \tilde{\mathcal{R}}_i^{\mathrm{F}}, \mathcal{B}_i^{\mathrm{F}}\right) \leftarrow \tilde{s}_i^{\mathrm{F}}$

$\left(c_i^{\mathrm{U}}\right) \leftarrow \texttt{VytvoritNeusporadanouKolekci}\left(a_i^{\mathrm{R}}, \mathcal{A}_i^{\mathrm{F}}\right)$

vrať: $\left(\tilde{s}_i^{\mathrm{F}}, c_i^{\mathrm{U}}\right)$

5.10.2 Sériově dezintegrovat prvek

Operace `Sériově dezintegrovat prvek` je dezintegrační operací, která slouží k vytvoření rozšířené substruktury $\tilde{s}_i^{\mathrm{F}}$ sítě n vyjádřené na podrobnější rozlišovací úrovni l^{F}, a to na základě prvku a_i^{R} figurujícího na referenční rozlišovací úrovni l^{R}, a uspořádané kolekce c_i^{O}, která vyjadřuje, že je prvek a_i^{R} složen z prvků rozšířené substruktury s_i^{F} (včetně stanovení jejich pořadí). Každý prvek rozšířené substruktury s_i^{F} je vazbami propojen právě se dvěma sousedy, přičemž pro první a poslední prvek je jedním z těchto sousedů jeden ze dvou konektorů, které jsou v rámci rozšířené substruktury $\tilde{s}_i^{\mathrm{F}}$ také vytvořeny (v případě liniového prvku každý z těchto sousedů inciduje s jiným jeho pólem). Výsledný počet prvků rozšířené substruktury s_i^{F} je dán hodnotou parametru sériové dezintegrace κ_{S}, příslušnost těchto prvků ke třídě pak pravidlem přidělování příslušnosti k třídě χ_{C}.

Vstupy:

- a_i^{R} – prvek figurující na referenční rozlišovací úrovni určený k sériové dezintegraci v rozšířenou substrukturu $\tilde{s}_i^{\mathrm{F}}$
- n – síť, ve které má být vytvořena rozšířená substruktura $\tilde{s}_i^{\mathrm{F}}$
- l^{F} – podrobnější rozlišovací úroveň, na níž má být vytvořena rozšířená substruktura $\tilde{s}_i^{\mathrm{F}}$
- κ_{S} – parametr sériové dezintegrace
- χ_{C} – pravidlo přidělování příslušnosti k třídě

Výstupy:

- $\tilde{s}_i^{\mathrm{F}}$ – nově vytvořená rozšířená substruktura
- c_i^{O} – uspořádaná kolekce, vůči niž vystupuje v roli celku prvek a_i^{R} a v roli částí prvky rozšířené substruktury $\tilde{s}_i^{\mathrm{F}}$

Zápis:

- $(\tilde{s}_i^{\mathrm{F}}, c_i^{\mathrm{O}}) \leftarrow \texttt{SerioveDezintegrovatPrvek}(a_i^{\mathrm{R}}, n, l^{\mathrm{F}}, \kappa_{\mathrm{P}}, \chi_{\mathrm{C}})$

Operace : $\texttt{SerioveDezintegrovatPrvek}(a_i^{\mathrm{R}}, n, l^{\mathrm{F}}, \kappa_{\mathrm{S}}, \chi_{\mathrm{C}})$

$\mathcal{A}_i^{\mathrm{F}} \leftarrow \emptyset$

$\tilde{\mathcal{R}}_i^{\mathrm{F}} \leftarrow \emptyset$

$(b_0^{\mathrm{F}}) \leftarrow \texttt{VytvoritKonektor}(n, l^{\mathrm{F}})$

$(b_1^{\mathrm{F}}) \leftarrow \texttt{VytvoritKonektor}(n, l^{\mathrm{F}})$

$\mathcal{B}_i^{\mathrm{F}} \leftarrow \{b_0^{\mathrm{F}}\} \cup \{b_1^{\mathrm{F}}\}$

$q_{\mathrm{P}} \leftarrow q_0(b_0^{\mathrm{F}})$

$q_{\mathrm{N}} \leftarrow q_0(b_1^{\mathrm{F}})$

$j \leftarrow 0$

dokud není $j = \kappa_{\mathrm{S}}$ **prováděj**

$j \leftarrow j + 1$

$(\epsilon_{\mathrm{A}}) \leftarrow \texttt{GenerovatPrislusnostKTridePodlePravidla}(\chi_{\mathrm{C}}, j)$

vyber případ ϵ_{A}

případ `N`

$(a_j^{\mathrm{F}}) \leftarrow \texttt{VytvoritNeliniovyPrvek}(n, l^{\mathrm{F}})$

$q_{\mathrm{A}} \leftarrow q_0({a_j}^{\mathrm{F}})$

$q_{\mathrm{B}} \leftarrow q_0({a_j}^{\mathrm{F}})$

případ `L`

$(a_j^{\mathrm{F}}) \leftarrow \texttt{VytvoritLiniovyPrvek}(n, l^{\mathrm{F}})$

$q_{\mathrm{A}} \leftarrow q_0({a_j}^{\mathrm{F}})$

$q_{\mathrm{B}} \leftarrow q_1({a_j}^{\mathrm{F}})$

ukonči vyber

$\mathcal{A}_i^{\mathrm{F}} \leftarrow \mathcal{A}_i^{\mathrm{F}} \cup \{a_j^{\mathrm{F}}\}$

$\tilde{r}^{\mathrm{F}} \leftarrow$ `VytvoritVazbuMeziPolyPrvku`$(q_{\mathrm{P}}, q_{\mathrm{A}})$

$\tilde{\mathcal{R}}_i^{\mathrm{F}} \leftarrow \tilde{\mathcal{R}}_i^{\mathrm{F}} \cup \{\tilde{r}^{\mathrm{F}}\}$

$q_{\mathrm{P}} \leftarrow q_{\mathrm{B}}$

ukonči dokud není

$(\hat{r}^{\mathrm{F}}) \leftarrow$ `VytvoritVazbuMeziPolyPrvku`$(q_{\mathrm{B}}, q_{\mathrm{N}})$

$\tilde{\mathcal{R}}_i^{\mathrm{F}} \leftarrow \tilde{\mathcal{R}}_i^{\mathrm{F}} \cup \{\hat{r}^{\mathrm{F}}\}$

$\tilde{s}_i^{\mathrm{F}} \leftarrow \left(\mathcal{A}_i^{\mathrm{F}}, \tilde{\mathcal{R}}_i^{\mathrm{F}}, \mathcal{B}_i^{\mathrm{F}}\right)$

$^{\circ}\mathcal{A}_i^{\mathrm{F}} \leftarrow \left(a_1^{\mathrm{F}}, a_2^{\mathrm{F}}, \ldots a_{\kappa_{\mathrm{S}}}^{\mathrm{F}}\right)$

$(c_i^{\mathrm{O}}) \leftarrow$ `VytvoritUsporadanouKolekci`$(a_i^{\mathrm{R}}, {}^{\circ}\mathcal{A}_i^{\mathrm{F}})$

vrať: $\left(\tilde{\mathcal{S}}_{\mathrm{I}}^{\mathrm{F}}, c^{\mathrm{U}}\right)$

5.10.3 Paralelně dezintegrovat prvek

Operace `Paralelně dezintegrovat prvek` je dezintegrační operací, která slouží k vytvoření rozšířené substruktury $\tilde{s}_i^{\mathrm{F}}$ sítě n vyjádřené na podrobnější rozlišovací úrovni l^{F}, a to na základě prvku a_i^{R} figurujícího na referenční rozlišovací úrovni l^{R}, a neuspořádané kolekce c_i^{U}, která vyjadřuje, že je prvek a_i^{R} složen z prvků rozšířené substruktury s_i^{F}. Jednotlivé prvky rozšířené substruktury s_i^{F} nejsou vzájemně propojeny vazbami, ale společně s každým z nich jsou vytvořeny dva konektory a konektorové vazby, které tento prvek s každým z těchto konektorů propojují (v případě liniového prvku každý z těchto konektorů inciduje s jiným jeho pólem). Výsledný počet prvků rozšířené substruktury s_i^{F} je dán hodnotou parametru paralelní dezintegrace κ_{P}, příslušnost těchto prvků ke třídě pak pravidlem přidělování příslušnosti k třídě χ_{C}.

Vstupy:

- a_i^{R} – prvek figurující na referenční rozlišovací úrovni určený k paralelní dezintegraci v rozšířenou substrukturu $\tilde{s}_i^{\mathrm{F}}$
- n – síť, ve které má být vytvořena rozšířená substruktura $\tilde{s}_i^{\mathrm{F}}$
- l^{F} – podrobnější rozlišovací úroveň, na níž má být vytvořena rozšířená substruktura $\tilde{s}_i^{\mathrm{F}}$
- κ_{P} – parametr paralelní dezintegrace
- χ_{C} – pravidlo příslušnosti k třídě

Výstupy:

- $\tilde{s}_i^{\mathrm{F}}$ – nově vytvořená rozšířená substruktura
- c_i^{U} – neuspořádaná kolekce, vůči niž vystupuje v roli celku prvek a_i^{R} a v roli částí prvky rozšířené substruktury $\tilde{s}_i^{\mathrm{F}}$

Zápis:

- $(\tilde{s}_i^{\mathrm{F}}, c_i^{\mathrm{U}}) \leftarrow \texttt{ParalelneDezintegrovatPrvek}(a_i^{\mathrm{R}}, n, l^{\mathrm{F}}, \kappa_{\mathrm{P}}, \chi_{\mathrm{C}})$

Operace : $\texttt{ParalelneDezintegrovatPrvek}(a_i^{\mathrm{R}}, n, l^{\mathrm{F}}, \kappa_{\mathrm{P}}, \chi_{\mathrm{C}})$

$\mathcal{A}_i^{\mathrm{F}} \leftarrow \emptyset$

$\tilde{\mathcal{R}}_i^{\mathrm{F}} \leftarrow \emptyset$

$(b_0^{\mathrm{F}}) \leftarrow \texttt{VytvoritKonektor}(n, l^{\mathrm{F}})$

$(b_1^{\mathrm{F}}) \leftarrow \texttt{VytvoritKonektor}(n, l^{\mathrm{F}})$

$\mathcal{B}_i^{\mathrm{F}} \leftarrow \{b_0^{\mathrm{F}}, b_1^{\mathrm{F}}\}$

$j \leftarrow 0$

dokud není $j = \kappa_{\mathrm{P}}$ **prováděj**

$j \leftarrow j + 1$

$\epsilon_{\mathrm{A}} \leftarrow \texttt{GenerovatPrislusnostKTridePodlePravidla}(\chi_{\mathrm{C}}, j)$

vyber případ ϵ_{A}

případ `N`

$(a^{\mathrm{F}}) \leftarrow \texttt{VytvoritNeliniovyPrvek}(n, l^{\mathrm{F}})$

$(\hat{r}_0^{\mathrm{F}}) \leftarrow \texttt{VytvoritVazbuMeziPolyPrvku}(q_0\,(b_0^{\mathrm{F}}), q_0(a^{\mathrm{F}}))$

$(\hat{r}_1^{\mathrm{F}}) \leftarrow \texttt{VytvoritVazbuMeziPolyPrvku}(q_0\,(a^{\mathrm{F}}), q_0(b_1^{\mathrm{F}}))$

případ `L`

$(a^{\mathrm{F}}) \leftarrow \texttt{VytvoritLiniovyPrvek}(n, l^{\mathrm{F}})$

$(\hat{r}_0^{\mathrm{F}}) \leftarrow \texttt{VytvoritVazbuMeziPolyPrvku}(q_0\,(b_0^{\mathrm{F}}), q_0(a^{\mathrm{F}}))$

$(\hat{r}_1^{\mathrm{F}}) \leftarrow \texttt{VytvoritVazbuMeziPolyPrvku}(q_1\,(a^{\mathrm{F}}), q_0(b_1^{\mathrm{F}}))$

ukonči vyber

$\mathcal{A}_i^{\mathrm{F}} \leftarrow \mathcal{A}_i^{\mathrm{F}} \cup \{a^{\mathrm{F}}\}$

ukonči dokud není

$\tilde{\mathcal{R}}_i^{\mathrm{F}} \leftarrow \tilde{\mathcal{R}}_i^{\mathrm{F}} \cup \{\hat{r}_0^{\mathrm{F}}, \hat{r}_1^{\mathrm{F}}\}$

$\tilde{s}_i^{\mathrm{F}} \leftarrow \left(\mathcal{A}_i^{\mathrm{F}}, \tilde{\mathcal{R}}_i^{\mathrm{F}}, \mathcal{B}_i^{\mathrm{F}}\right)$

$(c_i^{\mathrm{U}}) \leftarrow \texttt{VytvoritNeusporadanouKolekci}(a_i^{\mathrm{R}}, \mathcal{A}_i^{\mathrm{F}})$

vrať: $(\tilde{s}_i^{\mathrm{F}}, c_i^{\mathrm{U}})$

Kapitola 6
Návrh integračního algoritmu

Za účelem transformovat známou referenční strukturu s^{R} sytému $\mathbb{T}$ topologického popisu sítě n, vyjádřenou na referenční rozlišovací úrovni l^{R}, na hledanou méně podrobnou strukturu s^{G} tohoto systému, vyjádřenou na méně podrobné rozlišovací úrovni l^{G}, bude pro daný aplikační případ w_{RG} navrhován integrační algoritmus t_{RG}. Návrh toho algoritmu závisí na typech $\tau_{\mathrm{S}}^{\mathrm{R}}$ a $\tau_{\mathrm{S}}^{\mathrm{G}}$ přípustných struktur vyjádřených na rozlišovacích úrovních l^{R} a l^{G}. V rámci této kapitoly bude definován rámcový postup návrhu integračního algoritmu a uveden příklad aplikace tohoto postupu pro konkrétní aplikační případ w_{RG}.

