



Zadání bakalářské práce

Název:	ETCS – EVC – Implementace módů reversing, shunting a post trip
Student:	Matěj Malý
Vedoucí:	Ing. Jan Matoušek
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2023/2024

Pokyny pro vypracování

ETCS (European Train Control System) je jednotný celoevropský zabezpečovací systém. EVC (European Vital Computer) je centrální počítač vozidlové části systému. ETCS se v každém okamžiku nachází v určitém módu. Práce je součástí trenážeru vlakového zabezpečovacího zařízení vyvíjeného ve spolupráci s Fakultou dopravní ČVUT. Cílem této práce je implementace módů reversing, shunting a post trip pro komponentu EVC. Reversing je mód, do kterého se ETCS dostane v případě, že lokomotiva couvá, shunting je mód potřebný pro posouvání a post trip je mód, do kterého se systém dostane po nouzovém brždění až do zastavení vlaku a potvrzení strojvedoucím.

Pokyny pro vypracování:

1. Analyzujte současný stav projektu, zaměřte se na komponentu EVC.
2. Analyzujte chování systému v daných módech a přechody mezi módy podle SUBSET-026 v2.3.0.
3. Pomocí metod softwarového inženýrství navrhnete úpravy komponenty EVC řešící uvedené módy.
4. Implementujte a zdokumentujte navržené úpravy.
5. Úpravy v komponentě podrobte vhodným testům (jednotkové, integrační, akceptační).

Bakalářská práce

**ETCS – EVC –
IMPLEMENTACE MÓDŮ
REVERSING, SHUNTING
A POST TRIP**

Matěj Malý

Fakulta informačních technologií
Katedra softwarového inženýrství
Vedoucí: Ing. Jan Matoušek
11. května 2023

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2023 Matěj Malý. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení, je nezbytný souhlas autora.

Odkaz na tuto práci: Malý Matěj. *ETCS – EVC – Implementace módů reversing, shunting a post trip*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2023.

Obsah

Poděkování	ix
Prohlášení	x
Abstrakt	xi
Seznam zkratk	xii
1 Úvod	1
2 Cíl práce	3
3 Analýza	5
3.1 ERTMS	5
3.1.1 Popis ERTMS a jeho částí	5
3.1.2 Úrovně ETCS	6
3.1.3 Důvody pro zavedení ERTMS	7
3.2 Módy	7
3.2.1 Popis módů	8
3.2.2 Přechody mezi módy	10
3.3 Procedury	12
3.3.1 Procedura Shunting Initiated by Driver	12
3.3.2 Procedura End of Mission	12
3.3.3 Procedura Entry in Shunting	12
3.3.4 Procedura Train Trip	13
3.3.5 Procedura On-Sight	13
3.3.6 Procedura Start of Mission	13
3.3.7 Procedura Reversing	13
3.4 Dohled nad pohyby vlaku	14
3.4.1 Roll Away Protection	14
3.4.2 Reverse Movement Protection	14
3.4.3 Standstill supervision	14
3.5 Současný stav projektu trenažéru	15
3.5.1 Stav EVC	15
3.6 Funkční a nefunkční požadavky	16
3.6.1 Funkční požadavky	16
3.6.2 Nefunkční požadavky	17
4 Návrh	19
4.1 Procedury	19
4.2 Zprávy	31
4.2.1 Identifikátory zpráv	31
4.2.2 Zprávy posílané DMI	31
4.2.3 Zprávy přijímané od DMI	33

4.2.4	Zprávy odesílané RBC	34
4.2.5	Zprávy přijímané od RBC	35
4.2.6	Reprezentace zpráv v EVC	38
4.3	Pakety	41
4.3.1	Paket 42: Session Management	41
4.3.2	Paket 49: List of balises for SH Area	42
4.3.3	Paket 132: Danger for Shunting Information	42
4.3.4	Paket 138: Reversing area information	43
4.3.5	Paket 139: Reversing supervision information	43
4.3.6	Reprezentace paketů v EVC	44
4.4	Podmínky přechodu nepokryté procedurami	45
4.4.1	Komunikace s TIU	45
4.4.2	Komunikace s BTM	45
4.4.3	Exit Shunting	45
4.5	Služby	45
4.5.1	ShuntingService	45
4.5.2	ReversingService	45
4.5.3	EmergencyStopsService	46
4.5.4	ModeProfileService	46
4.5.5	MonitoringService	46
4.6	Přijímání zpráv	47
5	Implementace	49
5.1	Implementace zpráv	49
5.2	Implementace paketů	50
5.3	Implementace služeb	51
5.3.1	Šablona třídy AsyncProperty	52
5.4	Implementace zpracování zpráv	53
6	Testování	55
6.1	Testovací scénáře	55
6.1.1	Testovací zprávy	55
6.1.2	Navázání spojení	56
6.1.3	Podmínky přechodu 4 a 29	56
6.1.4	Zpracování paketů 138 a 139	57
6.1.5	Procedura Reversing	57
6.1.6	Procedura Shunting Initiated by Driver	58
6.1.7	Procedura End of Mission	59
6.1.8	Procedura Entry in Shunting	60
6.1.9	Procedura On-Sight	61
6.1.10	Procedura Train trip	62
6.1.11	Standstill supervision	64
6.1.12	Reverse movement protection	65
6.1.13	Roll away protection	65
6.1.14	Maximální vzdálenost v módu PT	67
6.1.15	Maximální vzdálenost v módu RV	68
6.1.16	Kontrola maximální rychlosti v módu RV	68
6.1.17	Kontrola maximální rychlosti v módu SH	69
6.1.18	Přijetí paketu 132: Danger for Shunting information	69
6.1.19	Exit from Shunting	69
6.1.20	Seznam balíz pro mód SH	70
6.2	Průběh a výsledky testování	71

7 Závěr	73
A Význam symbolů v modelech	75
B Změny v souborech	77
Obsah přiloženého archivu	83

Seznam obrázků

4.1	Popis průběhu procedury Shunting Initiated by Driver	20
4.2	Popis průběhu procedury Reversing	21
4.3	Popis části průběhu procedury Start of Mission	22
4.4	Popis průběhu procedury End of Mission	23
4.5	Popis průběhu procedury Train trip – část 1	24
4.6	Popis průběhu procedury Train trip – část 2	25
4.7	Popis průběhu procedury Train trip – část 3	26
4.8	Popis průběhu procedury Entry in Shunting – část 1	27
4.9	Popis průběhu procedury Entry in Shunting – část 2	28
4.10	Popis průběhu procedury On-Sight – část 1	29
4.11	Popis průběhu procedury On-Sight – část 2	30
4.12	Class diagram tříd - zprávy posílané DMI	39
4.13	Class diagram tříd - zprávy přijímané od DMI	39
4.14	Class diagram tříd reprezentujících zprávy pro komunikaci s RBC	40
4.15	Class diagram tříd reprezentujících pakety	44
6.1	Soubory pro testovací scénář 6.1.2	56
6.2	Soubory pro testovací scénář 6.1.3	56
6.3	Soubory pro testovací scénář 6.1.4	57
6.4	Soubory pro testovací scénář 6.1.5	57
6.5	Soubory pro testovací scénář 6.1.6.1	58
6.6	Soubory pro testovací scénář 6.1.6.2	58
6.7	Soubory pro testovací scénář 6.1.6.3	59
6.8	Soubory pro testovací scénář 6.1.7	59
6.9	Soubory pro testovací scénář 6.1.8.1	60
6.10	Soubory pro testovací scénář 6.1.8.2	61
6.11	Soubory pro testovací scénář 6.1.9.1	61
6.12	Soubory pro testovací scénář 6.1.9.2	62
6.13	Soubory pro testovací scénář 6.1.10	63
6.14	Soubory pro testovací scénář 6.1.10.2	63
6.15	Soubory pro testovací scénář 6.1.11.1	64
6.16	Soubory pro testovací scénář 6.1.11.2	64
6.17	Soubory pro testovací scénář 6.1.12	65
6.18	Soubory pro testovací scénář 6.1.13	66
6.19	Soubory pro testovací scénář 6.1.14	67
6.20	Soubory pro testovací scénář 6.1.15	68
6.21	Soubory pro testovací scénář 6.1.16	68
6.22	Soubory pro testovací scénář 6.1.17	69
6.23	Soubory pro testovací scénář 6.1.18	69
6.24	Soubory pro testovací scénář 6.1.19	69
6.25	Soubory pro testovací scénář 6.1.20.1	70
6.26	Soubory pro testovací scénář 6.1.20.2	70
6.27	Soubory pro testovací scénář 6.1.20.3	70

Seznam tabulek

3.1	Tabulka přechodů	10
4.1	Struktura zprávy 209: Mode change	31
4.2	Struktura zprávy 210: Waiting for SH request response	32
4.3	Struktura zprávy 211: Reversing possible	32
4.4	Struktura zprávy 212: Revoke brake command	32
4.5	BrakeCommand enum	32
4.6	Struktura zprávy 213: Waiting for emergency stop revocation	33
4.7	Struktura zprávy 214: Emergency stops revoked	33
4.8	Struktura zprávy 308: Shunting selected	33
4.9	Struktura zprávy 309: Mode change acknowledged	34
4.10	Struktura zprávy 310: Revoke brake command acknowledgement	34
4.11	Struktura zprávy 130: Request for Shunting	34
4.12	Struktura zprávy 147: Acknowledgement of Emergency Stop	35
4.13	Struktura zprávy 6: Recognition of exit from TRIP mode	35
4.14	Struktura zprávy 15: Conditional Emergency Stop	36
4.15	Struktura zprávy 16: Unconditional Emergency Stop	36
4.16	Struktura zprávy 18: Revocation of Emergency Stop	37
4.17	Struktura zprávy 27: SH Refused	37
4.18	Struktura zprávy 28: SH Authorised	38
4.19	Struktura paketu 42: Session Management	41
4.20	Struktura paketu 49: List of balises for SH Area	42
4.21	Struktura paketu 132: Danger for Shunting Information	42
4.22	Struktura paketu 138: Reversing area information	43
4.23	Struktura paketu 139: Reversing supervision information	43
A.1	Význam symbolů v modelech	75

Seznam výpisů kódu

5.1	Deklarace třídy ModeChangeAckMessage	49
5.2	Deklarace třídy SHAuthorisedMessage	50
5.3	Rozhraní IReversingDataService	51
5.4	Deklarace třídy ReversingDataService	51
5.5	Privátní proměnné třídy AsyncProperty	52
5.6	Implementace metody AsyncProperty::Invoke	52
5.7	Deklarace třídy ModeChangeAckMessageHandler	53
5.8	Konstruktor třídy ModeChangeAckMessageHandler	53

5.9	Metoda HandleMessageBody třídy ModeChangeAckMessageHandler	54
-----	--	----

Chtěl bych poděkovat především vedoucímu své práce, Ing. Janu Matouškovi, za veškerou pomoc, kterou mi vždy velmi ochotně poskytl. Dále pak panu doc. Ing. Martinovi Lesovi, Ph.D. a panu Bc. Vilémovi Pecnovi za jejich odborné rady. V neposlední řadě rodině a přátelům za jejich podporu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 11. května 2023

.....