6.1 Postup při návrhu integračního algoritmu

- Před vlastním návrhem integračního algoritmu t_{RG} nejprve specifikujeme typy přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{R}}$ a $\tau_{\mathrm{S}}^{\mathrm{G}}$.
- V souladu s typy přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{R}}$ a $\tau_{\mathrm{S}}^{\mathrm{G}}$ provedeme posouzení, zda je za všech okolností možné každý v úvahu přicházející prvek $a^{\mathrm{R}} \in \mathcal{A}^{\mathrm{R}}$ jednoznačně přiřadit některému v úvahu přicházejícímu prvku $a^{\mathrm{G}} \in \mathcal{A}^{\mathrm{G}}$ v roli jeho části. Toto posouzení provádíme na obecné úrovni na základě toho, jak rozsáhlou a jakým způsobem vymezenou část železniční (resp. drážní) infrastruktury tyto prvky reprezentují.
- V případě, že je výsledek výše uvedeného posouzení pozitivní a je to účelné, navrhneme takový typ odvozené struktury $\tau_{\mathrm{S}}^{\mathrm{D}}$, pro každý jejíž v úvahu přicházející prvek $a^{\mathrm{D}} \in \mathcal{A}^{\mathrm{D}}$ platí, že je mu za všech okolností možné jednoznačně přiřadit některý v úvahu přicházející prvek $a^{\mathrm{R}} \in \mathcal{A}^{\mathrm{R}}$ v roli jeho části a zároveň jej za všech okolností jednoznačně přiřadit některému v úvahu přicházejícímu prvku $a^{\mathrm{G}} \in \mathcal{A}^{\mathrm{G}}$ v roli jeho části. V souladu s typem odvozené struktury $\tau_{\mathrm{S}}^{\mathrm{D}}$ je pak nutné zavést odvozenou rozlišovací úroveň l^{D} a navrhnout integrační algoritmy t_{RD} a t_{DG}.
- V případě, že je výsledek výše uvedeného negativní, navrhneme takový typ odvozené struktury $\tau_{\mathrm{S}}^{\mathrm{D}}$, pro každý jejíž v úvahu přicházející prvek $a^{\mathrm{D}} \in \mathcal{A}^{\mathrm{D}}$ platí, že je jej za všech okolností možné jednoznačně přiřadit některému v úvahu přicházejícímu prvku $a^{\mathrm{R}} \in \mathcal{A}^{\mathrm{R}}$ v roli jeho části a zároveň některému v úvahu přicházejícímu prvku $a^{\mathrm{G}} \in \mathcal{A}^{\mathrm{G}}$ v roli jeho části, nebo že je mu za všech okolností možné jednoznačně přiřadit některý v úvahu přicházející prvek $a^{\mathrm{R}} \in \mathcal{A}^{\mathrm{R}}$ v roli jeho části a zároveň některý v úvahu přicházející prvek $a^{\mathrm{G}} \in \mathcal{A}^{\mathrm{G}}$ v roli jeho části. V souladu s typem odvozené struktury $\tau_{\mathrm{S}}^{\mathrm{D}}$ je nutné zavést odvozenou rozlišovací úroveň l^{D} a navrhnout transformační algoritmy t_{RD} a t_{DG}, z nichž právě jeden je algoritmem integračním a právě jeden algoritmem dezintegračním.
- V souladu s typem přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{G}}$ specifikujeme, jaké různé způsoby integrace budeme na základě v úvahu přicházejících tříd prvků množiny $\mathcal{A}^{\mathrm{G}}$ a případně i dalších faktorů při návrhu integračního algoritmu t_{RG} uplatňovat.

- Navrhneme způsob klasifikace prvků množiny $\mathcal{A}^{\mathrm{R}}$, který zajistí rozděleni prvků této množiny do několika disjunktních podmnožin podle specifikovaných způsobů integrace.

- Pro každý ze specifikovaných způsobů integrace navrhneme způsob klasifikace prvků příslušné podmnožiny množiny $\mathcal{A}^{\mathrm{R}}$, který zajisti rozdělení prvků této podmnožiny do disjunktních podmnožin $\mathcal{A}_i^{\mathrm{R}}$ tak, aby platilo, že integrací prvků každé množiny $\mathcal{A}_i^{\mathrm{R}}$ získáme jeden prvek a_i^{G}, který se má stát prvkem struktury s^{G}. Integrační algoritmus pak zkonstruujme tak, aby jeho prostřednictvím bylo možné provádět integraci prvků množiny $\mathcal{A}_i^{\mathrm{R}}$ v prvek a_i^{G} byla prováděna vždy v souladu s příslušným specifikovaným způsobem.

- V souladu s typem přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{G}}$ navrhneme, jakým způsobem mají být na základě existujících vazeb mezi prvky různých množin $\mathcal{A}_i^{\mathrm{R}}$ a případně i dalších faktorů vytvářeny vazby mezi prvky množiny $\mathcal{A}^{\mathrm{G}}$, a to včetně zohlednění hodnot průchodnosti η.

- V souladu s typem přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{G}}$ navrhneme, jakým způsobem mají být na základě hodnot veličin popisujících prvky množiny $\mathcal{A}^{\mathrm{R}}$ a případně i dalších faktorů generovány hodnoty veličin popisující prvky množiny $\mathcal{A}^{\mathrm{G}}$.

- Jako vstupy integračního algoritmu t_{RG} budeme požadovat strukturu s^{R} (rozumí se včetně popisu veškerých jejích komponent hodnotami veličin v souladu s typem přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{R}}$), množinu kolekcí $\mathcal{C}$ (je-li systém $\mathbb{T}$ popsán pouze na rozlišovací úrovni l^{R}, lze předpokládat, že se na vstupu jedná o prázdnou množinu, a dále síť n a rozlišovací úroveň l^{G} (ve smyslu jejich identifikátorů), jimž mají být přiřazovány jednotlivé komponenty struktury s^{R}. Další potřebné vstupy mohou být definovány na základě konkrétní realizace návrhu algoritmu t_{RG} v závislosti na požadovaných vstupech jednotlivých operací, kterých algoritmus využívá. Může se jednat např. o klasifikační kritéria, podmínky regularity a pravidla příslušnosti k třídě.

- Jako výstupy integračního algoritmu t_{RG} budeme požadovat strukturu s^{G} (rozumí se včetně popisu veškerých jejích komponent hodnotami veličin v souladu s typem přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{G}}$), množinu kolekcí $\mathcal{C}$ (na výstupu nově obsahující i kolekce vytvořené při integraci prvků množiny $\mathcal{A}^{\mathrm{R}}$ v prvky množiny $\mathcal{A}^{\mathrm{G}}$).

6.2 Příklad návrhu integračního algoritmu

Postup při návrhu integračního algoritmu t_{RG} budeme demonstrovat na aplikačním případu w_{RG} specifikovaném následujícím způsobem: Referenční struktura s^{R} je tvořena prvky reprezentujícími uzlové objekty na úrovni kolejí a koleje tyto uzlové objekty propojující. Uzlovými objekty na úrovni kolejí přitom mohou být výhybky, kolejové křižovatky a zakončení koleje (např. opatřená zarážedly). Navrhovaný integrační algoritmus t_{RG} má zajistit transformaci referenční struktury s^{R} na méně podrobnou strukturu s^{G}, která je tvořená prvky reprezentujícími jednotlivé dopravny s kolejovým rozvětvením a traťové úseky tyto dopravny propojující. Upřesňující informace týkající se vstupních podmínek budou uvedeny při specifikaci jednotlivých typů přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{R}}$ a $\tau_{\mathrm{S}}^{\mathrm{G}}$. Bude se přitom jednat o téměř minimální požadavky nutné pro realizaci navrhovaného integračního algoritmu t_{RG}.

6.2.1 Specifikace typů přípustné struktury

Pro typ přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{R}}$ uvažujeme, že struktura tohoto typu bude různorodou strukturou, jejíž součástí mohou být jak prvky třídy neliniových prvků, tak prvky třídy liniových prvků. V rámci těchto tříd nebudeme dále výslovně rozlišovat jednotlivé funkční typy prvků. Budeme předpokládat, že neliniové prvky reprezentují obecně uzlové objekty na úrovni kolejí a liniové prvky koleje, a to ve stavebním smyslu. Budeme uvažovat pouze s průchodnými vazbami.

Neliniové prvky budou popsány pouze hodnotou identifikároru ι a příslušnosti ke třídě ϵ_{A}. Tuto skutečnost vyjádříme jako

$$\mathcal{M}\left(\mathcal{A}^{\mathrm{RN}}\right) = \{\iota, \epsilon_{\mathrm{A}}\}\,.$$

Liniové prvky budou dále popsány hodnotami veličiny stavební délka, kterou označíme jako λ, vyjadřující stavební délku koleje reprezentované příslušným prvkem a^{RL} měřenou v ose koleje v metrech. Tuto skutečnost vyjádříme jako

$$\mathcal{M}\left(\mathcal{A}^{\mathrm{RL}}\right) = \{\iota, \epsilon_{\mathrm{A}}, \lambda\}\,.$$

Objekt reálného světa bude reprezentován neliniovým prvkem, pokud se bude jednat o samostatný uzlový objekt na úrovni kolejí. Objekt reálného světa bude reprezentován liniovým prvkem, pokud se bude jednat o kolej ve stavebním smyslu uvažovanou vždy v celé její délce mezi dvěma uzlovými objekty reálného světa. Při vytváření vazeb musí být respektována následující pravidla:

- Každá vazba vyjadřuje funkční návaznost dvou sousedících objektů reálného světa reprezentovaných prvky, mezi nimiž je vytvořena. Musí být vytvořena vždy, když tato návaznost existuje, v případě liniového prvku je navázána na ten jeho pól, který odpovídá tomu konci koleje, který příslušný pól reprezentuje.
- Žádný liniový prvek nemůže být vazbou propojen s liniovým prvkem.
- Každý liniový prvek musí být vazbami propojen právě se dvěma neliniovými prvky, každá z nichž je k tomuto liniovému prvku navázána na jiném pólu.
- Neliniové prvky mohou být mezi sebou vazbami propojovány bez omezení.

Pro typ přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{G}}$ uvažujeme, že struktura tohoto typu bude různorodou strukturou, jejíž součástí mohou být jak prvky třídy neliniových prvků, tak prvky třídy liniových prvků. V rámci těchto tříd nebudeme dále výslovně rozlišovat jednotlivé funkční typy prvků. Budeme předpokládat, že neliniové prvky reprezentují obecně dopravny s kolejovým rozvětvením (připustíme, že neliniovým prvkem může být reprezentováno také zakončení traťového úseku mimo dopravnu) a liniové prvky traťové úseky propojující dopravny s kolejovým rozvětvením (případně zakončení traťového úseku mimo dopravnu). Budeme uvažovat pouze s průchodnými vazbami.

Neliniové i liniové prvky budou popsány pouze hodnotou identifikároru ι a příslušnosti ke třídě ϵ_{A}. Tuto skutečnost vyjádříme jako

$$\mathcal{M}\left(\mathcal{A}^{\mathrm{G}}\right) = \{\iota, \epsilon_{\mathrm{A}}\}\,.$$

Objekt reálného světa bude reprezentován neliniovým prvkem, pokud se bude jednat o dopravnu s kolejovým rozvětvením vymezenou krajními výhybkami, kolejovými křižovatkami a ukončeními koleje této dopravně příslušícími, případně o zakončení traťového úseku. Objekt reálného světa bude reprezentován liniovým prvkem, pokud se bude jednat o ucelený traťový úsek mezi dvěma objekty reálného světa reprezentovanými neliniovými prvky. Při vytváření vazeb musí být respektována následující pravidla:

- Každá vazba vyjadřuje funkční návaznost dvou sousedících objektů reálného světa reprezentovaných prvky, mezi nimiž je vytvořena. Musí být vytvořena vždy, když tato návaznost existuje, v případě liniového prvku je navázána na ten jeho pól, který odpovídá tomu konci traťového úseku, který příslušný pól reprezentuje.
- Žádný prvek nemůže být vazbou propojen prvkem s příslušností ke stejné třídě.
- Každý liniový prvek musí být vazbami propojen právě se dvěma neliniovými prvky, každá z nichž je k tomuto liniovému prvku navázána na jiném pólu.

6.2.2 Aplikovaný postup při návrhu integračního algoritmu

Jelikož každý v úvahu přicházející objekt reálného světa reprezentovaný prvky struktury s^{R} je vždy zcela součástí některého v úvahu přicházejícího objektu reálného světa reprezentovaného prvkem struktury s^{G}, je možné transformaci struktury s^{R} na strukturu s^{G} provádět přímo. Dopravna s kolejovým rozvětvením vždy zahrnuje některé uzlové objekty na úrovni kolejí a některé koleje ve stavebním smyslu, traťový úsek vždy zahrnuje koleje ve stavebním smyslu. Zakončení traťového úseku vždy zahrnuje zakončení koleje.

Množinu prvků referenční struktury $\mathcal{A}^{\mathrm{R}}$ nejprve podrobíme binární klasifikaci na prvky neliniové a na prvky liniové. K tomu použijeme binární klasifikační kritérium třídy ψ_{C}. Nechť je toto klasifikační kritérium aplikované na prvek a^{R} definované následujícím způsobem:

$$\psi_{\mathrm{C}}\colon \quad \epsilon_{\mathrm{A}}(a^{\mathrm{R}}) = \mathtt{N}$$

Takové prvky množiny $\mathcal{A}^{\mathrm{R}}$, pro něž je toto kritérium splněno, zařadíme do množiny $\mathcal{A}^{\mathrm{RN}}$. Takové prvky množiny $\mathcal{A}^{\mathrm{R}}$, pro něž tato podmínka splněna není, zařadíme do množiny do množiny $\mathcal{A}^{\mathrm{RL}}$.

Prvky je nutné dále rozdělit podle toho, do prvků méně podrobné struktury s^{G} jaké třídy mají být integrovány. Víme, že neliniové prvky struktury s^{R}, tedy prvky množiny $\mathcal{A}^{\mathrm{RN}}$, reprezentují uzlové objekty na úrovni kolejí. Každý z těchto prvků bude integrován do nějakého prvku reprezentujícího dopravnu s kolejovým rozvětvením nebo zakončení traťového úseku, tedy do neliniového prvku struktury s^{G}. Liniové prvky struktury s^{R}, tedy prvky množiny $\mathcal{A}^{\mathrm{RL}}$ reprezentují jednotlivé koleje. Abychom mohli rozhodnout, zda se jedná o koleje staniční nebo traťové, tedy jestli mají být integrovány do neliniových prvků struktury s^{G}, reprezentujících dopravny s kolejovým rozvětvení, nebo do liniových prvků struktury s^{G} reprezentujících traťové úseky, musíme opět provést binární klasifikaci.

Pro prvky množiny $\mathcal{A}^{\mathrm{RL}}$ nemáme definovanou veličinu, která by nám přímo sdělovala, zda příslušný liniový prvek reprezentuje staniční nebo traťovou kolej. Budeme tedy muset využít expertní znalosti o síti n a odvodit z tuto skutečnost z hodnot jiných veličin. Předpokládejme například, že o síti n víme, že existuje hodnota Λ, pro kterou platí, že stavební délka všech staničních kolejí je menší než tato hodnota a stavební délka všech traťových kolejí je větší než tato hodnota. Potom můžeme zavést binární klasifikační kritérium délky ψ_{D}, které aplikováno na prvek a^{RL} je definované následujícím způsobem:

$$\psi_{\mathrm{D}}\colon \quad \lambda(a^{\mathrm{RL}}) < \Lambda$$

Takové prvky množiny $\mathcal{A}^{\mathrm{RL}}$, pro něž je tato podmínka splněna, budou zařazeny do množiny $\mathcal{A}^{\mathrm{RL}}_{\mathrm{N}}$. Takové prvky množiny $\mathcal{A}^{\mathrm{RL}}$, pro něž tato podmínka splněna není, budou zařazeny do množiny . Do neliniových prvků struktury s^{G} budeme integrovat jak prvky množiny $\mathcal{A}^{\mathrm{RN}}$, tak prvky množiny $\mathcal{A}^{\mathrm{RL}}_{\mathrm{N}}$, tedy prvky množiny $\mathcal{A}^{\mathrm{R}}_{\mathrm{N}}$, kterou získáme jejich sjednocením.