Abstrakt

Tato bakalářská práce rozšiřuje funkcionalitu trenažéru ETCS, který vzniká ve spolupráci s Fakultou dopravní ČVUT v Praze. ETCS je jednotný celoevropský vlakový zabezpečovací systém. Práce se zabývá komponentou EVC, která je jádrem vozidlové části systému a má za úkol zpracovávat a vyhodnocovat data z ostatních komponent. Výsledkem práce je přidání módů reversing (mód pro couvání), shunting (mód pro posun vlaku) a post trip (mód, do kterého systém přechází po nouzovém zastavení vlaku) do komponenty EVC. Práce obsahuje návrh a popis implementace přechodů do těchto módů a funkcionalit v těchto módech. Dále vysvětluje jakým způsobem bylo řešení testováno. Díky výsledkům práce se rozšiřují možnosti scénářů, které trenažér může simulovat.

Klíčová slova vlakový zabezpečovací systém, vlaková doprava, ETCS, trenažér ETCS, European Vital Computer, módy ETCS, C++

Abstract

This bachelor thesis extends the functionality of the ETCS simulator, which is being developed in collaboration with the Faculty of Transportation Sciences at CTU in Prague. ETCS is a united European train protection system. The focus of the thesis is the EVC component, which is the core of the on-board subsystem and is responsible for collection and evaluation of data from other components. The outcome of this thesis is the addition of modes reversing (mode used for reversing movements), shunting (mode used for shunting movements) and post trip (mode entered after emergency braking is over) to EVC. The thesis includes the design and description of the implementation of mode transitions and functionality in these modes. Furthermore, it describes how the solution was tested. The outcome of this thesis extends the possible scenarios that can be run in the simulator.

Keywords train protection system, train transport, ETCS, ETCS simulator, European Vital Computer, ETCS modes, C++

Seznam zkratek

ATP	Automatic Train Protection
BTM	Balise Transmission Module
DMI	Driver Machine Interface
EoA	End of Authority
ERTMS	European Railway Traffic Management System
ETCS	European Train Control System
EVC	European Vital Computer
GSM-R	Global System for Mobile Communications – Railway
JRU	Juridical Recording Unit
LRBG	Last relevant balise group
MA	Movement Authority
ODO	Odometrie
RAP	Roll Away Protection
RBC	Radio Block Center
RMP	Reverse Movement Protection
STM	Specific Transmission Module
SvL	Supervised Location
TAF	Track Ahead Free
TIU	Train Interface Unit



Kapitola 1

Úvod

Tato práce je součástí projektu vyvíjeného ve spolupráci s Fakultou dopravní ČVUT v Praze, který má za cíl vytvořit trenažér vlakového zabezpečovacího systému ETCS (European Train Control System). ETCS je jednotný celoevropský zabezpečovací systém, který má zvýšit bezpečnost a plynulost dopravy na evropských železnicích. [1] Systém se člení na několik samostatných komponent. Práce se soustředí na komponentu EVC (European Vital Computer), což je centrální počítač vozidlové části systému. EVC má za úkol zpracovávat a vyhodnocovat data z ostatních komponent systému.

Výstupy této práce rozšíří současnou funkcionalitu trenažéru přidáním nových módů. Konkrétně se jedná o módy reversing, shunting a post trip. Díky tomu bude možné simulovat situace, do kterých se reálný systém může dostat, a tak strojvedoucího, který se na trenažéru bude zaškolovat, na takové situace dobře připravit.

Hotový trenažér se bude používat pro školení strojvedoucích. Některé situace, které při řízení vlaku, který je vybaven systémem ETCS, mohou nastat, není možné natrénovat jinak než s pomocí trenažéru. Strojvedoucí se nemůže plně a zároveň bezpečně seznámit s fungováním systému na reálném stroji. Proto trenažér, který bude věrně simulovat fungování systému ETCS, je nutnou pomůckou pro zaškolení strojvedoucích.

Železnice v České republice jsou v současné době v postupném procesu zavádění ETCS. Problematika je proto velmi aktuální. [2]

Hlavním důvodem pro výběr tématu byla moje zkušenost s projektem. Na projektu jsem pracoval v rámci předmětů *Softwarový týmový projekt 1* a *Softwarový týmový projekt 2*. Chtěl jsem se na něm podílet i dále, protože se týká velmi aktuálního tématu a výsledný trenažér bude mít reálné využití.



Kapitola 2

Cíl práce

Cílem této práce je implementovat módy reversing, shunting a post trip pro komponentu EVC trenážeru ETCS. Pro správnou implementaci těchto módů je nutné v první řadě analyzovat současný stav trenážeru se zaměřením na komponentu EVC a popsat jakým způsobem ETCS přechází do daných módů a funkcionalitu v těchto módech. Dalším dílčím cílem je navrhnout, jak co nejvhodněji přidat přechody mezi módy a funkcionalitu do projektu EVC. Dále navrhnout zprávy, které se budou posílat mezi jednotlivými komponentami ETCS, a to, jak EVC bude na tyto zprávy reagovat. Následuje samotná implementace, na kterou navazuje testování.

Kapitola 3

Analýza

V této kapitole bude blíže popsán systém ERTMS. Bude vysvětlena jeho funkcionalita, důležitost a výhody jeho zavedení pro vlakovou dopravu na národní i nadnárodní úrovni. Budou popsány jednotlivé části, ze kterých se systém skládá, s důrazem na součást ETCS. Budou popsány komponenty ETCS s důrazem na komponentu EVC, jednotlivé módy, ve kterých se ETCS může nacházet, a přechody mezi jednotlivými módy. Bude popsán současný stav projektu a závěrem kapitoly budou vyjmenované funkční a nefunkční požadavky.

3.1 ERTMS

3.1.1 Popis ERTMS a jeho částí

ERTMS (European Railway Traffic Management System) je evropský systém řízení železniční dopravy, který se skládá ze dvou hlavních prvků: ETCS a GSM-R. ETCS (European Train Control System) je evropský vlakový zabezpečovací systém. Je to ATP (Automatic Train Protection) systém, který by měl nahradit všechny národní ATP systémy v Evropě. [3] GSM-R (Global System for Mobile Communications – Railway) je systém rádiové komunikace, která probíhá mezi vlaky a traťovou částí. Jedná se o hlasovou komunikaci nebo přenos dat.[4]

Celý systém můžeme rozdělit na dva subsystémy: traťovou a vozidlovou část.

3.1.1.1 Traťová část

3.1.1.1.1 Eurobalízy Eurobalíza je zařízení nainstalované v kolejišti, které obsahuje data týkající se infrastruktury tratě. Mohou to být například omezení rychlosti nebo sklon trati. V některých případech je možné data v těchto zařízeních měnit, jindy nikoliv. Vlak získá data nahraná na balíze přejetím nad touto balízou pomocí BTM, které má na palubě. Eurobalízy jsou pasivní zařízení, což znamená, že je není nutné napájet proudem. Při přejezdu vlaku jsou napájeny pomocí BTM. [5]

3.1.1.1.2 Radio Block Center Radio Block Center (RBC) je centralizovaná zabezpečovací jednotka, která se používá v 2. a 3. úrovni ETCS. RBC dostává od vlaku informace o poloze a odesílá vlaku povolení k jízdě a další data nutná k pohybu vlaku. Celá komunikace mezi RBC a vlakem probíhá pomocí GSM-R. [5]

3.1.1.2 Vozidlová část

3.1.1.2.1 European Vital Computer European Vital Computer (EVC) je jádrem vozidlové části ERTMS. Je součástí ATP systému a všechny ostatní součásti, které jsou přítomné ve vozidlové části, s EVC komunikují. [5]

3.1.1.2.2 Driver Machine Interface Driver Machine Interface (DMI) je rozhraní, které zprostředkovává komunikaci mezi řidičem a systémem. Většinou je realizované LCD dotykovým displejem. DMI umožňuje strojvedoucímu zadávat data do systému a zobrazuje mu výstupní data systému. [5]

3.1.1.2.3 Train Interface Unit Train Interface Unit (TIU) je rozhraní mezi vlakem a vlakovou částí ETCS. Pomocí TIU se například předává informace o pozici směrového ovladače. [5]

3.1.1.2.4 Juridical Recording Unit Juridical Recording Unit (JRU) shromažďuje důležitá data o průběhu jízdy, která mohou být později použita k analýze. JRU tedy funguje trochu jako taková černá skříňka. [5]

3.1.1.2.5 Balise Transmision Module Balise Transmision Module (BTM) je modul, který zpracovává signály přijaté z Eurobalíz pomocí antény. [5]

3.1.1.2.6 Odometrie Odometrie (ODO) se stará o výpočet ujeté vzdálenosti. Zároveň poskytuje informace o současné rychlosti a zrychlení. [5]

3.1.1.2.7 Specific Transmission Module Specific Transmission Module (STM) je zařízení, které slouží jako rozhraní mezi již existujícím národním zabezpečovacím systémem a ETCS. [5]

3.1.2 Úrovně ETCS

ETCS má pět různých úrovní, které společně s módem ETCS určují míru kontroly, kterou je systém schopen poskytnout. Tato úroveň je nastavena traťovou částí systému. Mezi jednotlivými úrovněmi je možné za určitých podmínek přecházet.

- **ETCS úroveň 0** se používá v situaci, kdy vlak vybaven systémem ETCS jede po trati, kde traťová část systému není instalována. Zároveň není na trati instalován ani národní systém.
- **ETCS úroveň STM** se používá v situaci, kdy vlak vybaven systémem ETCS jede po trati, která je vybavena národním systémem. S takovým systémem ETCS může komunikovat pomocí STM.
- **ETCS úroveň 1** se používá v situaci, kdy vlak vybaven systémem ETCS jede po trati, která je vybavena Eurobalízami.
- **ETCS úroveň 2** se používá v situaci, kdy vlak vybaven systémem ETCS jede po trati, která je vybavena Eurobalízami a kontrolována RBC. Kontrola volné trati a celistvosti vlaku je zajištěna traťovou částí systému.
- **ETCS úroveň 3** je podobná úrovni 2, ale kontrola pozice vlaku, volnosti trati a celistvosti vlaku je prováděna vozidlovou částí systému. [6, 2.1]

3.1.3 Důvody pro zavedení ERTMS

Hlavní důvod zavedení ERTMS je zvýšení bezpečnosti vlakového provozu. ERTMS je nejúčinnější systém řízení vlakového provozu na světě, který kromě bezpečnosti, přesnosti a spolehlivosti přináší i další výhody, jako je například snížení nákladů na provoz.

Dalším důležitým důvodem pro zavedení ERTMS je sjednocení vlakového zabezpečení v celé Evropě, které povede k zjednodušení mezinárodní přepravy. V tuto chvíli je v Evropě zavedeno více než 20 různých systémů, které mezi sebou nejsou propojené a neumí navzájem komunikovat. Proto je za současného stavu pro zajištění bezpečnosti nutné, aby vlaková souprava disponovala systémy všech zemí, kterými projíždí. To samozřejmě vede ke zvýšení nákladů na dopravu mezi jednotlivými zeměmi. [1]

Zavedení ERTMS s sebou přináší i další výhody. Díky ERTMS se zvyšuje propustnost tratí. Tato vyšší propustnost vede ke zvýšení kapacity na stejné infrastruktuře až o 40 %. ERTMS je prvním krokem k vyšším systémům automatizace. Tento systém vede ke zrychlení dopravy a umožňuje rychlosti až 500 km/h. [1]

Celkově by tedy zavedení ERTMS v Evropě mělo vést k zvýšení bezpečnosti, větší spolehlivosti, zvýšení propustnosti tratí a zjednodušení a zlevnění mezinárodní dopravy.

3.2 Módy

ETCS se během jízdy nachází v různých módech. Přejít do jiného módu může být vyvolán traťovou částí ETCS, případně samotným strojvedoucím. Strojvedoucí tak ale může učinit jen ve specifických případech. Aby ETCS mohlo přejít do určitého módu, musí infrastruktura tratí splňovat určité požadavky. Funkcionalita systému v módech se liší, tudíž je rozdílná i úroveň kontroly, kterou ETCS nabízí. [6, 2.3.1]

Každý mód má svoje jméno a zkratku uvedenou v závorkách. Ve verzi 2.3.0 jsou k dispozici následující módy:

- Full Supervision (FS)
- On Sight (OS)
- Staff Responsible (SR)
- Shunting (SH)
- Unfitted (UN)
- Sleeping (SL)
- Stand By (SB)
- Trip (TR)
- Post Trip (PT)
- System Failure (SF)
- Isolation (IS)
- No Power (NP)
- Non Leading (NL)
- STM European (SE)
- STM National (SN)
- Reversing (RV) [7, 4.3.2]

3.2.1 Popis módů

3.2.1.1 Full Supervision

Přechod do módu Full Supervision nemůže být iniciován strojvedoucím. Do tohoto módu se ETCS dostává automaticky v případě, že dostane potřebné povolení k jízdě a data popisující profil trati (Static Speed Profile a Gradient Profile). V tomto módu má ETCS veškerou zodpovědnost za bezpečnou jízdu. Tento mód je možné používat v úrovních 1, 2 a 3. [6, 2.3.2.1]

3.2.1.2 On Sight

Mód On Sight se používá ve chvíli, kdy vlak vjíždí do úseku, ve kterém je nějaký jiný vlak (příp. jiná překážka), nebo data o tom, jestli je úsek volný, nejsou k dispozici. Přechod do tohoto módu může být iniciován pouze traťovou částí systému. Velká část zodpovědnosti přechází v módu OS na strojvedoucího, tudíž přechod do tohoto módu musí být potvrzen strojvedoucím. Pokud se tak nestane do daného časového limitu, jsou spuštěny provozní brzdy. ETCS dále kontroluje maximální povolenou rychlost. Tento mód je možné používat v úrovních 1, 2 a 3. [6, 2.3.2.2]

3.2.1.3 Staff Responsible

V módu Staff Responsible strojvedoucí plně zodpovídá za to, že vlak bude dopraven do oblasti, která je vybavená ETCS. ETCS stále provádí nějaké kontroly, jako je třeba kontrola maximální rychlosti, případně maximální povolené ujeté vzdálenosti. Strojvedoucí ale může tyto hodnoty v módu SR libovolně měnit. Tento mód je možné používat v úrovních 1, 2 a 3. [6, 2.3.2.3]

3.2.1.4 Shunting

Do módu Shunting EVC přejde v případě, že obdrží příkaz od traťové části systému, nebo poté, co je tento mód zvolen strojvedoucím. V případě příkazu od traťové části je strojvedoucí povinen potvrdit přechod do módu SH. Pokud toto potvrzení není obdrženo do daného časového limitu, jsou spuštěny provozní brzdy. V módu SH je povoleno couvat. Mód Shunting zajišťuje, že se vlak nedostane za hranice oblasti, která je určená k posunu vlaků. Dále se v tomto módu kontroluje, aby vlak nepřesáhl maximální povolenou rychlost při posouvání. [6, 2.3.2.4] Tato rychlost je dána národní hodnotou. [7, 4.4.8.1.1] Tento mód je možné používat v úrovních 0, 1, 2 a 3. [6, 2.3.2.4]

3.2.1.5 Unfitted

Mód Unfitted se používá pro přejezd úseků, které nejsou vybaveny ETCS, případně úseků, kde je traťová část ETCS v procesu instalace. Dále se tento mód používá v oblastech s národním zabezpečovacím systémem, v případě, že vlak nedisponuje modulem STM. Tento mód je možné používat pouze v úrovni 0. [6, 2.3.2.5] [5]

3.2.1.6 Sleeping

Mód Sleeping se používá pro lokomotivy, které jsou ovládány vzdáleně. Tyto lokomotivy pouze dodávají tažnou sílu a jsou ovládány z vedoucí lokomotivy. V módu SL není strojvedoucí přítomen v kabině lokomotivy. Tento mód je možné používat v úrovních 0, 1, 2, 3 a STM. [6, 2.3.2.6]

3.2.1.7 Stand By

Systém se dostává do módu Stand By poté, co strojvedoucí spustí stanoviště strojvedoucího. Poté, co jsou přístroje vypnuty, se také používá mód SB. Tento mód je možné používat v úrovních 0, 1, 2, 3 a STM. [6, 2.3.2.7]

3.2.1.8 Trip

V definovaných případech, které jsou považována za kritická bezpečnostní rizika, přejde systém do módu Trip. Bezprostředně po přechodu do tohoto módu jsou spuštěny nouzové brzdy. ETCS monitoruje vlak až do úplného zastavení. Tento mód je možné používat v úrovních 1, 2, 3 a STM. [6, 2.3.2.8]

3.2.1.9 Post Trip

Do módu Post Trip se EVC dostává po potvrzení módu Trip strojvedoucím (*acknowledgement of Trip mode*). V tomto módu není možné jet dopředu a vzdálenost, kterou smí vlak ujet směrem dozadu, je omezena národní hodnotou. Tento mód je možné používat v úrovních 1, 2 a 3. [6, 2.3.2.9]

3.2.1.10 System Failure

Do módu System Failure se systém dostane v případě chyby spojené s bezpečností. Ihned jsou spuštěny nouzové brzdy. Tento mód je možné používat v úrovních 0, 1, 2, 3 a STM. [6, 2.3.2.10]

3.2.1.11 Isolation

V módu Isolation je vozidlová část ETCS odříznuta od zbytku systému, včetně například brzd. Nelze přijímat data od traťové části ani z vozidla. Tento mód je možné používat v úrovních 0, 1, 2, 3 a STM. [6, 2.3.2.11]

3.2.1.12 No Power

Mód No Power se používá v případě, že vozidlová část ETCS není napájena. V takovém případě jsou spuštěny nouzové brzdy. Tento mód je možné používat v úrovních 0, 1, 2, 3 a STM. [7, 4.4.4]

3.2.1.13 Non Leading

Mód Non Leading se používá pro lokomotivy, které pouze dodávají tažnou sílu a nejsou vedoucí lokomotivou, ale nejsou ovládané vzdáleně. V tomto módu je na rozdíl od módu SL přítomen strojvedoucí v kabině. Tento mód je možné používat v úrovních 0, 1, 2, 3 a STM. [6, 2.3.2.13]

3.2.1.14 STM European

V tomto módu se pomocí STM data z národního zabezpečovacího systému předávají EVC. Tento mód je možné používat v úrovni STM. [6, 2.3.2.14], [7, 4.4.16]

3.2.1.15 STM National

V módu STM National je využíváno traťových částí národního zabezpečovacího systému. Slouží v případech, kdy vozidlová část disponuje ETCS, ale traťová část disponuje pouze původním národním zabezpečovacím systémem. Tento mód je možné používat v úrovni STM. [6, 2.3.2.15]

3.2.1.16 Reversing

V módu Reversing je strojvedoucímu dovoleno změnit směr jízdy vlaku. Vzdálenost, kterou smí vlak couvat, je omezená, stejně jako rychlost couvání. Tyto limity dostává vlak od traťové části systému. [7, 4.4.18] V případě, že vlak překročí maximální povolenou vzdálenost, jsou spuštěny nouzové brzdy. Tento mód je možné používat v úrovních 1, 2 a 3. [6, 2.3.2.16]

3.2.2.1 Podmínky přechodu

Dále budou popsány podmínky, které se týkají módů, které jsou již implementovány nebo budou přidány v rámci této práce.