Vycházejme z předpokladu, že oblasti jednotlivých dopraven s kolejovým rozvětvením na úrovni struktury s^{R}, ve kterých se vyskytují prvky množiny $\mathcal{A}_{\mathrm{N}}^{\mathrm{R}}$, jsou od sebe odděleny prvky množiny $\mathcal{A}_{\mathrm{L}}^{\mathrm{RL}}$. Proto definujeme strukturu $s_{\mathrm{N}}^{\mathrm{R}}$, která bude obsahovat pouze prvky množiny $\mathcal{A}_{\mathrm{N}}^{\mathrm{R}}$ a vazby, které neincidují s prvky množiny $\mathcal{A}_{\mathrm{L}}^{\mathrm{RL}}$. Prvky struktury $s_{\mathrm{N}}^{\mathrm{R}}$ klasifikujeme podle komponent souvislosti, čímž získáme k množin prvků reprezentujících objekty reálného světa nacházející se v oblastech jednotlivých dopraven, případně zakončení koleje nacházející se v oblasti zakončení traťového úseku. Na základě každé z k takto získaných množin $\left(\mathcal{A}_{\mathrm{N}}^{\mathrm{R}}\right)_i$ vytvoříme integrací jeden nový neliniový prvek a^{GN} a neuspořádanou kolekci c^{U}.

V okamžiku, kdy jsme vyčerpali všechny množiny $\left(\mathcal{A}_{\mathrm{N}}^{\mathrm{R}}\right)_i$, přistoupíme k tvorbě liniových prvků méně podrobné rozlišovací úrovně. Ty budeme vytvářet na základě prvků množiny $\mathcal{A}_{\mathrm{L}}^{\mathrm{RL}}$. Prvky množiny $\mathcal{A}_{\mathrm{L}}^{\mathrm{RL}}$, které reprezentují traťové koleje mezi jednotlivými dopravnami s kolejovým rozvětvením, chceme rozdělit do disjunktních podmnožin podle toho, které dopravny tyto koleje propojují. V tuto chvíli už můžeme využít informace obsažené v kolekcích množiny $\mathcal{C}$ a odvodit tuto skutečnost na základě prvků množiny $\mathcal{A}_{\mathrm{N}}^{\mathrm{R}}$ sousedících s prvky množiny $\mathcal{A}_{\mathrm{L}}^{\mathrm{RL}}$ ve struktuře s^{R}. Prvky množiny $\mathcal{A}_{\mathrm{L}}^{\mathrm{RL}}$ sousedící s prvky množiny $\mathcal{A}_{\mathrm{N}}^{\mathrm{R}}$ integrovanými ve stejné prvky zařadíme vždy do společné množiny $\left(\mathcal{A}_{\mathrm{L}}^{\mathrm{RL}}\right)_i$. Na základě každé z h takto získaných množiny $\left(\mathcal{A}_{\mathrm{L}}^{\mathrm{RL}}\right)_i$ vytvoříme integrací jeden nový neliniový prvek a^{GL} a neuspořádanou kolekci c^{U}. Během postupného vytváření neliniových prvků figurujících na méně podrobné rozlišovací úrovni l^{G} postupně vytváříme také vazby mezi těmito prvky (resp. jejich příslušnými póly) a již vytvořenými neliniovými prvky.

Vyčerpáním všech množin $\left(\mathcal{A}_{\mathrm{N}}^{\mathrm{R}}\right)_i$ získáváme kompletní strukturu s^{R} a rozšiřujeme množinu $\mathcal{C}$ o nové neuspořádané kolekce do výsledné podoby. Kompletní přehled požadovaných vstupů a získaných výstupů a možná podoba navrženého integračního algoritmu je uvedena níže.

Vstupy:

- s^{R} – referenční struktura
- n – síť, které mají být přiřazeny komponenty méně podrobné struktury
- l^{G} – méně podrobná rozlišovací úroveň, které mají být přiřazeny komponenty méně podrobné struktury
- $\mathcal{C}$ – počáteční množina kolekcí (typicky prázdná množina)
- ψ_{C} – klasifikační kritérium třídy
- ψ_{D} – klasifikační kritérium délky

Výstupy:

- s^{G} – méně podrobná struktura
- $\mathcal{C}$ – výsledná množina kolekcí

Zápis:

- $(s^{\mathrm{G}}, \mathcal{C}) \leftarrow \texttt{IntegracniAlgoritmusRG}(s^{\mathrm{R}}, n, l^{\mathrm{G}}, \mathcal{C}, \psi_{\mathrm{C}}, \psi_{\mathrm{D}})$

Algoritmus : $\texttt{IntegracniAlgoritmusRG}(s^{\mathrm{R}}, \mathcal{C}, n, l^{\mathrm{G}}, \psi_{\mathrm{C}}, \psi_{\mathrm{D}})$

$\mathcal{A}^{\mathrm{G}} \leftarrow \emptyset$

$\mathcal{R}^{\mathrm{G}} \leftarrow \emptyset$

$(\mathcal{A}^{\mathrm{R}}, \mathcal{R}^{\mathrm{R}}) \leftarrow s^{\mathrm{R}}$

$(\mathcal{A}^{\mathrm{RN}}, \mathcal{A}^{\mathrm{RL}}) \leftarrow \texttt{BinarneKlasifikovatPrvky}(\mathcal{A}^{\mathrm{R}}, \psi_{\mathrm{C}})$

$(\mathcal{A}^{\mathrm{RL}}_{\mathrm{N}}, \mathcal{A}^{\mathrm{RL}}_{\mathrm{L}}) \leftarrow \texttt{BinarneKlasifikovatPrvky}(\mathcal{A}^{\mathrm{RL}}, \psi_{\mathrm{D}})$

$\mathcal{A}^{\mathrm{R}}_{\mathrm{N}} = \mathcal{A}^{\mathrm{RN}} \cup \mathcal{A}^{\mathrm{RL}}_{\mathrm{N}}$

$(\mathcal{R}^{\mathrm{R}}_{\mathrm{L}\Gamma}) \leftarrow \texttt{SestavitMnozinuVazebIncidujicichSObjekty}(\mathcal{A}^{\mathrm{RL}}_{\mathrm{L}}, s^{\mathrm{R}})$

$\mathcal{R}^{\mathrm{R}}_{\mathrm{N}} = \mathcal{R}^{\mathrm{R}} \setminus \mathcal{R}^{\mathrm{R}}_{\mathrm{L}\Gamma}$

$s^{\mathrm{R}}_{\mathrm{N}} = (\mathcal{A}^{\mathrm{R}}_{\mathrm{N}}, \mathcal{R}^{\mathrm{R}}_{\mathrm{N}})$

$\left(\left\{(\mathcal{A}^{\mathrm{R}}_{\mathrm{N}})_1, \ldots (\mathcal{A}^{\mathrm{R}}_{\mathrm{N}})_k\right\}\right) \leftarrow \texttt{KlasifikovatPrvkyPodleKomponentSouvislosti}(s^{\mathrm{R}}_{\mathrm{N}})$

pro všechny $(\mathcal{A}^{\mathrm{R}}_{\mathrm{N}})_i \in \left\{(\mathcal{A}^{\mathrm{R}}_{\mathrm{N}})_1, \ldots (\mathcal{A}^{\mathrm{R}}_{\mathrm{N}})_k\right\}$ **proveď**

$(a^{\mathrm{GN}}, c^{\mathrm{U}}) \leftarrow \texttt{NeusporadaneIntegrovatPrvky}\left((\mathcal{A}^{\mathrm{R}}_{\mathrm{N}})_i, \mathtt{L}, \mathtt{n}, \mathtt{l}^{\mathrm{G}}\right)$

$\mathcal{A}^{\mathrm{G}} \leftarrow \mathcal{A}^{\mathrm{G}} \cup \{a^{\mathrm{GN}}\}$

$\mathcal{C} \leftarrow \mathcal{C} \cup \{c^{\mathrm{U}}\}$

ukonči pro všechny

$j \leftarrow 1$

pro všechny $a^{\mathrm{RL}}_{\mathrm{L}} \in \mathcal{A}^{\mathrm{RL}}_{\mathrm{L}}$ **proveď**

$(\mathcal{A}^{\mathrm{GN}}_{\Gamma})_j \leftarrow \emptyset$

$\left(\left\{a^{\mathrm{RN}}_{\Gamma_0}\right\}\right) \leftarrow \texttt{SestavitMnozinuPrvkuIncidujicichSObjektem}(q_0(a^{\mathrm{RL}}_{\mathrm{L}}), s^{\mathrm{R}})$

$\left(\left\{a^{\mathrm{RN}}_{\Gamma_1}\right\}\right) \leftarrow \texttt{SestavitMnozinuPrvkuIncidujicichSObjektem}(q_1(a^{\mathrm{RL}}_{\mathrm{L}}), s^{\mathrm{R}})$

$\left(\left\{a^{\mathrm{GN}}_{\Gamma_0}\right\}\right) \leftarrow \texttt{SestavitMnozinuKompozicnichPrvku}\left(a^{\mathrm{RN}}_{\Gamma_0}, \mathcal{C}\right)$

$\left(\left\{a^{\mathrm{GN}}_{\Gamma_1}\right\}\right) \leftarrow \texttt{SestavitMnozinuKompozicnichPrvku}\left(a^{\mathrm{RN}}_{\Gamma_1}, \mathcal{C}\right)$

$\mathcal{A}^{\mathrm{GN}}_{\Gamma} \leftarrow \left\{a^{\mathrm{GN}}_{\Gamma_0}, a^{\mathrm{GN}}_{\Gamma_1}\right\}$

$i \leftarrow 0$

prováděj

$i \leftarrow i + 1$

dokud není $i = j$ **nebo** $\mathcal{A}_{\Gamma}^{\mathrm{GN}} = \left(\mathcal{A}_{\Gamma}^{\mathrm{GN}}\right)_i$

jestliže $i = j$ **proveď**

$\left(\mathcal{A}_{\Gamma}^{\mathrm{GN}}\right)_i \leftarrow \mathcal{A}_{\Gamma}^{\mathrm{GN}}$

$\left(\mathcal{A}_{\mathrm{L}}^{\mathrm{RL}}\right)_i \leftarrow \left\{a_{\mathrm{L}}^{\mathrm{RL}}\right\}$

$\left(a_{\Gamma_0}^{\mathrm{GN}}\right)_i \leftarrow a_{\Gamma_0}^{\mathrm{GN}}$

$\left(a_{\Gamma_1}^{\mathrm{GN}}\right)_i \leftarrow a_{\Gamma_1}^{\mathrm{GN}}$

$j \leftarrow j + 1$

jinak proveď

$\left(\mathcal{A}_{\mathrm{L}}^{\mathrm{RL}}\right)_i \leftarrow \left(\mathcal{A}_{\mathrm{L}}^{\mathrm{RL}}\right)_i \cup \left\{a_{\mathrm{L}}^{\mathrm{RL}}\right\}$

ukonči jestliže

ukonči pro všechny

$h \leftarrow j - 1$

pro všechny $\left(\mathcal{A}_{\mathrm{L}}^{\mathrm{RL}}\right)_i \in \left\{\left(\mathcal{A}_{\mathrm{L}}^{\mathrm{RL}}\right)_1, \left(\mathcal{A}_{\mathrm{L}}^{\mathrm{RL}}\right)_2, \ldots \left(\mathcal{A}_{\mathrm{L}}^{\mathrm{RL}}\right)_h\right\}$ **proveď**

$\left(a^{\mathrm{GL}}, c^{\mathrm{U}}\right) \leftarrow \texttt{NeusporadaneIntegrovatPrvky}\left(\left(\mathcal{A}_{\mathrm{L}}^{\mathrm{RL}}\right)_i, \mathrm{L}, n, l^{\mathrm{G}}\right)$

$\mathcal{A}^{\mathrm{G}} \leftarrow \mathcal{A}^{\mathrm{G}} \cup \left\{a^{\mathrm{GL}}\right\}$

$\mathcal{C} \leftarrow \mathcal{C} \cup \left\{c^{\mathrm{U}}\right\}$

$\left(r_0^{\mathrm{G}}\right) \leftarrow \texttt{VytvoritVazbuMeziPolyPrvku}\left(q_0\left(\left(a_{\Gamma_0}^{\mathrm{GN}}\right)_i\right), q_0(a^{\mathrm{GL}}), 1, n, l^{\mathrm{G}}\right)$

$\left(r_1^{\mathrm{G}}\right) \leftarrow \texttt{VytvoritVazbuMeziPolyPrvku}\left(q_1(a^{\mathrm{GL}}), q_0\left(\left(a_{\Gamma_1}^{\mathrm{GN}}\right)_i\right), 1, n, l^{\mathrm{G}}\right)$