- **5**
Vlak stojí na místě, úroveň ETCS je 0 nebo 1 a strojvedoucí zvolí mód Shunting.
- **6**
Vlak stojí na místě, úroveň ETCS je 2 nebo 3 a byla přijata informace *Shunting granted by RBC* na základě požadavku zasláního strojvedoucím.
- **7**
Strojvedoucí potvrdí Trip, vlak stojí na místě a úroveň ETCS není 0 ani STM.
- **8**
Mód Staff Responsible je navržen strojvedoucímu a strojvedoucí zvolí SR. Navržení módu SR může nastat jen ve specifických situacích, konkrétně se jedná o proceduru *Start Of Mission* a proceduru *Train Trip*.
- **15**
Požadavek o potvrzení módu On Sight se zobrazí strojvedoucímu a strojvedoucí potvrdí mód OS. Požadavek o potvrzení módu OS se zobrazuje pouze ve specifických situacích, konkrétně se jedná o proceduru *On Sight* a proceduru *Start Of Mission*.
- **19**
Strojvedoucí zvolí *exit Shunting* a vlak stojí na místě.
- **28**
Stanoviště strojvedoucího je vypnuté.
- **29**
Vozidlová část ETCS není napájena.
- **31**
EVC má povolení k jízdě (Movement Authority), SSP (Static Speed Profile) i gradient a žádný specifický mód není vyžadován (Mode Profile) a úroveň ETCS je 2 nebo 3.
- **37**
Strojvedoucí zvolí *override* a vlak jede menší rychlostí, než je maximální rychlost pro spuštění funkce *override*.
- **40**
EVC používá *Mode Profile*, který definuje vstup do oblasti pro mód On-Sight a maximální bezpečný přední konec vlaku je uvnitř této oblasti.
- **49**
EVC přijme informaci „zastavit, pokud je mód SH“ (*stop if in shunting*) a *override* není aktivní.
- **50**
Požadavek o potvrzení módu Shunting se zobrazí strojvedoucímu a strojvedoucí potvrdí mód SH. Požadavek o potvrzení módu SH se zobrazuje pouze ve specifických situacích, konkrétně se jedná o proceduru *Entry In Shunting* a proceduru *Start Of Mission*.
- **51**
EVC používá *Mode Profile*, který definuje vstup do oblasti pro posun vlaků a maximální bezpečný přední konec vlaku je uvnitř této oblasti.

- **52**
Právě přejetá balíza není v seznamu očekávaných balíz pro mód SH a *override* není aktivní.
- **59**
Vlak stojí na místě a strojvedoucí potvrdil couvání (*reversing*). Požadavek o potvrzení módu RV se zobrazuje pouze ve specifických situacích, konkrétně se jedná o proceduru *Reversing*.
- **65**
Je přijat balízový telegram ve verzi, která není kompatibilní s jazykem ETCS a úroveň ETCS je 1, 2 nebo 3. [7, 4.6.3]

3.3 Procedurey

Procedurey popisují, jak má EVC reagovat na informace, které dostane z jiných komponent ETCS, případně na určité události. Procedurey se zaměřují na přechody módů a změny stavu.

Následující procedurey jsou ty, které se týkají módů RV, SH a PT. Pro implementaci těchto módů je tudíž nezbytné implementovat i tyto procedurey.

3.3.1 Procedura Shunting Initiated by Driver

Tato procedura popisuje způsob, jakým systém přejde do módu SH z módu FS, OS, SR, PT nebo UN v případě, že přechod do módu SH byl vyvolán strojvedoucím. [8, 5.6]

Při této proceduře mohou být splněny následující podmínky přechodu: 5, 6.

3.3.2 Procedura End of Mission

Ukončení mise znamená, že traťová část přestane dohlížet nad pohyby vlaku. Tato procedura je iniciována vozidlovou částí systému přechodem do určitých módů. Procedura *End of Mission* definuje způsob přechodu do módu:

- SB z módů FS, OS, UN, NL, SR nebo RV
- SL z módu SB (pokud už procedura proběhla při přechodu do módu SB, tak se znovu neprovádí)
- SH z módů FS, OS, SR, PT nebo UN

Podle toho, do jakého módu systém přešel, se musí vymazat některá data. Při přechodu do módu:

- SB se vymaže *Movement Authority*
- SL se vymaže *Movement Authority*
- SH se vymažou *Movement Authority* a *Train Data* [8, 5.5]

Při této proceduře mohou být splněny následující podmínky přechodu: 5, 6.

3.3.3 Procedura Entry in Shunting

Tato procedura popisuje vjezd vlaku do oblastí určené pro posun vlaků (*shunting area*). Rozkaz k přechodu do módu SH dostane vlaková část systému pomocí paketu 80: Mode profile. Tento přechod musí být potvrzen strojvedoucím. Ve chvíli, kdy systém přejde do módu SH se vykoná procedura *End of Mission*. [8, 5.7]

Při této proceduře mohou být splněny následující podmínky přechodu: 5, 6, 50, 51.

3.3.4 Procedura Train Trip

Procedura *Train Trip* popisuje situaci, která nastane potom, co systém přešel do módu TR. Po úplném zastavení vlaku může systém přejít hned do několika módů (PT, UN, SE, SN) v závislosti na úrovni ETCS. Dále se v této proceduře řeší přechody z módu PT. [8, 5.11]

Při této proceduře mohou být splněny následující podmínky přechodu: 5, 6, 7, 8, 15, 31.

3.3.5 Procedura On-Sight

ETCS musí přejít do módu OS předtím, než dorazí na začátek oblasti určené pro OS, nebo nejpozději v moment, kdy vlak dorazí na začátek této oblasti. Dále je také v jistých situacích nutné, aby strojvedoucí přechod do módu OS potvrdil. Tento přechod popisuje procedura *On-Sight*. [8, 5.9]

Při této proceduře mohou být splněny následující podmínky přechodu: 15, 40.

3.3.6 Procedura Start of Mission

Procedura *Start of Mission* popisuje zahájení mise, ke kterému může dojít:

- po probuzení vlaku
- po ukončení posunu
- po ukončení mise
- poté, co se řízená lokomotiva stane vedoucí lokomotivou

Ve všech těchto situacích se ETCS nachází v módu SB, ale stav systému se může lišit tím, jaká data jsou uložena ve vlakové části. [8, 5.4]

Při této proceduře mohou být splněny následující podmínky přechodu: 5, 6, 8, 15 a 50.

3.3.7 Procedura Reversing

Během této procedury vlak může přejít do módu RV, který dovoluje vlaku zpětný pohyb. Používá se v případě, kdy hrozí nebezpečí a je potřeba zacouvat zpět na bezpečné místo. Oblasti, kde je povolené couvání, jsou EVC oznámeny pomocí paketu 138: Reversing area information. Dále se pomocí paketu 139: Reversing supervision information posílají informace o maximální možné rychlosti a vzdálenosti, kterou je možné jet v těchto oblastech. [8, 5.13]

Při této proceduře mohou být splněny následující podmínky přechodu: 59.

3.4 Dohled nad pohyby vlaku

Zabezpečovací systém dohlíží na to, aby se vlak nepohyboval směrem, který je v dané situaci nežádoucí. Tyto funkcionality systému se týkají i nově přidávaných módů. U jednotlivých funkcionalit je vždy vyjmenované, ve kterých módech se používají.

3.4.1 Roll Away Protection

Tato funkcionality ETCS zabraňuje vlaku v pohybu směrem, který neodpovídá současné pozici směrového ovladače v aktivním stanovišti. Pokud je ovladač v neutrální pozici, RAP zabraňuje jak pohybu vpřed, tak couvání. V případě, že systém zaznamená pohyb vlaku o více než povolenou vzdálenost, která je dána národní hodnotou, spustí se brzdy. Zároveň se strojvedoucím zobrazí upozornění s důvodem použití brzd. Brzdy jsou spuštěny až do úplného zastavení vlaku a následného potvrzení strojvedoucím. Po ukončení brždění se současná pozice použije jako výchozí bod pro RAP. [9, 3.14.2, 3.14.1]

Roll Away Protection se používá v následujících módech: SH, FS, SR, OS, UN, PT, SE a RV. [7, 4.5.2]

3.4.2 Reverse Movement Protection

RMP zabraňuje vlaku v pohybu směrem opačným k povolenému směru. Povolený směr je dán současným platným MA, pokud je k dispozici. Pokud systém zaznamená pohyb v opačném směru, který překročil maximální povolenou vzdálenost (daná národní hodnotou), spustí se brzdy. Systém informuje strojvedoucího o důvodu spuštění brzd. Brzdí se až do úplného zastavení vlaku a následného potvrzení strojvedoucím. Po ukončení brždění se současná pozice použije jako výchozí bod pro RMP. [9, 3.14.3]

Reverse Movement Protection se používá v následujících módech: FS, SR, OS, PT, SE a RV. [7, 4.5.2]

3.4.3 Standstill supervision

Standstill supervision se používá pouze v módu SB. [7, 4.5.2] Systém zabraňuje jakémukoliv pohybu vlaku. Při ujetí vzdálenosti, která překročí maximální povolenou (dána národní hodnotou), se spustí brzdy. Systém informuje strojvedoucího o důvodu spuštění brzd. Brzdí se až do úplného zastavení vlaku a následného potvrzení strojvedoucím. [9, 3.14.4]

3.5 Současný stav projektu trenážéru

Projekt ETCS trenážéru, který je společným dílem Fakulty informačních technologií ČVUT v Praze a Fakulty dopravní ČVUT v Praze, běží již několik semestrů. V současné době na něm pracuje přibližně 30 studentů. Někteří v rámci předmětů *Softwarový týmový projekt 1* a *Softwarový týmový projekt 2*, jiní v rámci bakalářských prací. Trenážér se skládá z více podprojektů (EVC, RBC, DMI, lektorské PC, vizualizace). Studenti pracující na projektu jsou rozděleni do týmů, kterým jsou přiřazeny jednotlivé komponenty. Projekt EVC, jehož součástí budou výsledky této práce, je tedy pouze částí rozsáhlejšího celku.

3.5.1 Stav EVC

Projekt EVC zaznamenal během posledního běhu předmětu *Softwarový týmový projekt 2* velké změny. Došlo k celkovému přepisu modulu EVC a přechodu na novou architekturu. Tyto změny měly za cíl zlepšit přehlednost kódu a zároveň zajistit jednodušší rozšiřitelnost. Rozsah testování této nové verze ale nebyl příliš značný. To by mohlo v průběhu implementace způsobit problémy, je totiž možné, že bude potřeba opravit chyby, které při přepisu vznikly a nebyly objeveny.

V současné verzi projektu se již používají následující módy:

- Full Supervision
- Stand By
- No Power
- Trip
- On Sight
- Staff Responsible

Je implementována funkcionality v módech a některé přechody mezi módy. Ne všechny možné přechody jsou implementovány, například veškeré přechody týkající se procedury Override nejsou implementované, protože procedura Override není implementována.

Přidání nových módů (RV, SH, PT) znamená přidání funkcionality v těchto módech a přechody mezi nimi, ale i přechody mezi novými módy a módy, které se již používají.

3.5.1.1 Architektura

V projektu se využívá dvou konceptů, které je pro pochopení fungování komponenty EVC potřeba blíže popsat.

3.5.1.1.1 Service Service, nebo také služba, je třída, která se zabývá jednou funkcionalitou. Jedná se například o posílání zpráv, přijímání zpráv, zpracování dat, nebo ukládání dat. Veškeré tyto třídy dědí od rozhraní `IService`. Ostatní třídy pomocí rozhraní mohou tyto služby využívat.

3.5.1.1.2 Message Handler Message Handler je třída, která zpracovává jednu z příchozích zpráv. Každá z tříd, které dědí od třídy `MessageHandler`, má na starosti zpracování jedné konkrétní zprávy. Vždy když EVC obdrží zprávu od některé z dalších komponent ETCS, zavolá se metoda `HandleMessageBody` příslušného handleru. V této metodě se pro správné zpracování zprávy využívá metod jednotlivých služeb.

3.6 Funkční a nefunkční požadavky

3.6.1 Funkční požadavky

- **F1 Kontrola maximální rychlosti v módu RV**
EVC bude kontrolovat maximální rychlost v módu RV tak, aby nepřekročila hodnotu, která byla poslána z traťové části systému (RBC).
- **F2 Kontrola maximální ujeté vzdálenosti v módu RV**
EVC bude kontrolovat maximální vzdálenost ujetou v módu RV tak, aby nepřekročila hodnotu, která byla poslána z traťové části systému (RBC).
- **F3 Kontrola maximální rychlosti v módu SH**
EVC bude kontrolovat maximální rychlost v módu SH tak, aby nepřekročila hodnotu, která je určena národními hodnotami.
- **F4 Roll Away Protection**
EVC bude kontrolovat pohyby vlaku v módech SH, FS, SR, OS, UN, PT, SE a RV. Vlak bude reagovat na pohyby v nežádoucím směru tak, jak je definováno specifikací.
- **F5 Reverse Movement Protection**
EVC bude kontrolovat zpětný pohyb vlaku v módech FS, SR, OS, PT, SE a RV. Vlak bude reagovat na zpětný pohyb tak, jak je definováno specifikací.
- **F6 Standstill supervision**
EVC bude bránit pohybu vlaku v módu SB. Vlak bude reagovat na pohyby vlaku tak, jak je definováno specifikací.
- **F7 Přejechy z módu Reversing**
Z módu Reversing bude EVC schopné přecházet do módu SB za podmínky 28 a do módu NP za podmínky 29.
- **F8 Přejechy z módu Shunting**
Z módu Shunting bude EVC schopné přecházet do módu SB za podmínek 19 a 28, do módu TR za podmínek 49, 52 a 65 a do módu NP za podmínky 29.
- **F9 Přejechy z módu Post Trip**
Z módu Post Trip bude EVC schopné přecházet do módu SH za podmínek 5, 6 a 50, do módu FS za podmínky 31, do módu SR za podmínky 8, do módu OS za podmínky 15 a do módu NP za podmínky 29.
- **F10 Přejechy do módu Reversing**
EVC bude schopné přecházet do módu Reversing z módu FS za podmínky 59 a z módu OS za podmínky 59.
- **F11 Přejechy do módu Shunting**
EVC bude schopné přecházet do módu Shunting z módu SB za podmínek 5, 6 a 50, z módu FS za podmínek 5, 6, 50 a 51, z módu SR za podmínek 5, 6 a 51 a z módu OS za podmínek 5, 6, 50 a 51.
- **F12 Přejechy do módu Post Trip**
EVC bude schopné přecházet do módu Post Trip z módu TR za podmínky 7.

3.6.2 Nefunkční požadavky

- **N1 Programovací jazyk**
Pro implementaci bude použit programovací jazyk C++.
- **N2 Verze subsetu**
Pro implementaci bude použita specifikace ze subsetu verze 2.3.0.



Kapitola 4

Návrh

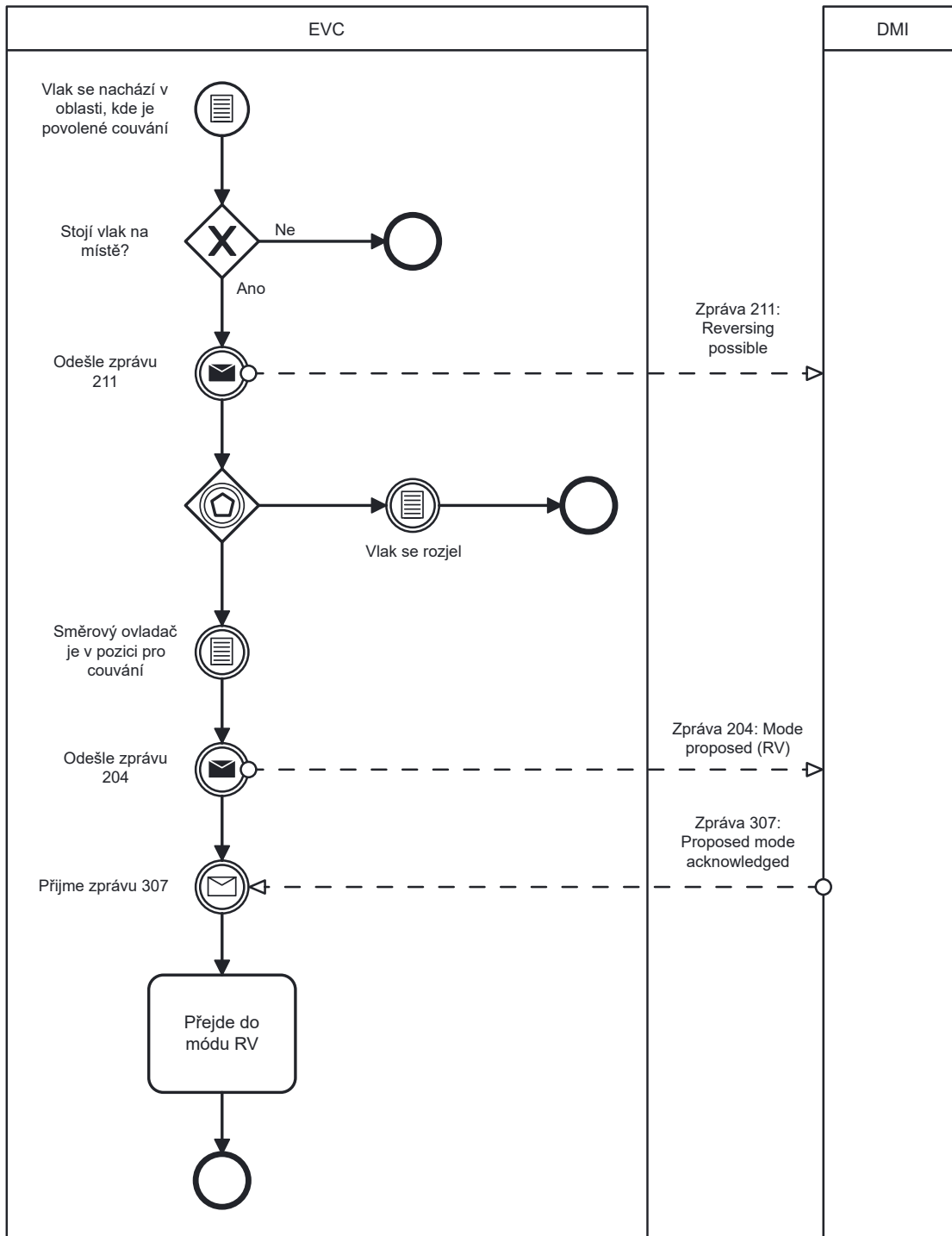
Tato kapitola se věnuje návrhu procedur, které se týkají módu RV, SH a PT. Dále pak obsahuje popis struktury zpráv a paketů, které se během těchto procedur posílají, popisy jednotlivých proměnných, které zprávy obsahují a situace, kdy se zprávy posílají nebo přijímají. Další sekce obsahuje návrh implementace přechodů, které nejsou pokryté procedurami. V závěru kapitoly jsou popsány služby, které vzniknou a budou zpracovávat data z přijatých zpráv a správně nastavovat vnitřní stav systému.