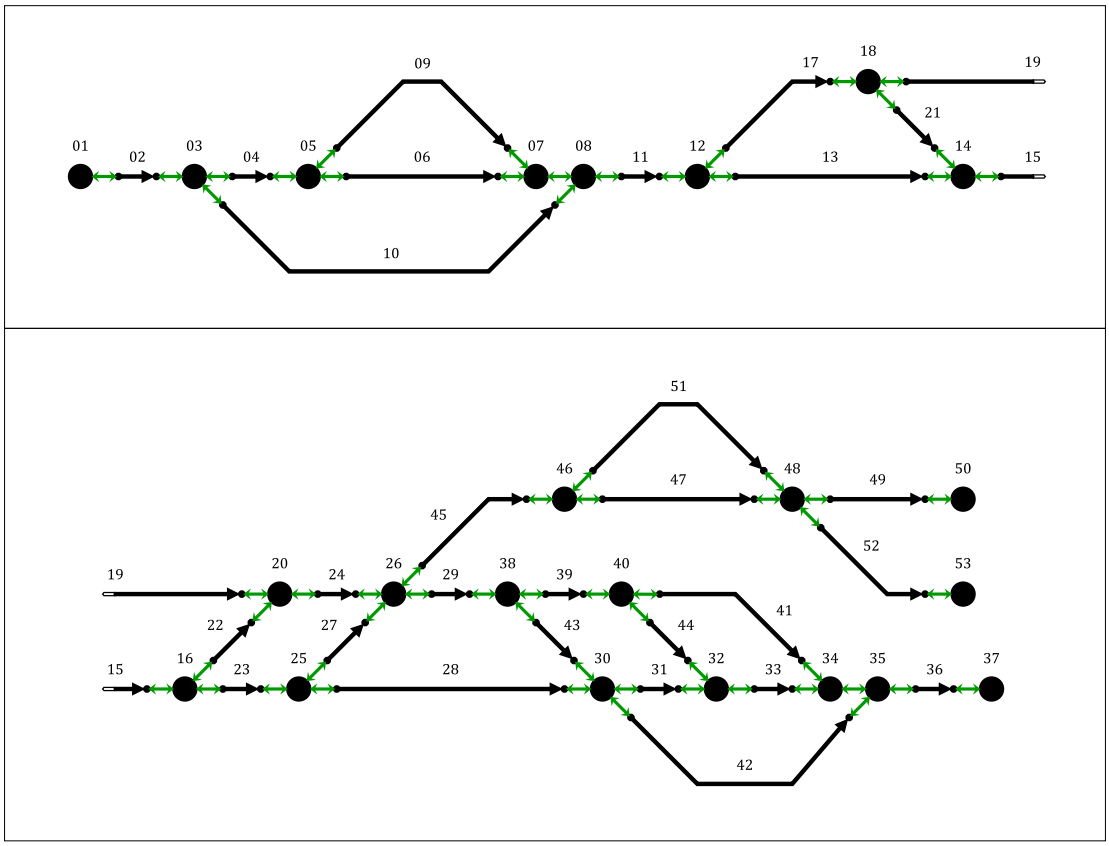
$\mathcal{R}^{\mathrm{G}} \leftarrow \mathcal{R}^{\mathrm{G}} \cup \left\{r_0^{\mathrm{G}}, r_1^{\mathrm{G}}\right\}$

ukonči pro všechny

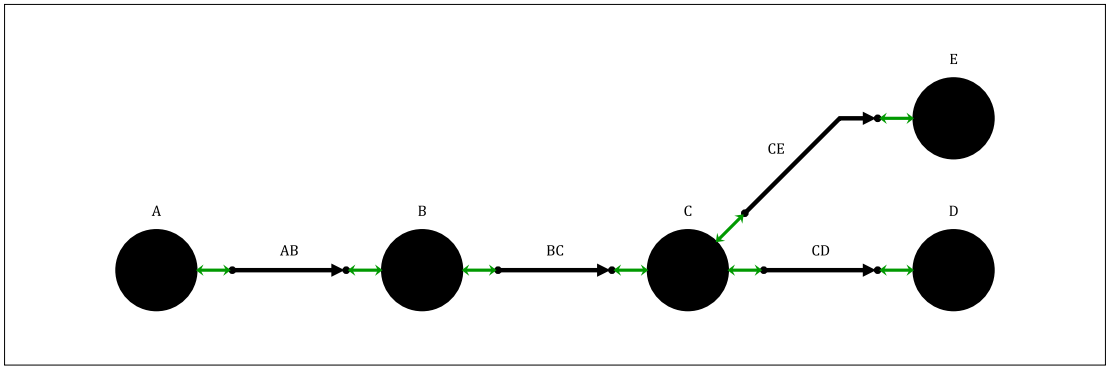
$s^{\mathrm{G}} \leftarrow \left(\mathcal{A}^{\mathrm{G}}, \mathcal{R}^{\mathrm{G}}\right)$

vrať: $\left(s^{\mathrm{G}}, \mathcal{C}\right)$

Příklad možné referenční struktury s^{R}, na níž může být aplikován navržený integrační algoritmus t_{RG}, je uveden na obrázku 6.1. V případě navrženého integračního algoritmu t_{RG} závisí průběh integračního procesu na hodnotách stavební délky λ, jimiž jsou popsány liniové prvky, která figuruje v klasifikačním kritériu délky ψ_{D}.



Obrázek 6.1. Příklad referenční struktury pro navržený integrační algoritmus.



Obrázek 6.2. Příklad méně podrobné struktury, která může vzniknout jako výstup navrženého integračního algoritmu.

Za předpokladu, že podmínka klasifikačního kritéria λ pro integraci liniových prvků v prvky neliniové není splněna pouze pouze pro prvky 11, 23, 24 a 45, což znamená, že tyto prvky reprezentují koleje, jejichž stavební délka vyjádřená v metrech není menší než expertně stanovená mezní hodnota této veličiny Λ, získáme aplikací algoritmu t_{RG} strukturu s^{F} vyjádřenou na obrázku 6.2.

Kapitola 7
Návrh dezintegračního algoritmu

Za účelem transformovat známou referenční strukturu s^{R} sytému $\mathbb{T}$ topologického popisu sítě n, vyjádřenou na referenční rozlišovací úrovni l^{R}, na hledanou podrobnější strukturu s^{F} tohoto systému, vyjádřenou na podrobnější rozlišovací úrovni l^{F}, bude pro daný aplikační případ w_{RF} navrhován dezintegrační algoritmus t_{RF}. Návrh toho algoritmu závisí na typech $\tau_{\mathrm{S}}^{\mathrm{R}}$ a $\tau_{\mathrm{S}}^{\mathrm{F}}$ přípustných struktur vyjádřených na rozlišovacích úrovních l^{R} a l^{F}. V rámci této kapitoly bude definován rámcový postup návrhu integračního algoritmu a uveden příklad aplikace tohoto postupu pro konkrétní aplikační případ w_{RF}.

7.1 Postup při návrhu dezintegračního algoritmu

- Před vlastním návrhem dezintegračního algoritmu t_{RF} nejprve specifikujeme typy přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{R}}$ a $\tau_{\mathrm{S}}^{\mathrm{F}}$.
- V souladu s typy přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{R}}$ a $\tau_{\mathrm{S}}^{\mathrm{F}}$ provedeme posouzení, zda je za všech okolností možné každému v úvahu přicházejícímu prvku $a^{\mathrm{R}} \in \mathcal{A}^{\mathrm{R}}$ jednoznačně přiřadit některý v úvahu přicházející prvek $a^{\mathrm{F}} \in \mathcal{A}^{\mathrm{F}}$ v roli jeho části. Toto posouzení provádíme na obecné úrovni na základě toho, jak rozsáhlou a jakým způsobem vymezenou část železniční (resp. drážní) infrastruktury tyto prvky reprezentují.
- V případě, že je výsledek výše uvedeného posouzení pozitivní a je to účelné, navrhneme takový typ odvozené struktury $\tau_{\mathrm{S}}^{\mathrm{D}}$, pro každý jejíž v úvahu přicházející prvek $a^{\mathrm{D}} \in \mathcal{A}^{\mathrm{D}}$ platí, že je jej za všech okolností možné jednoznačně přiřadit některému v úvahu přicházejícímu prvku $a^{\mathrm{R}} \in \mathcal{A}^{\mathrm{R}}$ v roli jeho části a zároveň je mu za všech okolností možné jednoznačně přiřadit některý v úvahu přicházející prvek $a^{\mathrm{F}} \in \mathcal{A}^{\mathrm{F}}$ v roli jeho části. V souladu s typem odvozené struktury $\tau_{\mathrm{S}}^{\mathrm{D}}$ je pak nutné zavést odvozenou rozlišovací úroveň l^{D} a navrhnout dezintegrační algoritmy t_{RD} a t_{DF}.
- V případě, že je výsledek výše uvedeného negativní, navrhneme takový typ odvozené struktury $\tau_{\mathrm{S}}^{\mathrm{D}}$, pro každý jejíž v úvahu přicházející prvek $a^{\mathrm{D}} \in \mathcal{A}^{\mathrm{D}}$ platí, že je mu za všech okolností možné jednoznačně přiřadit některý v úvahu přicházející prvek $a^{\mathrm{R}} \in \mathcal{A}^{\mathrm{R}}$ v roli jeho části a zároveň některý v úvahu přicházející prvek $a^{\mathrm{G}} \in \mathcal{A}^{\mathrm{G}}$ v roli jeho části, nebo že je jej za všech okolností možné jednoznačně přiřadit některému v úvahu přicházejícímu prvku $a^{\mathrm{R}} \in \mathcal{A}^{\mathrm{R}}$ v roli jeho části a zároveň některému v úvahu přicházejícímu prvku $a^{\mathrm{G}} \in \mathcal{A}^{\mathrm{G}}$ v roli jeho části. V souladu s typem odvozené struktury $\tau_{\mathrm{S}}^{\mathrm{D}}$ je nutné zavést odvozenou rozlišovací úroveň l^{D} a navrhnout transformační algoritmy t_{RD} a t_{DG}, z nichž právě jeden je algoritmem dezintegračním a právě jeden algoritmem integračním.
- V souladu s typem přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{F}}$ specifikujeme, jaké různé způsoby dezintegrace budeme na základě v úvahu přicházejících tříd prvků množiny $\mathcal{A}^{\mathrm{R}}$ a případně i dalších faktorů při návrhu integračního algoritmu t_{RF} uplatňovat.

- Navrhneme způsob klasifikace prvků množiny $\mathcal{A}^{\mathrm{R}}$, který zajistí rozděleni prvků této množiny do několika disjunktních podmnožin podle specifikovaných způsobů dezintegrace. Dezintegrační algoritmus pak zkonstruujme tak, aby jeho prostřednictvím bylo možné provádět dezintegraci prvku a_i^{G} na substrukturu $\tilde{s}_i$ (resp. rozšířenou substrukturu $\tilde{s}_i$) odpovídající jejich vnitřním strukturám vždy v souladu s příslušným specifikovaným způsobem.
- V souladu s typem přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{F}}$ navrhneme, jakým způsobem mají být na základě existujících vazeb mezi prvky množiny $\mathcal{A}^{\mathrm{F}}$ a případně i dalších faktorů vytvářeny vazby mezi prvky různých struktur (resp. rozšířených struktur) s_i^{F}, které jsou substrukturami (resp. rozšířenými substrukturami) nově vznikající struktury s^{F}. Pokud využíváme dezintegrace prvků a_i^{G} na rozšířené substruktury, je možné nutnou míru zásahu zpracovatele při realizaci dezintegračního algoritmu t_{RF} vytvořením aparátu pro automatizované ztotožňování konektorů.
- V souladu s typem přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{F}}$ navrhneme, jakým způsobem mají být na základě hodnot veličin popisujících prvky množiny $\mathcal{A}^{\mathrm{R}}$ a případně i dalších faktorů generovány hodnoty veličin popisující prvky množiny $\mathcal{A}^{\mathrm{F}}$.
- Jako vstupy dezintegračního algoritmu t_{RF} budeme požadovat strukturu s^{F} (rozumí se včetně popisu veškerých jejích komponent hodnotami veličin v souladu s typem přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{F}}$), množinu kolekcí $\mathcal{C}$ (je-li systém $\mathbb{T}$ popsán pouze na rozlišovací úrovni l^{R}, lze předpokládat, že se na vstupu jedná o prázdnou množinu, a dále síť n a rozlišovací úroveň l^{F} (ve smyslu jejich identifikátorů), jimž mají být přiřazovány jednotlivé komponenty struktury s^{F}. Další potřebné vstupy mohou být definovány na základě konkrétní realizace návrhu algoritmu t_{RF} v závislosti na požadovaných vstupech jednotlivých operací, kterých algoritmus využívá. Může se jednat např. o klasifikační kritéria, podmínky regularity, pravidla příslušnosti k třídě a množiny předdefinovaných vzorů (rozšířené) vnitřní struktury.
- Jako výstupy dezintegračního algoritmu t_{RF} budeme požadovat strukturu s^{F} (rozumí se včetně popisu veškerých jejích komponent hodnotami veličin v souladu s typem přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{F}}$), množinu kolekcí $\mathcal{C}$ (na výstupu nově obsahující i kolekce vytvořené při dezintegraci prvků množiny $\mathcal{A}^{\mathrm{R}}$ v prvky množiny $\mathcal{A}^{\mathrm{F}}$).

7.2 Příklad návrhu dezintegračního algoritmu

Postup při návrhu dezintegračního algoritmu t_{RF} budeme demonstrovat na aplikačním případu w_{RF} specifikovaném následujícím způsobem: Referenční struktura s^{R} je tvořena prvky reprezentujícími dopravny a traťové úseky tyto dopravny propojující. Navrhovaný integrační algoritmus t_{RF} má zajistit transformaci referenční struktury s^{R} na podrobnější strukturu s^{F}, která je tvořená prvky reprezentující kolejové trasy. Upřesňující informace týkající se vstupních podmínek budou uvedeny při specifikaci jednotlivých typů přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{R}}$ a $\tau_{\mathrm{S}}^{\mathrm{F}}$. Bude se přitom jednat o téměř minimální požadavky nutné pro realizaci navrhovaného integračního algoritmu t_{RF}.

7.2.1 Specifikace typů přípustné struktury

Pro typ přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{R}}$ uvažujeme, že struktura tohoto typu bude různorodou strukturou, jejíž součástí mohou být jak prvky třídy neliniových prvků, tak prvky třídy liniových prvků. V rámci těchto tříd nebudeme dále výslovně rozlišovat jednotlivé funkční typy prvků. Budeme předpokládat, že neliniové prvky reprezentují obecně

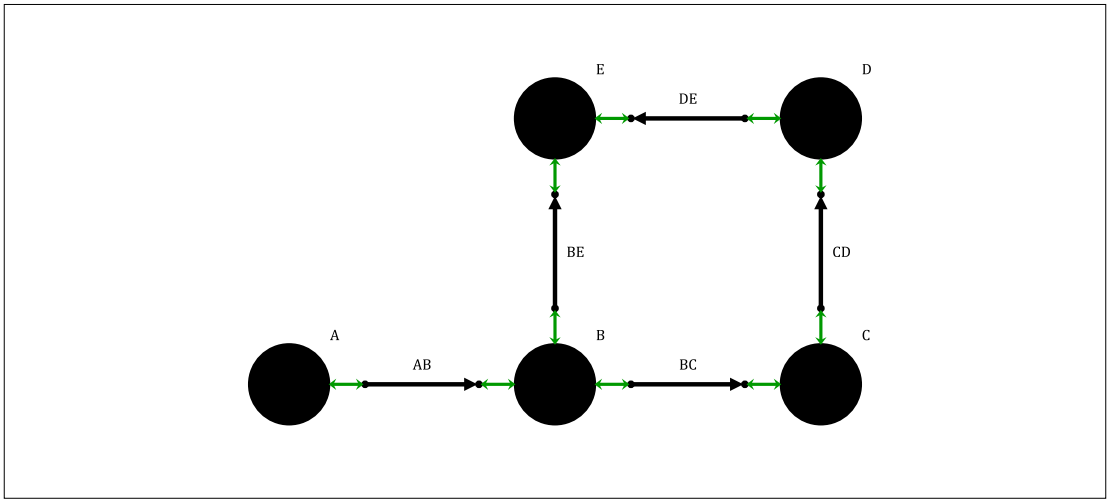
dopravny a liniové prvky traťové úseky propojující dopravny. Budeme uvažovat pouze s průchodnými vazbami.