Návrh komunikace mezi EVC a DMI byl konzultován s týmem, který v současné době pracuje na komponentě DMI. Na základě zpětné vazby byly některé zprávy poupraveny, případně přidány.

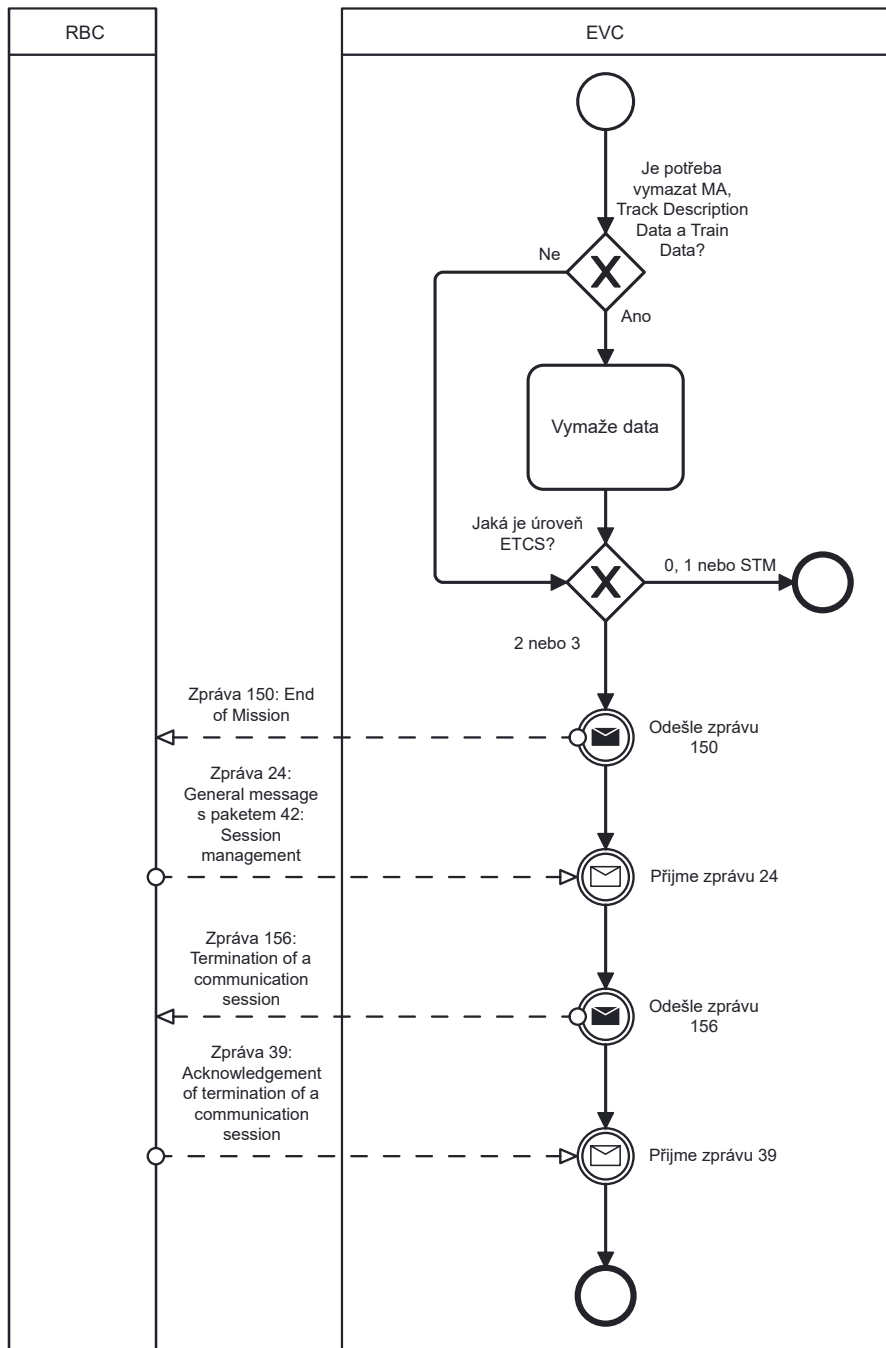
4.1 Procedury

V této sekci budou pomocí modelů blíže popsány jednotlivé procedury a definovány situace, ve kterých se posílají zprávy mezi jednotlivými komponentami. Procedury pokrývají většinu podmínek přechodu, které byly popsány ve funkčních požadavcích.

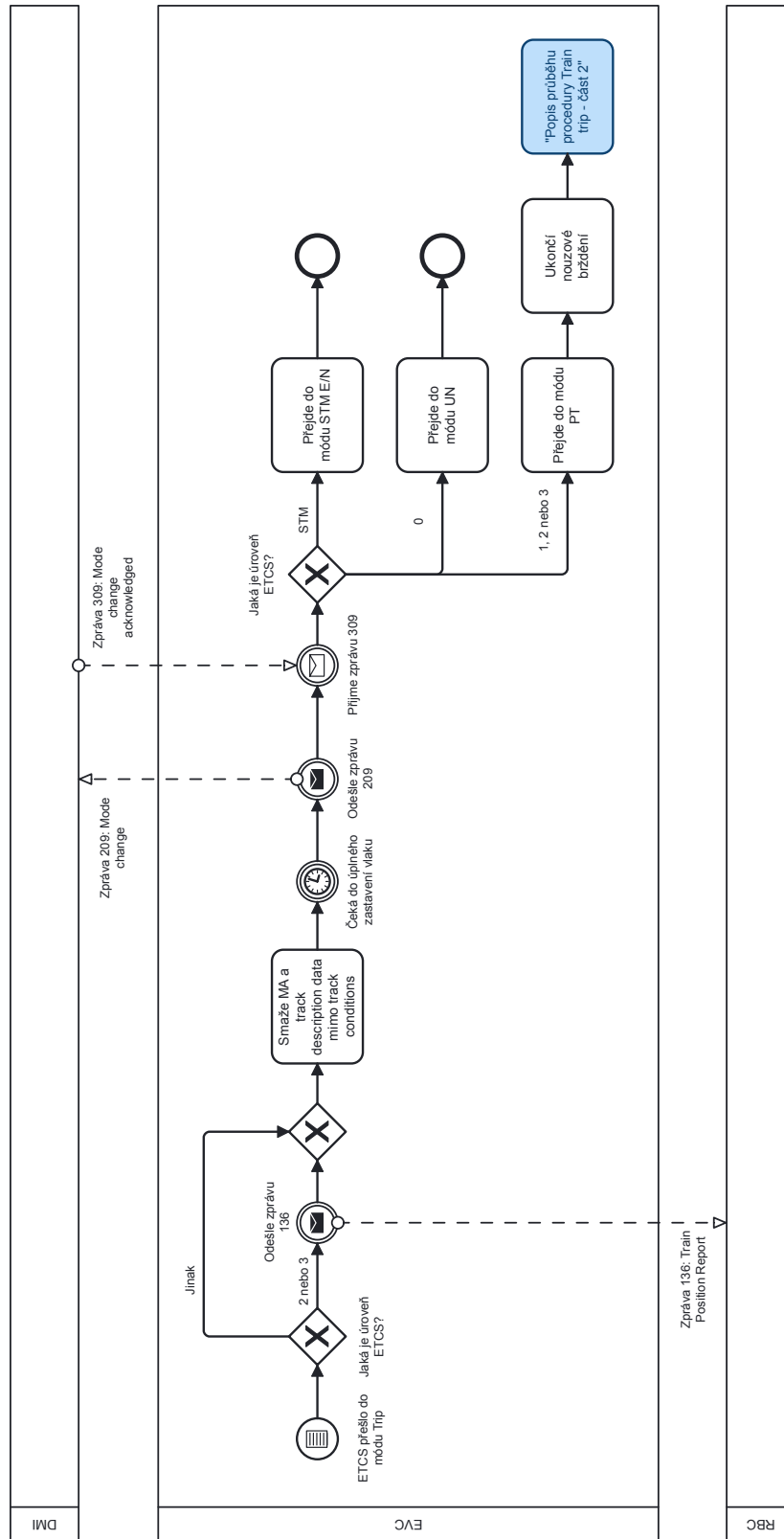
Pro znázornění průběhu procedur byla vybrána BPMN notace a to hlavně kvůli tomu, že poskytuje možnost znázornění aktérů v procesu a zpráv, které se mezi aktéry posílají. Je tedy jednoduše možné zachytit zprávy, které si mezi sebou jednotlivé komponenty vyměňují. Význam symbolů, které jsou použité v modelech, je možné dohledat v příloze A.