Neliniové prvky budou popsány hodnotou identifikátoru ι, příslušnosti ke třídě ϵ_{A} a dále veličinami počet kolejí, počet výhybek a počet kolejových křižovatek, které označíme jako σ_{K}, σ_{V}, resp. σ_{X}, vyjadřujícími počet kolejí v dopravním smyslu, počet výhybek, resp. počet kolejových křižovatek, které přísluší dopravně reprezentované příslušným neliniovým prvkem. Tuto skutečnost vyjádříme jako

$$\mathcal{M}\left(\mathcal{A}^{\mathrm{RN}}\right) = \{\iota, \epsilon_{\mathrm{A}}, \sigma_{\mathrm{K}}, \sigma_{\mathrm{V}}, \sigma_{\mathrm{X}}\}\,.$$

Pro účely realizace algoritmu t_{RF} připustíme rozšíření množiny $\mathcal{M}\left(\mathcal{A}^{\mathrm{RN}}\right)$ o další veličiny. Liniové prvky budou dále popsány hodnotami veličiny počet kolejí, kterou označíme jako σ_{K}, vyjadřující počet kolejí, které přísluší traťovému úseku reprezentovanému příslušným prvkem a^{RL}. Tuto skutečnost vyjádříme jako

$$\mathcal{M}\left(\mathcal{A}^{\mathrm{RL}}\right) = \{\iota, \epsilon_{\mathrm{A}}, \sigma_{\mathrm{K}}\}\,.$$



Obrázek 7.1. Příklad referenční struktury pro navržený dezintegrační algoritmus

Objekt reálného světa bude reprezentován neliniovým prvkem, pokud se bude jednat o dopravnu. Objekt reálného světa bude reprezentován liniovým prvkem, pokud se bude jednat o ucelený traťový úsek mezi dopravnami. Při vytváření vazeb musí být respektována následující pravidla:

- Vazba vyjadřuje funkční návaznost dvou sousedících objektů reálného světa reprezentovaných prvky, mezi nimiž je vytvořena. Musí být vytvořena vždy, když tato návaznost existuje, v případě liniového prvku je navázána na ten jeho pól, který odpovídá tomu konci traťového úseku, který příslušný pól reprezentuje.
- Žádný prvek nemůže být vazbou propojen prvkem s příslušností ke stejné třídě.
- Každý liniový prvek musí být vazbami propojen právě se dvěma neliniovými prvky, každá z nichž je k tomuto liniovému prvku navázána na jiném pólu.

Pro typ přípustné struktury $\tau_{\mathrm{S}}^{\mathrm{F}}$ uvažujeme, že struktura tohoto typu bude stejnorodou strukturou liniových prvků, která může být rozšířena o konektory. V rámci třídy liniových prvků nebudeme dále výslovně rozlišovat jednotlivé funkční typy prvků. Budeme

předpokládat, že liniové prvky reprezentují obecně kolejové trasy. Budeme uvažovat s průchodnými i neprůchodnými vazbami.

Liniové prvky budou popsány pouze hodnotou identifikátoru ι a příslušnosti ke třídě ϵ_{A}. Tuto skutečnost vyjádříme jako

$$\mathcal{M}(\mathcal{A}^{\mathrm{RN}}) = \{\iota, \epsilon_{\mathrm{A}}\}\,.$$

Konektory mohou být dále popsány dalšími veličinami zavedenými pro účely realizace algoritmu t_{RF}.

Objekt reálného světa bude reprezentován neliniovým prvkem, pokud se bude jednat o kolejovou trasu. Oblast železniční (resp.) drážní infrastruktury můžeme na kolejové trasy pro účely vyjádření struktur l^{F} rozčlenit zcela libovolně, ale musí být jimi zcela popsána, aniž by docházelo k jejich překryvu. Při vytváření vazeb musí být respektována následující pravidla:

- Každá průchodná vazba vyjadřuje funkční návaznost dvou kolejových tras, mezi nimiž je vytvořena, nebo funkční napojení kolejové trasy na konektor. Musí být vytvořena vždy, když tato návaznost existuje, tedy v případě, že se jedná na sebe navazující kolejové trasy.
- Každá neprůchodná vazba vyjadřuje fyzickou návaznost dvou kolejových tras, mezi nimiž je vytvořena, která není návazností funkční. Musí být vytvořena vždy, když tato návaznost existuje, tedy v případě, že se jedná o sbíhající se kolejové trasy.
- Neliniové prvky mohou být mezi sebou vazbami navázanými na póly reprezentující příslušné konce příslušných kolejových tras bez omezení.

7.2.2 Aplikovaný postup při návrhu dezintegračního algoritmu

Jelikož struktura s^{F} může být navržena takovým způsobem, aby každý její prvek reprezentoval takovou část infrastruktury, která je vždy zcela součástí některého v úvahu přicházejícího objektu reálného světa reprezentovaného prvkem struktury s^{R}, je možné transformaci struktury s^{R} na strukturu s^{F} provádět přímo. Kolejové trasy mohou být členěny libovolně, tedy i tak, aby mohly být vždy cela zahrnuty do oblasti reprezentující některou dopravnu nebo traťový úsek.

Množinu prvků referenční struktury $\mathcal{A}^{\mathrm{R}}$ nejprve podrobíme binární klasifikaci na prvky neliniové a na prvky liniové. K tomu použijeme binární klasifikační kritérium třídy ψ_{C}. Nechť je toto klasifikační kritérium aplikované na prvek a^{R} definované následujícím způsobem:

$$\psi_{\mathrm{C}}\colon\quad \epsilon_{\mathrm{A}}(a^{\mathrm{R}}) = \mathtt{N}$$

Takové prvky množiny $\mathcal{A}^{\mathrm{R}}$, pro něž je toto kritérium splněno, zařadíme do množiny $\mathcal{A}^{\mathrm{RN}}$. Takové prvky množiny $\mathcal{A}^{\mathrm{R}}$, pro něž tato podmínka splněna není, zařadíme do množiny do množiny $\mathcal{A}^{\mathrm{RL}}$. Prvky každé z těchto množin odpovídajících jejich třídám budeme dezintegrovat jiným způsobem, v obou případech však za vzniku rozšířené vnitřní struktury.

Nejprve přistoupíme k dezintegraci neliniových prvků množiny $\mathcal{A}^{\mathrm{RN}}$ reprezentujících dopravny. Generování vnitřní struktury by v případě těchto prvků bylo nesmírně složité a vyžadovalo by značné množství hodnot různých popisných veličin, proto přistoupíme k dezintegraci přiřazením rozšířené vnitřní struktury. Předpokládáme, že máme k dispozici množinu předdefinovaných vzorů rozšířené vnitřní struktury ${}^{*}\tilde{\mathcal{S}}^{\mathrm{F}}_{\mathrm{R}}$ pro dezintegraci prvků figurujících na rozlišovací úrovni l^{R} na jejich vnitřní strukturu, která je v souladu se specifikací přípustného typu struktury γ^{F}, vyjádřenou na rozlišovací úrovni l^{F}.

O každém z těchto vzorů předpokládáme, že má modelované konektory v místech, kde je možné strukturu jím definovanou propojit s jinými strukturami, je popsán hodnotami těch samých veličin jako neliniové prvky množiny $\mathcal{A}^{\mathrm{RN}}$, a dále veličinou δ udávající přípustné konfigurace konektorových skupin pro účely jejich integrace společně s rozšířenou strukturou vzoru ${}^{*}\tilde{\mathcal{S}}^{\mathrm{F}}_{\mathrm{R}}$ v konektory figurující na rozlišovací úrovni l^{R}. Tuto skutečnost vyjádříme jako

$$\mathcal{M}\left({}^{*}\tilde{\mathcal{S}}^{\mathrm{F}}_{\mathrm{R}}\right) = \{\iota, \epsilon_{\mathrm{A}}, \sigma_{\mathrm{K}}, \sigma_{\mathrm{V}}, \sigma_{\mathrm{X}}, \delta\}\,.$$

Při dezintegraci každého prvku $a^{\mathrm{RN}} \in \mathcal{A}^{\mathrm{RN}}$ budeme hledat podmnožinu vzorů ${}^{*}\tilde{\mathcal{S}}^{\mathrm{F}}_{\mathrm{R}}$, které vyhoví podmínkám regularity, které je nutné splnit, aby na jejich základě mohla být vytvořena rozšíření vnitřní struktura prvku a^{RN}. Rozšířené vnitřní struktury nejprve podrobíme posouzení topologické regularity. Pro posouzení topologické regularity pro dezintegraci prvku a^{RN} pomocí vzoru rozšířené vnitřní struktury ${}^{*}\tilde{s}^{\mathrm{F}}_{\mathrm{R}}$ stanovíme podmínku φ_{T}, že v rámci přípustných konfigurací konektorových skupin vzoru ${}^{*}\tilde{s}^{\mathrm{F}}_{\mathrm{R}}$ musí existovat taková konfigurace, která obsahuje stejný počet konektorových skupin, jako je vazeb incidujících s prvkem a^{RN} (udávaných pro tento účel pro neliniové prvky nově zavedenou veličinou γ^{R}), a zároveň počet konektorů množiny ${}^{*}\mathcal{B}^{\mathrm{F}}_{\mathrm{R}}$ vzoru rozšířené vnitřní struktury ${}^{*}\tilde{s}^{\mathrm{F}}_{\mathrm{R}}$ musí být stejný, jako je součet hodnot veličiny σ_{K}, vyjadřující počet kolejí traťového úseku, všech prvků incidujících s prvkem a^{RN} (udávaných pro tento účel pro neliniové prvky nově zavedenou veličinou γ^{F}). Podmínka topologické regularity φ_{T} aplikovaná prvek a^{RN} a vzor ${}^{*}\tilde{s}^{\mathrm{F}}_{\mathrm{R}}$ je definovaná následujícím způsobem:

$$\varphi_{\mathrm{T}}\colon\ \ \exists d \in \delta({}^{*}\tilde{s}^{\mathrm{F}}_{\mathrm{R}})\,; d = \gamma^{\mathrm{R}}(a^{\mathrm{RN}})\ \wedge\ \left|{}^{*}\mathcal{B}^{\mathrm{F}}_{\mathrm{R}}\right| = \gamma^{\mathrm{F}}(a^{\mathrm{RN}})$$

Každý vzor ${}^{*}\tilde{s}^{\mathrm{F}}_{\mathrm{R}}$ dále podrobíme posouzení parametrické regularity. K tomu využijeme podmínku parametrické regularity φ_{P}. Budeme porovnávat hodnoty veličin počet koleji σ_{K}, počet výhybek σ_{V} a počet kolejových křižovatek $_{\mathrm{X}}$ popisujících daný vzor s hodnotami těchto veličin na straně prvku a^{RN}. Požadujeme, aby se hodnoty každé z těchto veličin pro vzor i prvek rovnaly. Podmínka topologické regularity φ_{P} aplikovaná prvek a^{RN} a vzor ${}^{*}\tilde{s}^{\mathrm{F}}_{\mathrm{R}}$ je definovaná následujícím způsobem:

$$\varphi_{\mathrm{P}}\colon\ \ \sigma_{\mathrm{K}}({}^{*}\tilde{s}^{\mathrm{F}}_{\mathrm{R}}) = \sigma_{\mathrm{K}}(a^{\mathrm{RN}})\ \wedge\ \sigma_{\mathrm{V}}({}^{*}\tilde{s}^{\mathrm{F}}_{\mathrm{R}}) = \sigma_{\mathrm{V}}(a^{\mathrm{RN}})\ \wedge\ \sigma_{\mathrm{X}}({}^{*}\tilde{s}^{\mathrm{F}}_{\mathrm{R}}) = \sigma_{\mathrm{X}}(a^{\mathrm{RN}})$$

Tímto jsme omezili výběr přípustných vzorů rozšířených vnitřních struktur, na které je možné dezintegrovat prvek a^{RN}. Volba výsledného vzoru vnitřní struktury, který bude pro dezintegraci prvku a^{RN} použit, je závislá na zpracovateli, který z množiny topologicky a parametricky regulárních vzorů rozšířených vnitřních struktur pro účely dezintegrace prvku a^{RN}, vybere ten, která odpovídá skutečnosti. Může se stát, že žádný takový vzor rozšířené vnitřní struktur v této množině neexistuje (množina může být prázdná, nebo žádný ze vzorů rozšířených vnitřních struktur této množiny neodpovídá zpracovatelem rozeznané realitě). V takovém případě musí zpracovatel vytvořit nový vzor rozšířené vnitřní struktury (pokud možno se záměrem, aby již byla pro daný účel dezintegrace prvku a^{RN} po všech stránkách vyhovující), popsat jej odpovídajícími hodnotami požadovaných veličin a nechat jej opět posoudit na topologickou a parametrickou regularitu vůči prvku a^{RN}.

Teprve na základě zpracovatelem vybraného vzoru bude vytvořena vzoru odpovídající rozšířená substruktura $\tilde{s}^{\mathrm{FL}}_{\mathrm{SN}}$ a vytvořena kolekce vyjadřující, že prvky této její prvky jsou součástmi prvku a^{RN}. Z přípustných konfigurací konektorových skupin musí zpracovatel vybrat tu vyhovující a v souladu v ní přiřadit nově vzniklým konektorům těchto jednotlivých skupin některou z vazeb incidujících s prvkem a^{RN}. Dojde k zavedení nové veličiny ζ, kterou budou popsány konektory. Každému konektoru bude přiřazena

hodnota této veličiny vyjadřující identifikátor příslušné vazby. Celý tento postup bude opakován pro všechny prvky množiny $\mathcal{A}^{\mathrm{RN}}$.

Po vyčerpání všech prvků množiny $\mathcal{A}^{\mathrm{RN}}$ bude zahájena dezintegrace liniových prvků množiny $\mathcal{A}^{\mathrm{RL}}$. Na tyto prvky bude aplikován princip paralelní dezintegrace, při které bude jako parametr využit počet kolejí. Pravidlo přidělování příslušnosti k třídě prvku χ_{C} s ohledem na zamýšlený záměr definuje tak, že pro každou hodnotu parametru ρ_{C} generování příslušnosti k třídě ϵ_{A} prvku topologického popisu železniční (resp. drážní) infrastruktury danou pořadím nově vznikajícího prvku $a^{\mathrm{FL}}_{\mathrm{SL}}$ bude generovat hodnotu `L`. Bude tedy možné jej zapsat následujícím způsobem:

$$\chi_{\mathrm{C}}\colon \quad \forall \rho_{\mathrm{C}}; \epsilon_{\mathrm{A}}(\rho_{\mathrm{C}}) = \mathtt{L}.$$

Každý prvek a^{RL} bude tím pádem dezintegrován na stejnorodou rozšířenou substrukturu $\tilde{s}^{\mathrm{FL}}_{\mathrm{SL}}$ liniových prvků o celkovém počtu $\sigma_{\mathrm{K}}(a^{\mathrm{RL}})$, ke každému pólu každého z nichž bude navázán jeden konektor. Zároveň dojde k vytvořené nové neuspořádané kolekce c^{U} vyjadřující, že prvky této substruktury jsou částmi prvku a^{RL}. Také uvedeným konektorům, vždy podle příslušných pólů s konektory incidujících nově vznikajících prvků figurujících na podrobnější rozlišovací úrovni l^{R}, přiřadíme hodnotu veličiny ζ nabývající hodnoty identifikátoru vazby incidující s prvkem a^{RL},

Ihned po přiřazení této hodnoty konektorům vytvořeným v průběhu dezintegrace prvku a^{RL} můžeme přistoupit k přiřazováním právě vytvořených konektorů již existujícím konektorům vytvořeným při dezintegraci neliniových prvků. Podmínka propojitelnosti konektoru $b^{\mathrm{F}}_{\mathrm{L}}$ vytvořeného při dezintegraci liniového prvku s konektorem vytvořeným při dezintegraci neliniového prvku φ_{S} je definována následujícím způsobem:

$$\varphi_{\mathrm{S}}\colon \quad \zeta(b^{\mathrm{F}}_{\mathrm{L}}) = \zeta(b^{\mathrm{F}}_{\mathrm{N}})$$

Tato podmínka vyjadřuje, že je možné ztotožnit a následně vazbou nahradit dva konektory různých rozšířených substruktur, které odpovídají stejné vazbě figurující na referenční rozlišovací úrovni. Pokud má traťový úsek reprezentovaný dezintegrovaným prvkem a^{RL} více než jednu kolej, možností takového propojení existuje více. Na této úrovni již musí rozhodnout zpracovatel, které konektory je možné ztotožnit. Na základě tímto způsobem sestavené množiny přiřazení konektorů budou vytvářeny průchodné vazby mezi póly prvků iniciujícími s předmětnými konektory. Po vytvoření vazby dojde k odstranění již nepotřebných konektorů a konektorových vazeb.

Okamžikem, kdy jsou tímto způsobem dezintegrovány všechny prvky množiny $\mathcal{A}^{\mathrm{RL}}$ a vytvořeny i související vazby, získáváme kompletní strukturu s^{F} i množinu kolekcí $\mathcal{C}$ doplněnou o nové neuspořádané kolekce. Kompletní přehled požadovaných vstupů a získaných výstupů a možná podoba navrženého dezintegračního algoritmu je uvedena níže.

Vstupy:

- s^{R} – referenční struktura
- n – síť, které mají být přiřazeny komponenty podrobnější struktury
- l^{F} – podrobnější rozlišovací úroveň, které mají být přiřazeny komponenty podrobnější struktury
- $\mathcal{C}$ – počáteční množina kolekcí (typicky prázdná množina)
- ψ_{C} – klasifikační kritérium třídy

- φ_{T} – podmínka topologické regularity
- φ_{P} – podmínka parametrické regularity
- χ_{C} – pravidlo přidělování příslušnosti k třídě
- φ_{S} – podmínka propojitelnosti

Výstupy:

- s^{F} – podrobnější struktura
- $\mathcal{C}$ – výsledná množina kolekcí

Zápis:

- $(s^{\mathrm{F}}, \mathcal{C}) \leftarrow \texttt{DezintegracniAlgoritmusRF}\left(s^{\mathrm{R}}, \mathcal{C}, n, l^{\mathrm{F}}, {}^{*}\tilde{\mathcal{S}}^{\mathrm{F}}_{\mathrm{R}}, \psi_{\mathrm{C}}, \varphi_{\mathrm{T}}, \varphi_{\mathrm{P}}, \chi_{\mathrm{C}}, \varphi_{\mathrm{S}}\right)$

Algoritmus : $\texttt{DezintegracniAlgoritmusRF}\left(s^{\mathrm{R}}, \mathcal{C}, n, l^{\mathrm{F}}, {}^{*}\tilde{\mathcal{S}}^{\mathrm{F}}_{\mathrm{R}}, \psi_{\mathrm{C}}, \varphi_{\mathrm{T}}, \varphi_{\mathrm{P}}, \chi_{\mathrm{C}}, \varphi_{\mathrm{S}}\right)$

$\mathcal{A}^{\mathrm{F}} \leftarrow \emptyset$

$\mathcal{R}^{\mathrm{F}} \leftarrow \emptyset$

$(\mathcal{A}^{\mathrm{R}}, \mathcal{R}^{\mathrm{R}}) \leftarrow s^{\mathrm{R}}$

$(\mathcal{A}^{\mathrm{RN}}, \mathcal{A}^{\mathrm{RL}}) \leftarrow \texttt{BinarneKlasifikovatPrvky}(\mathcal{A}^{\mathrm{R}}, \psi_{\mathrm{C}})$

pro všechny $a^{\mathrm{RN}} \in \mathcal{A}^{\mathrm{RN}}$ **proveď**

$(\mathcal{R}^{\mathrm{R}}_{\Gamma}) \leftarrow \texttt{SestavitMnozinuVazebIncidujicichSObjektem}(a^{\mathrm{RN}}, s^{\mathrm{R}})$

$\Gamma^{\mathrm{R}} \leftarrow \left|\mathcal{R}^{\mathrm{R}}_{\Gamma}\right|$

$(\mathcal{A}_{\Gamma}) \leftarrow \texttt{SestavitMnozinuPrvkuIncidujicichSObjektem}(a^{\mathrm{RN}}, s^{\mathrm{R}})$

$\Gamma^{\mathrm{F}} \leftarrow 0$

pro všechny $a^{\mathrm{R}}_{\Gamma} \in \mathcal{A}^{\mathrm{R}}_{\Gamma}$ **proveď**

$\Gamma^{\mathrm{F}} \leftarrow \Gamma^{\mathrm{F}} + \sigma_{\mathrm{K}}(a^{\mathrm{R}}_{\Gamma})$

ukonči pro všechny

$(a^{\mathrm{RN}}) \leftarrow \texttt{PriraditHodnotuVelicinyObjektu}(\gamma^{\mathrm{R}}, \Gamma^{\mathrm{R}}, a^{\mathrm{RN}})$

$(a^{\mathrm{RN}}) \leftarrow \texttt{PriraditHodnotuVelicinyObjektu}(\gamma^{\mathrm{F}}, \Gamma^{\mathrm{F}}, a^{\mathrm{RN}})$

$(\tilde{s}^{\mathrm{FL}}_{\mathrm{SN}}, c^{\mathrm{U}}) \leftarrow \texttt{DezintegrovatPrvekPrirazenimStruktury}\left(a^{\mathrm{RN}}, {}^{*}\tilde{\mathcal{S}}^{\mathrm{F}}_{\mathrm{R}}, l^{\mathrm{F}}, n, \varphi_{\mathrm{T}}, \varphi_{\mathrm{P}}\right)$

$\left(\mathcal{A}^{\mathrm{FL}}_{\mathrm{SN}}, \tilde{\mathcal{R}}^{\mathrm{F}}_{\mathrm{SN}}, \mathcal{B}^{\mathrm{F}}_{\mathrm{SN}}\right) \leftarrow \tilde{s}^{\mathrm{FL}}_{\mathrm{SN}}$

$(\mathcal{D}) \leftarrow \delta(\tilde{s}^{\mathrm{FL}}_{\mathrm{SN}})$

$\mathcal{D}_{\mathrm{O}+} \leftarrow \emptyset$

pro všechny $d \in \mathcal{D}$ **proveď**

jestliže $|d| = \Gamma^{\mathrm{R}}$ **proveď**

$\mathcal{D}_{\mathrm{O}+} \leftarrow \mathcal{D}_{\mathrm{O}+} \cup \{d\}$

ukonči jestliže

ukonči pro všechny

$(d_{\mathrm{U}}) \leftarrow \texttt{VybratVyhovujiciObjektZpracovatelem}(\mathcal{D}_{\mathrm{O}+})$

pro všechny $\mathcal{B}_{\mathrm{U}} \in d_{\mathrm{U}}$ **proveď**

$(r^{\mathrm{R}}_{\Gamma}) \leftarrow \texttt{VybratVyhovujiciObjektZpracovatelem}(\mathcal{R}^{\mathrm{R}}_{\Gamma})$

$(\mathcal{B}_{\mathrm{U}}) \leftarrow \texttt{PriraditHodnotuVelicinyObjektum}(\zeta, \iota(r^{\mathrm{R}}_{\Gamma}), \mathcal{B}_{\mathrm{U}})$

$\mathcal{R}^{\mathrm{R}}_{\Gamma} \leftarrow \mathcal{R}^{\mathrm{R}}_{\Gamma} \setminus \{r^{\mathrm{R}}_{\Gamma}\}$

ukonči pro všechny

$\left(\tilde{\mathcal{R}}^{\mathrm{F}}_{\mathrm{SN}}\right) \leftarrow \texttt{SestavitMnozinuVazebIncidujicichSObjekty}(\mathcal{B}^{\mathrm{F}}_{\mathrm{SN}}, \tilde{s}^{\mathrm{FL}}_{\mathrm{SN}})$

$\mathcal{R}^{\mathrm{F}}_{\mathrm{SN}} \leftarrow \tilde{\mathcal{R}}^{\mathrm{F}}_{\mathrm{SN}} \setminus \hat{\mathcal{R}}^{\mathrm{F}}_{\mathrm{SN}}$

$\mathcal{A}^{\mathrm{F}} \leftarrow \mathcal{A}^{\mathrm{F}} \cup \mathcal{A}^{\mathrm{FL}}_{\mathrm{SN}}$

$\mathcal{R}^{\mathrm{F}} \leftarrow \mathcal{R}^{\mathrm{F}} \cup \mathcal{R}^{\mathrm{F}}_{\mathrm{SN}}$

$\mathcal{B}^{\mathrm{F}}_{\mathrm{N}} \leftarrow \mathcal{B}^{\mathrm{F}}_{\mathrm{N}} \cup \mathcal{B}^{\mathrm{F}}_{\mathrm{SN}}$

$\mathcal{C} \leftarrow \mathcal{C} \cup \{c^{\mathrm{U}}\}$

ukonči pro všechny

pro všechny $a^{\mathrm{RL}} \in \mathcal{A}^{\mathrm{RL}}$ **proveď**

$(\{r^{\mathrm{R}}_0\}) \leftarrow \texttt{SestavitMnozinuVazebIncidujicichSObjektem}(q_0(a^{\mathrm{RN}}), s^{\mathrm{R}})$

$(\{r^{\mathrm{R}}_1\}) \leftarrow \texttt{SestavitMnozinuVazebIncidujicichSObjektem}(q_1(a^{\mathrm{RN}}), s^{\mathrm{R}})$

$(\tilde{s}^{\mathrm{FL}}_{\mathrm{SL}}, c^{\mathrm{U}}) \leftarrow \texttt{ParalelneDezintegrovatPrvek}(a^{\mathrm{RL}}, n, l^{\mathrm{F}}, \sigma_{\mathrm{K}}(a^{\mathrm{RL}}), \chi_{\mathrm{C}})$

$\left(\mathcal{A}^{\mathrm{FL}}_{\mathrm{SL}}, \tilde{\mathcal{R}}^{\mathrm{F}}_{\mathrm{SL}}, \mathcal{B}^{\mathrm{F}}_{\mathrm{SL}}\right) \leftarrow \tilde{s}^{\mathrm{FL}}_{\mathrm{SL}}$

pro všechny $a^{\mathrm{FL}}_{\mathrm{SL}} \in \mathcal{A}^{\mathrm{FL}}_{\mathrm{SL}}$ **proveď**

$(\{b^{\mathrm{F}}_0\}) \leftarrow \texttt{SestavitMnozinuKonektoruIncidujicichSObjektem}(q_0(a^{\mathrm{FL}}_{\mathrm{SL}}), \tilde{s}^{\mathrm{FL}}_{\mathrm{SL}})$

$(\{b^{\mathrm{F}}_1\}) \leftarrow \texttt{SestavitMnozinuKonektoruIncidujicichSObjektem}(q_1(a^{\mathrm{FL}}_{\mathrm{SL}}), \tilde{s}^{\mathrm{FL}}_{\mathrm{SL}})$

$(b^{\mathrm{F}}_0) \leftarrow \texttt{PriraditHodnotuVelicinyObjektu}(\zeta, \iota(r^{\mathrm{R}}_0), b^{\mathrm{F}}_0)$

$(b^{\mathrm{F}}_1) \leftarrow \texttt{PriraditHodnotuVelicinyObjektu}(\zeta, \iota(r^{\mathrm{R}}_1), b^{\mathrm{F}}_1)$

ukonči pro všechny

$(\mathcal{P}) \leftarrow \texttt{SestavitMnozinuPrirazeniKonektoru}(\mathcal{B}^{\mathrm{F}}_{\mathrm{SL}}, \mathcal{B}^{\mathrm{F}}_{\mathrm{N}}, \varphi_{\mathrm{S}})$

pro všechny $p \in \mathcal{P}$ **proveď**

$(b^{\mathrm{F}}_{\mathrm{L}}, b^{\mathrm{F}}_{\mathrm{N}}) \leftarrow p$

$(\{q_{\mathrm{N}}\}) \leftarrow \texttt{SestavitMnozinuPoluIncidujicichSObjektem}(b^{\mathrm{F}}_{\mathrm{N}},)$

$(\{q_{\mathrm{L}}\}) \leftarrow \texttt{SestavitMnozinuPoluIncidujicichSObjektem}(b^{\mathrm{F}}_{\mathrm{L}},)$

$(r^{\mathrm{F}}) \leftarrow \texttt{VytvoritVazbuMeziPolyPrvku}(q_{\mathrm{N}}, q_{\mathrm{L}}, 1, n, l^{\mathrm{G}})$

$\mathcal{R}^{\mathrm{F}} \leftarrow \mathcal{R}^{\mathrm{F}} \cup \{r^{\mathrm{F}}\}$

$\left(\hat{\mathcal{R}}^{\mathrm{F}}_{\mathrm{P}}\right) \leftarrow \texttt{SestavitMnozinuVazebIncidujicichSObjekty}(\{b^{\mathrm{F}}_{\mathrm{N}}, b^{\mathrm{F}}_{\mathrm{L}}\})$

$\texttt{OdstranitObjekty}\left(\hat{\mathcal{R}}^{\mathrm{F}}_{\mathrm{P}}\right)$

$\texttt{OdstranitObjekty}(\{b^{\mathrm{F}}_{\mathrm{N}}, b^{\mathrm{F}}_{\mathrm{L}}\})$

pro všechny

$\mathcal{A}^{\mathrm{F}} \leftarrow \mathcal{A}^{\mathrm{F}} \cup \mathcal{A}^{\mathrm{FL}}_{\mathrm{SL}}$