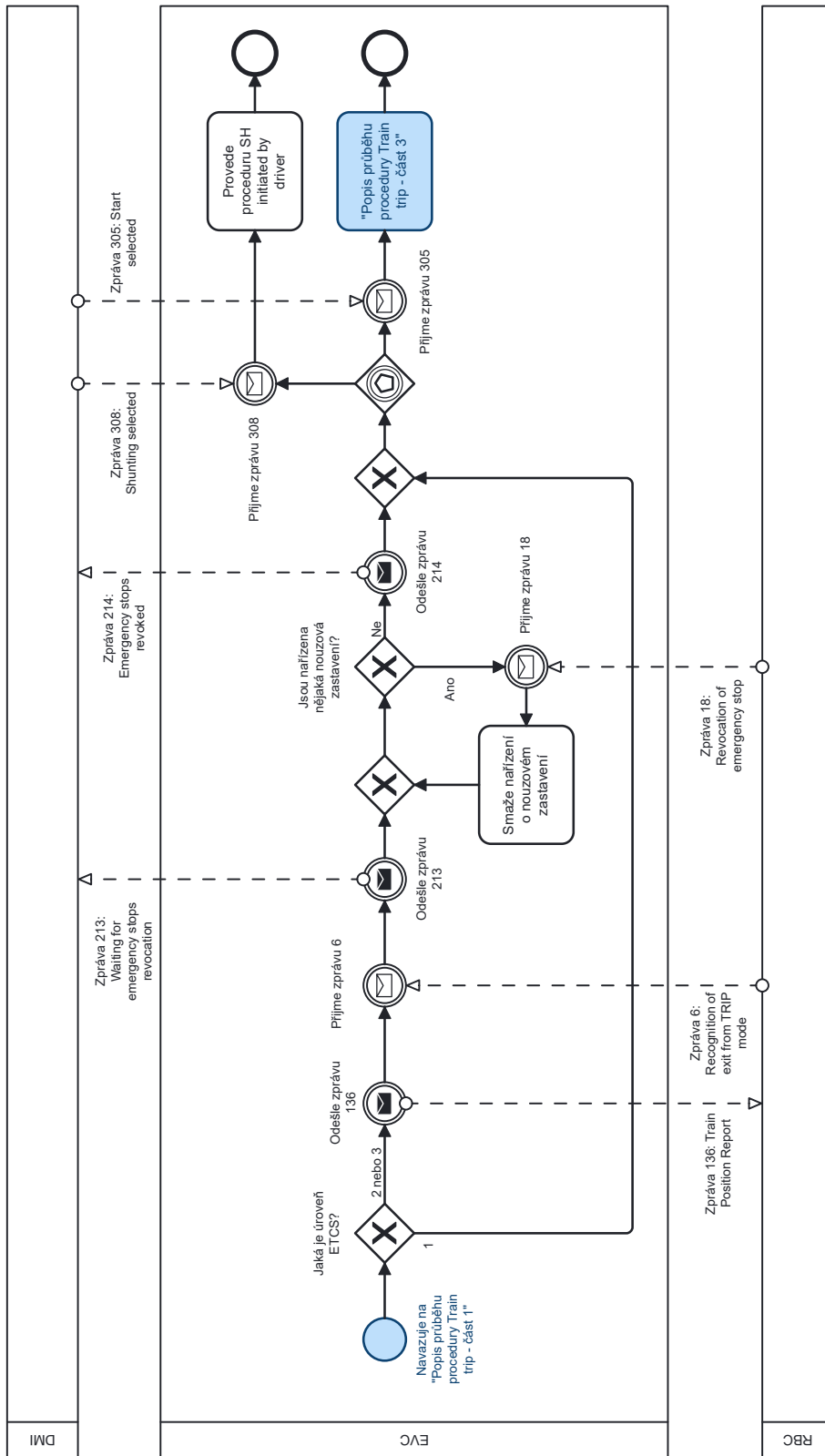
■ **Obrázek 4.2** Popis průběhu procedury Reversing



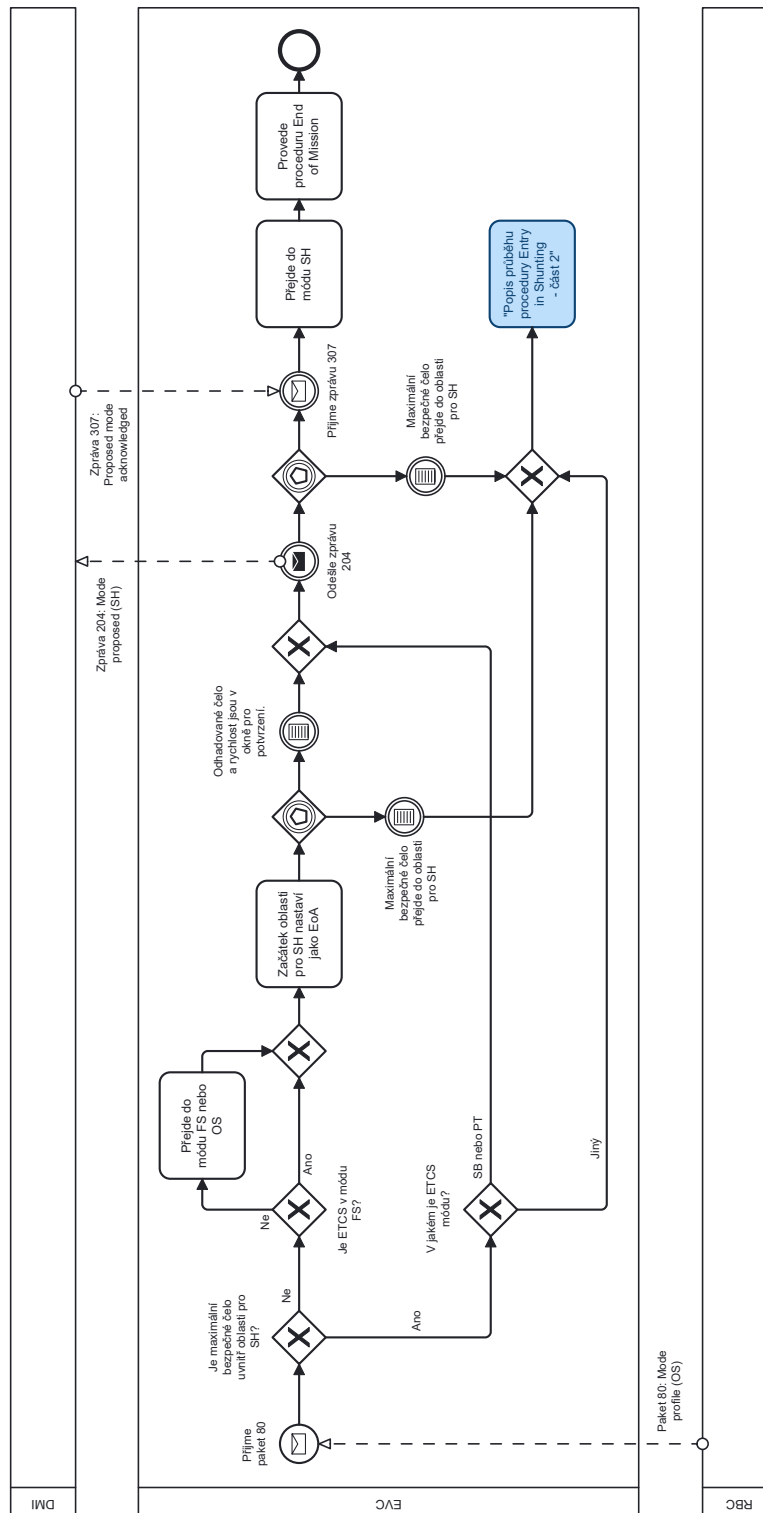
■ Obrázek 4.4 Popis průběhu procedury End of Mission



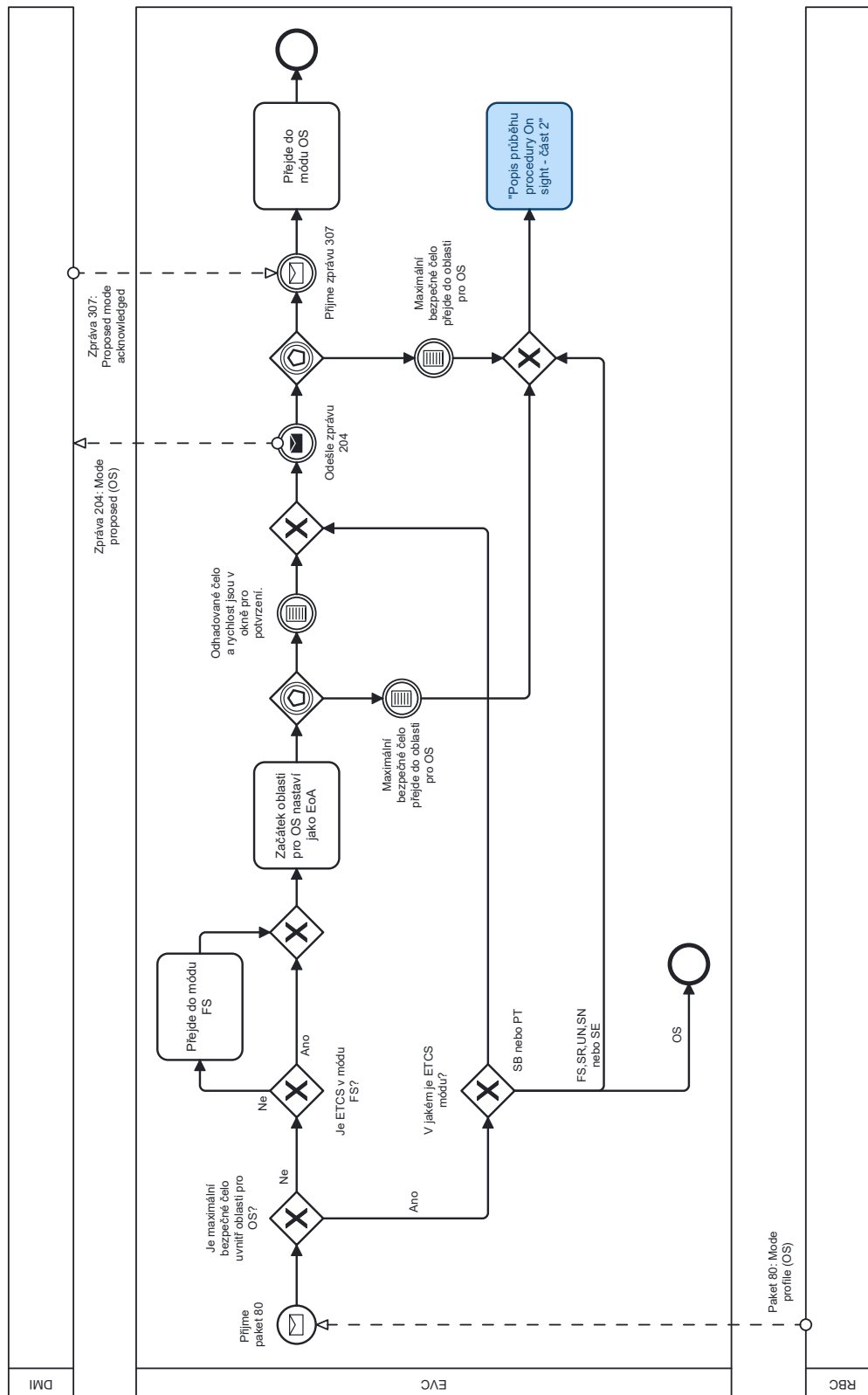
■ Obrázek 4.5 Popis průběhu procedury Train trip – část 1



■ Obrázek 4.6 Popis průběhu procedury Train trip – část 2



■ Obrázek 4.8 Popis průběhu procedury Entry in Shunting – část 1



■ Obrázek 4.10 Popis průběhu procedury On-Sight – část 1

4.2 Zprávy

V rámci procedur, které budou implementovány, EVC komunikuje s ostatními komponentami systému (především s DMI a RBC). Tato komunikace probíhá prostřednictvím zpráv. Komunikace s RBC, struktura a obsah zpráv, které se posílají mezi EVC a RBC, je definovaná specifikací (System Requirements Specification Chapter 8 Messages). Komunikace mezi EVC a DMI není definovaná, tudíž je nutné vytvořit některé nové zprávy.