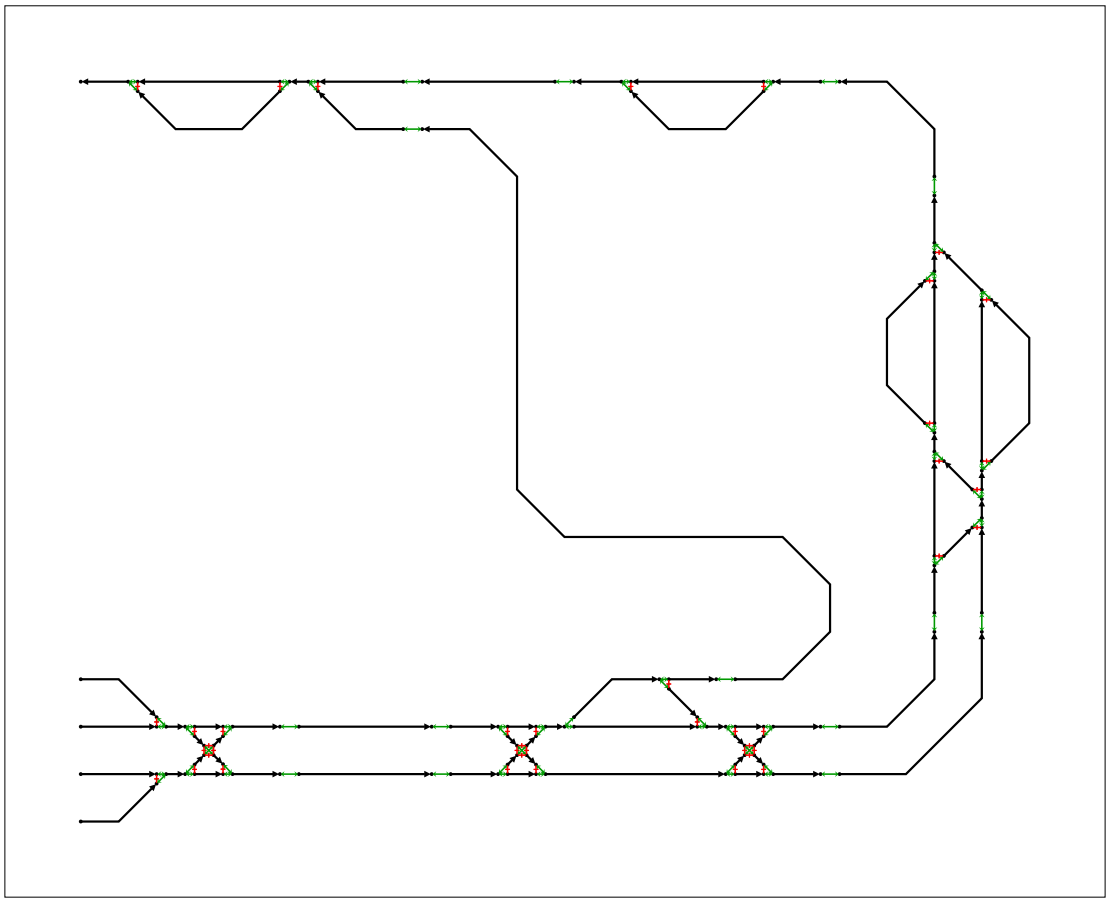
$\mathcal{C} \leftarrow \mathcal{C} \cup \{c^{\mathrm{U}}\}$

ukonči pro všechny

$s^{\mathrm{F}} \leftarrow (\mathcal{A}^{\mathrm{F}}, \mathcal{R}^{\mathrm{F}})$

vrať: $(s^{\mathrm{F}}, \mathcal{C})$

Příklad možné referenční struktury s^{R}, na níž může být aplikován navržený dezintegrační algoritmus t_{RF}, je uveden na obrázku 7.1. V případě navrženého dezintegračního algoritmu t_{RG} závisí průběh integračního procesu na hodnotách několika různých veličin, jimiž jsou posány neliniové a liniové prvky a také na vstupech zpracovatele. Za předpokladu neliniových prvků popsaných hodnotami $\sigma_{\mathrm{K}}(\mathtt{AB}) = \sigma_{\mathrm{K}}(\mathtt{BC}) = \sigma_{\mathrm{K}}(\mathtt{CD}) = 2$ a $\sigma_{\mathrm{K}}(\mathtt{DE}) = \sigma_{\mathrm{K}}(\mathtt{BE}) = 1$ a výběru z množiny předdefinovaných rozšířených vzorů vnitřních struktur můžeme získat strukturu s^{F} např. v podobě, v jaké je vyjádřená na obrázku 7.2.



Obrázek 7.2. Příklad méně podrobné struktury, která může vzniknout jako výstup navrženého dezintegračního algoritmu.

Kapitola 8
Grafický editor VMŽI

V rámci své práce na výzkumných projektech autor této práce vytvořil a stále rozvíjí skript pracující v prostředí softwarové aplikace počítačem podporovaného projektování umožňující pracovat s vizualizovanými daty VMŽI, který je v současné době určen zejména pro naplňování databází postavených na struktuře VMŽI daty. Některé rysy a funkcionality, které díky tomuto skriptu získává grafický editor zejména ve vztahu k objektům VMŽI, budou popsány níže.

8.1 Reprezentace objektů VMŽI grafickými objekty

Za účelem konzistentního naplňování struktury VMŽI daty a pro potřeby jejich dalšího zpracování je možné v softwarovém prostředí grafického editoru vytvářet datové objekty společně s jejich grafickými obrazy, reflektujícími geometrické aspekty reálných objektů modelem popisovaných. Vizualizace objektů VMŽI v tomto prostředí prostřednictvím grafických (též geometrických) objektů (dále GO) umožňuje mimo jiné podchytit prostorové souvislosti mezi těmito objekty, a tím je reálným objektům, které reprezentují, přiblížit. Touto cestou je tvůrci datového popisu poskytována cenná zpětná vazbu o tom, zda jsou zadaná nebo vygenerovaná data, jimiž je model naplňován, správná.

Na základě uvedených prostorových souvislostí zjistitelných v prostředí grafického editoru, je v závislosti na funkcích, jež grafický editor podporuje, dále možné automaticky nebo za spoluúčasti tvůrce datového popisu získávat další potřebná data vstupující do VMŽI. Tato data mohou do VMŽI vstupovat buď přímo jako hodnoty atributů instancí jednotlivých tříd, nebo mohou být k výpočtu hodnot těchto atributů využita.

8.2 Přiřazení datových položek grafickým objektům

Grafický editor pro práci s daty VMŽI umožňuje každou instanci libovolné neabstraktní třídy VMŽI (datový objekt) vyjádřit jako GO popsaný hodnotami jejích atributů. Každý datový objekt VMŽI v prostředí grafického editoru je vyjádřen prostřednictvím právě jednoho GO. Struktura VMŽI je koncipována tak, že každý datový objekt VMŽI je popsán pomocí jednoho nebo několika záznamů různých tabulek, vzájemně propojených na základě hodnoty atributu `id`. Každý z těchto záznamů se přitom skládá z jedné nebo několika datových položek. Každá datová položka v rámci VMŽI je tedy jednoznačně identifikovatelná pomocí datového objektu, k němuž přísluší, názvu tabulky, jíž přísluší záznam, jehož je datová položka součástí, a názvu atributu, jehož hodnotu datová položka vyjadřuje.

Za předpokladu respektování pravidla, že na základě hodnoty atributu `id` mohou být propojeny jen ty tabulky, žádná dvojice z nichž nemá jiný společný atribut, než je právě atribut `id` (což je pravidlo respektované jak v rámci aktuální verze jádra VMŽI, tak v rámci všech v současnosti existujících rozšíření modelu), je možné se při jednoznačné

identifikaci datové položky obejít i bez znalosti názvu tabulky. V zájmu snazšího přenosu dat mezi grafickým editorem a databází založené na struktuře VMŽI však jako jednu ze složek identifikace datové položky VMŽI název tabulky také vyžadujme.

Grafický editor umožňuje k jednotlivým geometrickým objektům reprezentujícím datové objekty VMŽI v něm vytvářeným připojit data strukturovaná následujícím způsobem:

[`tabulka`; `atribut`; `hodnotaAtributu`],

přičemž `tabulka` reprezentuje název tabulky, proměnná `atribut` název atributu a proměnná `hodnotaAtributu` jeho hodnotu. Takto strukturovaných datových položek je ke každému GO možné připojit libovolné množství.

8.3 Grafická podoba grafických objektů

Do jaké míry grafická podoba GO vystihuje charakter reálného objektu, který je předobrazem datového objektu, jejž příslušný GO reprezentuje, a hodnoty atributů tomuto objektu přiřazených, je do značné míry závislé na třídě, jíž je daný datový objekt instancí, ale také na konkrétním způsobu implementace v prostředí grafického editoru. Ta může pro některé třídy mimo jiné zohledňovat např. skutečnost, na jaké síťové úrovni se pracuje, čili jakými hodnotami atributů je příslušná síťová úroveň popsána.

V případě, že má způsob grafické reprezentace instancí určité třídy (např. i v kontextu určité síťové úrovně) význam pro odvození hodnot atributů nebo grafické reprezentace jiných instancí, je třeba tuto skutečnost při návrhu grafické reprezentace instancí této třídy zohlednit. V jiných případech má způsob grafické reprezentace instancí formou GO význam pouze z hlediska vizualizace dat. I to však může být poměrně významnou pomůckou pro tvůrce datového popisu.

Instance tříd VMŽI, u nichž nemá pokročilejší způsob grafické vizualizace opodstatnění, je možné formou GO reprezentovat na vysoce symbolické úrovni, např. v podobě bodu o stanovených souřadnicích v rámci prostoru grafického editoru. Hodnoty těchto souřadnic v takovýchto případech nemusejí nutně souviset s hodnotami jejich atributů ani jinými vlastnostmi. To se týká zejména o instance tříd globálního charakteru, příp. některých asociačních tříd.

Instance některých jiných tříd v rámci specifických rozšíření VMŽI naopak reprezentují vysoce konkrétní objekty reálného světa, u nichž má jejich umístění v prostoru zásadní význam. Při práci s daty VMŽI týkajícími se prostorových aspektů modelovaných objektů reálného světa lze využít toho, že grafický editor umožňuje efektivně pracovat se souřadnicovými systémy. Toho může být využito např. při přiřazování hodnot souřadnic jednotlivých geobodům, příp. při jejich transformaci.

8.4 Systémové objekty

Pro potřeby správy metadat týkajících se konkrétní instance struktury VMŽI naplňované daty v prostředí grafického editoru jsou dále zavedeny systémové objekty. Systémové objekty jsou datové objekty reprezentované speciálními GO zakládanými v prostředí grafického editoru před vlastním započetím tvorby datového popisu konkrétní instance VMŽI, nesoucí atributy, jejichž hodnoty vyjadřují globální nastavení pracovního prostředí a mohou mít vliv na chování editoru popisu železniční infrastruktury pracujícího v rámci prostředí grafického editoru.

Na rozdíl od GO reprezentujících jednotlivé instance tříd VMŽI nemají GO reprezentující systémové objekty předobraz v objektech VMŽI. Jsou vytvářeny výhradně za účelem uložení nastavených hodnot systémových proměnných při práci v grafickém editoru. GO reprezentujícím systémové objekty je možné přiřadit data strukturovaná stejným způsobem jako GO reprezentujícím objekty VMŽI. Názvy tabulek, atributů a přípustné hodnoty atributů uplatňující se při popisu těchto objektů jsou však zavedeny pouze vnitřně pro účely grafického editoru a nikterak nefigurují ve struktuře VMŽI. Obdobně jako na objekty VMŽI můžeme na systémové objekty nahlížet jako na datové objekty, které jsou instancemi svých tříd.

8.5 Vrstvy reprezentující třídy VMŽI

Grafický editor umožňuje pracovat s vrstvami uspořadatelnými do stromové struktury, do nichž mohou být jednotlivé GO, tedy i ty reprezentující jednotlivé instance tříd VMŽI, zařazeny. Jelikož VMŽI využívá pouze jednonásobné dědičnosti, je možné v prostředí grafického editoru uvažovat s takovým návrhem stromové struktury vrstev, který z hlediska dědičnosti reflektuje strukturu tříd VMŽI.

Každá třída VMŽI v rámci grafického editoru reprezentována vrstvou, jejíž název je shodný s názvem této třídy (příp. existuje vzájemně jednoznačné zobrazení množiny názvů tříd VMŽI do množiny názvů jim odpovídacích vrstev v prostředí grafického editoru založených) a jako podvrstvy obsahuje vrstvy reprezentující všechny třídy, které jsou jejími specializacemi. Každý GO vyskytující se v prostředí grafického editoru zařazen do vrstvy s názvem odpovídajícím názvu třídy, jejíž instance je příslušným GO reprezentována.

Vrcholové struktury vrstev grafického editoru respektuje rozdělení vrstev reprezentujících třídy VMŽI na vrstvy primárních objektů, na vrstvy sekundárních objektů a na vrstvy sekundárních objektů. Výhodou zařazení GO do výše uvedeným způsobem strukturalizovaných vrstev grafického editoru je zejména možnost s těmito objekty hromadně pracovat, a to na základě příslušnosti objektů VMŽI, jejichž jsou jednotlivé GO obrazem, příp. též systémových objektů, k třídám. Nemusí se přitom nutně nutně jednat o třídy terminální, jichž jsou datové objekty instancemi a do vrstev jimž odpovídajícím jsou GO, které jsou jejich obrazem, zařazeny. To samé se týká také vrstev odpovídajících abstraktním třídám, které zastřešují objekty všech neabstraktních tříd, které jsou jejich specializacemi.

8.6 Systém topologického popisu sítě ve VMŽI

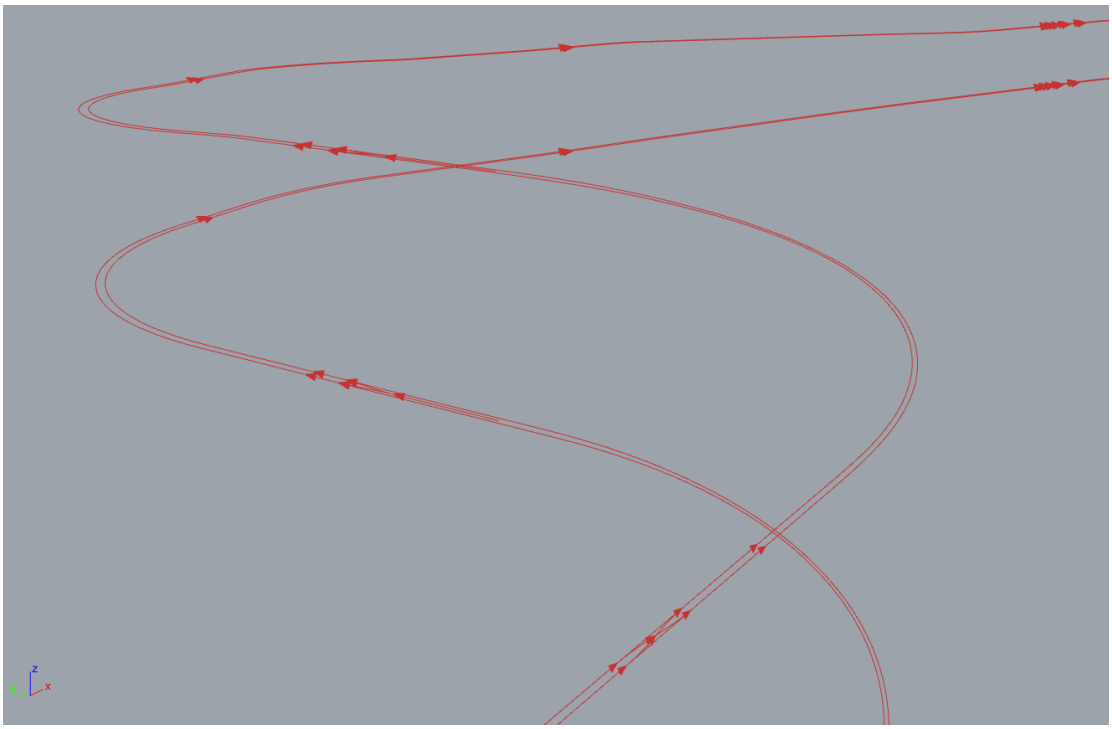
Systém topologického popisu sítě $\mathbb{T}$, který byl obecně popsán v kapitole 3, byl navrhován do značné míry s přihlédnutím k přístupu, kterým je topologický popis železniční infrastruktury řešen v rámci RTM a VMŽI. Objekty a další aspekty popisu systému $\mathbb{T}$ a jeho okolí uvažované při specifikaci tohoto systému je možné mapovat na prostředky využívané pro konstrukci VMŽI níže uvedeným způsobem:

- obecný objekt objekt o – ve vybraných případech instance některé třídy primárních objektů
- veličina $m(\mathcal{O})$ – ve vybraných případech atribut zavedený pro některou třídu objektů, ne všechny veličiny definované v rámci obecného popisu systému $\mathbb{T}$ a jeho okolí jsou reprezentovány konkrétním atributem VMŽI, je však možné zavést specifické rozšíření jádra VMŽI, které tento požadavek zohledňují

- hodnota veličiny $v(m, o)$ – ve vybraných případech hodnota atributu objektu, ne všechny hodnoty veličin definované v rámci obecného popisu systému $\mathbb{T}$ a jeho okolí jsou reprezentovány konkrétním hodnotou atributu VMŽI, je však možné zavést specifické rozšíření jádra VMŽI, které tento požadavek zohledňují
- síť – instance třídy `Network`, při popisu systému $\mathbb{T}$ je uvažována jedna konkrétní síť, v rámci VMŽI mohou být založeno více sítí, např. pro reprezentaci okolí systému $\mathbb{T}$
- rozlišovací úroveň l – instance třídy `LevelNetwork`, ačkoliv oproti RTM byly v rámci VMŽI zavedeny nové atributy umožňující popsat instance této třídy, možnosti vyjádření míry podrobnosti rozlišovací úrovně jsou stále omezené
- neliniový prvek a^{EN} systému $\mathbb{T}$ figurující na rozlišovací úrovni l^{E} – instance třídy `NonLinearElement` přiřazená instanci třídy `Network` reprezentující síť n a instanci třídy `LevelNetwork` reprezentující rozlišovací úroveň l^{E}
- liniový prvek systému a^{EL} systému $\mathbb{T}$ figurující na rozlišovací úrovni l^{E} – instance třídy `LinearElement` přiřazená instanci třídy `Network` reprezentující síť n a instanci třídy `LevelNetwork` reprezentující rozlišovací úroveň l^{E}
- neliniový prvek u^{EN} topologického okolí systému $\mathbb{T}$ figurující na rozlišovací úrovni l^{E} – instance třídy `NonLinearElement` bez přiřazení instanci třídy `Network` reprezentující síť n a přiřazená instanci třídy `LevelNetwork` reprezentující rozlišovací úroveň l^{E}
- liniový prvek u^{EL} topologického okolí systému $\mathbb{T}$ figurující na rozlišovací úrovni l^{E} – instance třídy `LinearElement` bez přiřazení instanci třídy `Network` reprezentující síť n a přiřazená instanci třídy `LevelNetwork` reprezentující rozlišovací úroveň l^{E}
- prvek e^{E} netopologického okolí sytému $\mathbb{T}$ figurující na rozlišovací úrovni l^{E} – instance některé ze specializací třídy `NetEntity`, tedy třídy zavedené v rámci specifického rozšíření VMŽI, přiřazená instanci třídy `Network` reprezentující síť n a instanci třídy `LevelNetwork` reprezentující rozlišovací úroveň l^{E}
- konektor b^{E} figurující na rozlišovací úrovni l^{E} – jedná se o nově zavedený koncept nad rámec RTM i VMŽI, který bude do VMŽI doplnit výhledově, aktuálně může být reprezentován jako instance třídy `NonLinearElement` nebo `LinearElement` přiřazená speciálně založené instanci třídy `Network` reprezentující síť konektorů a instanci třídy `LevelNetwork` reprezentující rozlišovací úroveň l^{E}
- pól prvku $q(a)$ – v rámci RTM a VMŽI není reprezentován jako samostatný objekt, je vyjádřen implicitně v rámci jednotlivých instancí třídy `PositionedRelation` navázaných na příslušnou instanci třídy `NonLinearElement` nebo `LinearElement` reprezentující prvek a prostřednictvím kombinací hodnot jejich atributů `id_Positioning-NetElement_A` a `positionOnA`, resp. `id_PositioningNetElement_B` a `positionOnB`
- vazba r^{E} systému $\mathbb{T}$ figurující na rozlišovací úrovni l^{E} – instance třídy `Positioned-Relation` přiřazená instanci třídy `Network` reprezentující síť n a instanci třídy `LevelNetwork` reprezentující rozlišovací úroveň l^{E}
- struktura s^{E} systému $\mathbb{T}$ vyjádřená na rozlišovací úrovni l^{E} – soubor všech instancí tříd `NonLinearElement`, `NonLinearElement` a `PositionedRelation` přiřazených instanci třídy `Network` reprezentující síť n a instanci třídy `LevelNetwork` reprezentující rozlišovací úroveň l^{E}
- neuspořádaná kolekce c^{U} – instance třídy `UnorderedCollection`
- uspořádaná kolekce c^{U} – instance třídy `OrderedCollection`

8.7 Operace prováděné nad daty grafického editoru

Grafický editor VMŽI umožňuje provádět širokou škálu operací s datovými objekty vázanými na jednotlivé GO. Do editačního skriptu byly mimo jiné začleněny i mnohé z operací představených v rámci této práce, jichž je možné využít ke konstruování transformačních algoritmů. Jednou z funkcionalit, kterou editor nabízí, je právě ucelená transformace všech síťových objektů přiřazených jedné síťové úrovni na síťové objekty jiné síťové úrovně, což odpovídá transformaci struktury systému topologického popisu sítě $\mathbb{T}$ z jedné rozlišovací úrovně na jinou. Každá tato transformace probíhá mezi dvěma danými síťovými úrovněmi, na jejichž popisu pomocí hodnot atributů závisí příslušný transformační algoritmus.



Obrázek 8.1. Příklad liniových síťových prvků reperezentujících kolejové trasy v prostředí grafického editoru po transformaci z roviny xy do prostoru xyz.

Mezi tyto algoritmy byly začleněny i některé ucelené transformační algoritmy navržené v souladu s metodiku představenou v rámci této práce. Mezi tyto algoritmy patří integrační algoritmus pracující na podobném principu jako algoritmus představený v oddíle 6.2.2. Jeho referenční rozlišovací úroveň je však ještě poněkud podrobnější, neboť výhybky zde nejsou modelovány s využitím neliniových síťových prvků, ale s využitím liniových síťových tak, že každá větev výhybky je reprezentována jedním liniovým síťovým prvek. Tento algoritmus publikoval autor ve vědeckém článku [18]. V rámci grafického editoru je pomocí tohoto algoritmu možné transformovat topologický a topografický popis infrastruktury vyjádřený na síťové úrovni o hodnotách atributů `descriptionLevel micro`, `dimension xy` a `representation realistic` na topologický popis infrastruktury vyjádřený na síťové úrovni o hodnotách atributů `descriptionLevel macro`, `dimension xy` a `representation schematic`. Jiný transformační algoritmus, jehož principy budou také publikovány, [19] pracuje navíc také s detailním popisem směrových a výškových poměrů, jimiž jsou jednotlivé liniové síťové prvky popsány

s využitím síťových entit rozšiřujícího bloku `Geometry` a s přiřazenými geometrickými souřadnicemi souřadnicového systému grafického editoru. Transformace probíhá mezi síťovou úrovní o hodnotách atributů `descriptionLevel micro`, `dimension xy` a `representation realistic` na síťovou úroveň o hodnotách atributů `descriptionLevel micro`, `dimension xyz` a `representation realistic`. Jedná se o transformaci zejména topografického charakteru, při níž jsou jednotlivé liniové síťové prvky transformovány z dvourozměrné podoby do trojrozměrné podoby, což se projevuje mimo jiné i změnou hodnoty jejich atributu `length`. Příklad grafické vizualizace těchto liniových síťových prvků vyjádřených na obou zmíněných síťových úrovních je uveden na obrázku 8.1. V prostředí grafického editoru VMŽI mohou být tedy v souladu s představenou metodikou navrhované transformační algoritmy uplatňovány v praxi a verifikovány užitím.

Kapitola 9
Závěr

V rámci této práce byla navržena metodika integrace a dezintegrace prvků topologického popisu železniční infrastruktury figurujících na jednotlivých zavedených rozlišovacích úrovních dané sítě n. Tento systém byl na obecné úrovni detailně popsán včetně zavedení způsobu označování objektů a terminologie, kterou bylo možné následně uplatnit také při návrhu operací, jež jsou stavebními kameny transformačních algoritmů. Byl navržen metodický postup, jak tyto algoritmy sloužící k integraci a dezintegraci ucelených struktur tohoto systému vyjádřených na různých rozlišovacích úrovních navrhovat a v souladu s tímto postupem byly demonstrovány také konkrétní příklady návrhu integračního a dezintegračního algoritmu. Zatímco integrační algoritmus je možné navrhnout tak, aby pracoval zcela automaticky, u většiny dezintegračních algoritmů lze očekávat nutnou spoluúčast zpracovatele, který musí v kritický okamžik doplnit chybějící informaci.

Obecně navrženou metodiku je možné aplikovat v praxi, a to například v prostředí grafického editoru Víceúčelového modelu železniční infrastruktury, ve kterém mohou být navrženě algoritmy zároveň verifikovány užitím. Uplatnění v praxi lze očekávat i u dalších dílčích konceptů v této práci nově zavedených a popsaných. Tím je například koncept konektoru, u nějž jsou v současné době sledovány možnosti začlenění do struktury výměnného formátu railML 3 [13] pro účely rozdělování a spojování datových souborů na topologické úrovni.

Literatura

[1] HLUBUČEK, Adam. Význam popisu infrastruktury pro inteligentní dopravní systémy na železnici. *Vědeckotechnický sborník ČD*. 2016, 42, ISSN 1214-9047.

[2] LIESKOVSKÝ, Aleš, MYSLIVEC, Ivo. ETCS a ATO – poprvé společně. *Reportér AŽD*. Praha. 2011, 2011(3), s. 46 – 48.

[3] ČÍHAL, Robert. *Prostorový popis železniční sítě a data objektů železniční infrastruktury v IS SŽDC* [přednáška]. Brno, KMP CONSULT, 15. 4. 2016.

[4] RAŠKA, *Jakub. Koncepce rozvoje IS SŽ ve směru k DTM, DTMŽ a BIM* [přednáška]. Praha, Prostorový popis železniční sítě ve správě SŽ, 28. 3. 2023.

[5] MAGNIEN, Airy. *RailTopoModel* [přednáška]. Paris, 7th UIC RailTopoModel and 30th railML konference, 3. 11. 2016.

[6] UIC. *Feasibility study UIC RailTopoModel and Data Exchange Format* [soubor pdf]. 2013.

[7] UIC. *UIC International Railway Standard IRS 30100 RailTopoModel* [soubor pdf]. 2016.

[8] UIC. *RailTopoModel v1.0* [soubory pdf]. 2016.

[9] BISCHOF Stefan, SCHENNER, Gottfried. Rail Topology Ontology: A Rail Infrastructure Base Ontology. The Semantic Web – ISWC 2021. ISWC 2021. *Lecture Notes in Computer Science* , 2021, vol 12922. Springer, Cham.
Dostupné z: https://doi.org/10.1007/978-3-030-88361-4_35

[10] 10. railML.org. railML.org [online] Dresden: railML.org [cit. 2023-01-31].
Dostupné z: https://www.railml.org/

[11] railML.org. *railML Schema Version 3.2* [archiv zip]. 2021.

[12] RAHMIG, Christian. *Bericht der Infrastruktur-Arbeitsgruppen bei railML 2.5 und 3.2* . 41st railML Conference. Dresden: railML.org, 26. 4. 2022

[13] RAHMIG, Christian, HLUBUČEK Adam, ZHUCHYI Larysa, KOLMORGEN, Vasco Paul. Splitting and Connecting of Railway Infrastructure Network Representations: a railML Case Study [konferenční příspěvek v recenzním řízení]. 2023

[14] LESO, Martin, KOUTECKÝ, Petr., KAMENICKÝ, Dušan, HLUBUČEK, Adam. Dopravní sál FD – železniční laboratoř pro výuku i výzkum. *Vědeckotechnický sborník Správy železnic*. 2019, 2019(1) ISSN 2694-9172.

[15] HLUBUČEK, Adam. Digital Track Map for the VEXA expert system. *Acta Polytechnica CTU Proceedings*. 2022, 35, s. 8 – 13.
Dostupné z: https://doi.org/10.14311/APP.2022.35.0008

[16] HLUBUČEK, Adam. RailTopoModel and railML 3 in Overall Context. *Acta Polytechnica CTU Proceedings*. 2017, 11, s. 16 – 21.
Dostupné z: https://doi.org/10.14311/APP.2017.11.0016.

[17] HLUBUČEK, Adam. Possibilities of High-Speed Railway Turnout Data Description. *Acta Polytechnica CTU Proceedings*. 2021, 31, s. 18 – 26.
Dostupné z: https://doi.org/10.14311/APP.2021.31.0018.

[18] HLUBUČEK, Adam. Integration of Railway Infrastructure Topological Description Elements from the MicroL2 to the MacroN0,L0 Level of Detail. *Neural Network World*, 2023, s. 19 –34.
Dostupné z: https://doi.org/10.14311/NNW.2023.33.002

[19] HLUBUČEK, Adam. *Determining 3D Coordinates Based on the Track Geometry Descriprion* [konferenční příspěvek přijatý k publikaci]. Acta Polytechnica CTU Proceedings. 2023.

[20] HLUBUČEK, Adam. *Studie k disertační práci: Metodiky integrace a dezintegrace prvků topologického popisu železniční infrastruktury*, 2018.

Příloha A
Seznam zkratek

ATC Automatic train control
ATO Automatic train operation
AVV Automatické vedené vlaku
BIM Building information modeling
CAGI Česká asociace pro geoinformace
ČD České dráhy
ČSD Československé státní dráhy
ČVUT České vysoké učení technické v Praze
DTMŽ Digitální technická mapa železnice
ERA European Railway Agency
ETCS European Train Control System
ID identifikace
IRS International Railway Solution (původně Standard)
SQL Structured Query Language
GIS geografický informační systém
GO grafický (též geometrický) objekt
railML Railway Markup Language
RTM RailTopoModel
TSI Technické specifikace interoperability
UIC Union internationale des chemins de fer
UML Unified Modeling Language
VMŽI Víceúčelový model železniční infrastruktury
XML Extensible Markup Language