4.2.1 Identifikátory zpráv

Podle id zprávy je možné rozlišit, od které komponenty pochází. Případně, pro kterou komponentu je určená.

- 1 až 199 jsou rezervované pro zprávy od RBC pro EVC
- 100 až 199 jsou rezervované pro zprávy od EVC pro RBC
- 200 až 299 jsou rezervované pro zprávy od EVC pro DMI
- 300 až 399 jsou rezervované pro zprávy od DMI pro EVC
- 400 až 499 jsou rezervované pro zprávy od ODO
- 500 až 599 jsou rezervované pro zprávy od BTM
- 601 je zpráva od EVC pro TIU
- 601 je zprávy od TIU pro EVC

4.2.2 Zprávy posílané DMI

4.2.2.1 Zpráva 209: Mode change

Zpráva se posílá v procedurách *Entry In Shunting* a *On-Sight*. Značí, že nastala změna módu. Strojvedoucí tuto změnu módu musí potvrdit a DMI pošle zprávu 309: Mode change acknowledged.

■ **Tabulka 4.1** Struktura zprávy 209: Mode change

Proměnná	Datový typ	Poznámka
NID_MESSAGE	MessageID (int)	Pevná hodnota 209
mode	EVCMode	Mód, do kterého ETCS přešlo

4.2.2.2 Zpráva 210: Waiting for SH request response

Zpráva se posílá v proceduře *SH Initiated by Driver*. Oznamuje strojvedoucímu, že EVC zaslalo RBC požadavek o přechod do módu SH a čeká na odpověď.

■ **Tabulka 4.2** Struktura zprávy 210: Waiting for SH request response

Proměnná	Datový typ	Poznámka
NID_MESSAGE	MessageID (int)	Pevná hodnota 210

4.2.2.3 Zpráva 211: Reversing possible

Zpráva se posílá v proceduře *Reversing* v případě, že vlak stojí na místě v oblasti, ve které je povoleno couvat. Tato zpráva oznamuje strojvedoucímu, že je možné přejít do módu RV.

■ **Tabulka 4.3** Struktura zprávy 211: Reversing possible

Proměnná	Datový typ	Poznámka
NID_MESSAGE	MessageID (int)	Pevná hodnota 211

4.2.2.4 Zpráva 212: Revoke brake command

Zpráva se posílá v případě, že vlak zabrzdil až do úplného zastavení a k uvolnění brzd je potřeba potvrzení od strojvedoucího. Zpráva je součástí funkcionalit RAP, RMP a Standstill supervision.

■ **Tabulka 4.4** Struktura zprávy 212: Revoke brake command

Proměnná	Datový typ	Poznámka
NID_MESSAGE	MessageID (int)	Pevná hodnota 212
reason	BrakeCommand (int)	Důvod, proč byli brzdy spuštěny

Důvod pro spuštění servisních brzd je reprezentován pomocí výčtové třídy BrakeCommand, která je popsána tabulkou 4.5.

■ **Tabulka 4.5** BrakeCommand enum

Hodnota	Název	Význam
0	NONE	Žádný příkaz k brždění nebyl vydán
1	RAP	Roll away protection
2	RMP	Reverse movement protection
3	STANDSTILL_SUPERVISION	Standstill supervision
4	RAP_RMP	Brzdy spuštěny z důvodu RAP i RMP
5	SPEED_EXCEEDED	Brzdy spuštěny z důvodu překročení maximální povolené rychlosti pro daný mód

4.2.2.5 Zpráva 213: Waiting for emergency stops revocation

Zpráva se posílá v proceduře *Train trip*. Dokud RBC neodvolá veškeré zprávy o nouzovém zastavení, tak není možné pokračovat v proceduře.

■ **Tabulka 4.6** Struktura zprávy 213: Waiting for emergency stop revocation

Proměnná	Datový typ	Poznámka
NID_MESSAGE	MessageID (int)	Pevná hodnota 213

4.2.2.6 Zpráva 214: Emergency stops revoked

Zpráva se posílá v proceduře *Train trip*. EVC již nemá žádné zprávy o nouzovém zastavení, strojvedoucí tedy může zvolit *start* nebo *shunting*.

■ **Tabulka 4.7** Struktura zprávy 214: Emergency stops revoked

Proměnná	Datový typ	Poznámka
NID_MESSAGE	MessageID (int)	Pevná hodnota 214

4.2.3 Zprávy přijímané od DMI

4.2.3.1 Zpráva 308: Shunting selected

Zpráva se posílá vždy, když strojvedoucí na displeji zmáčkne tlačítko *Shunting*. Tato zpráva zahajuje proceduru *Shunting Initiated by Driver* a je součástí procedury *Start of Mission*.

Zpráva se dále posílá, když strojvedoucí na displeji zmáčkne tlačítko *Exit Shunting*, kterým si vyžádá opuštění módu SH. EVC na tuto zprávu reaguje přechodem do módu SB za podmínky, že vlak stojí na místě.

■ **Tabulka 4.8** Struktura zprávy 308: Shunting selected

Proměnná	Datový typ	Poznámka
NID_MESSAGE	MessageID (int)	Pevná hodnota 308
shunting	bool	True - strojvedoucí zvolil Shunting, False - strojvedoucí zvolil Exit shunting

4.2.3.2 Zpráva 309: Mode change acknowledged

Zpráva, která se posílá jako odpověď na zprávu 209: Mode change.

■ **Tabulka 4.9** Struktura zprávy 309: Mode change acknowledged

Proměnná	Datový typ	Poznámka
NID_MESSAGE	MessageID (int)	Pevná hodnota 309
mode	EVCMode	Mód, který strojvedoucí potvrdil

4.2.3.3 Zpráva 310: Revoke brake command acknowledgement

Zpráva se posílá jako odpověď na zprávu 212: Revoke brake command po potvrzení strojvedoucím. Zpráva je součástí funkcionalit RAP, RMP a Standstill supervision.

■ **Tabulka 4.10** Struktura zprávy 310: Revoke brake command acknowledgement

Proměnná	Datový typ	Poznámka
NID_MESSAGE	MessageID (int)	Pevná hodnota 310

4.2.4 Zprávy odesílané RBC

4.2.4.1 Zpráva 130: Request for Shunting

Zpráva se odesílá během procedury *Shunting Initiated by Driver*. Strojvedoucí pomocí této zprávy žádá o přechod do módu SH. Vždy obsahuje paket 0: Position Report, nebo paket 1: Position Report based on two balise groups.

■ **Tabulka 4.11** Struktura zprávy 130: Request for Shunting

Proměnná	Datový typ	Poznámka
NID_MESSAGE	MessageID (int)	Pevná hodnota 130
L_MESSAGE	unsigned int	Délka zprávy v bytech
T_TRAIN	unsigned int	Časová značka odeslání zprávy
NID_ENGINE	unsigned int	Identifikátor vlaku
m_Packets	vector<Packet>	Paket 0 nebo 1

4.2.4.2 Zpráva 147: Acknowledgement of Emergency Stop

Zpráva se odesílá jako odpověď na zprávu 15: Conditional Emergency Stop a zprávu 16: Unconditional Emergency Stop.

■ **Tabulka 4.12** Struktura zprávy 147: Acknowledgement of Emergency Stop

Proměnná	Datový typ	Poznámka
NID_MESSAGE	MessageID (int)	Pevná hodnota 147
L_MESSAGE	unsigned int	Délka zprávy v bytech
T_TRAIN	unsigned int	Časová značka odeslání zprávy
NID_ENGINE	unsigned int	Identifikátor vlaku
NID_EM	unsigned short	Identifikátor nouzové zprávy, kterou traťová část potvrzuje
Q_EMERGENCYSTOP	unsigned short	Značí, jestli vlak již přešel místo pro nouzové zastavení ve chvíli, kdy přijal zprávu o nouzovém zastavení
m_Packets	vector<Packet>	Zpráva obsahuje paket 0 nebo 1

4.2.5 Zprávy přijímané od RBC

4.2.5.1 Zpráva 6: Recognition of exit from TRIP mode

Zpráva se přijímá během procedury *Train Trip*. RBC touto zprávou potvrzuje přechod z módu TR do módu PT.

■ **Tabulka 4.13** Struktura zprávy 6: Recognition of exit from TRIP mode

Proměnná	Datový typ	Poznámka
NID_MESSAGE	MessageID (int)	Pevná hodnota 6
L_MESSAGE	unsigned int	Délka zprávy v bytech
T_TRAIN	unsigned int	Časová značka odeslání zprávy
M_ACK	unsigned short	Značí, jestli zpráva musí být potvrzena vozidlovou částí
NID_LRBG	unsigned int	Identifikátor LRBG

4.2.5.2 Zpráva 15: Conditional Emergency Stop

Tato zpráva obsahuje informace o tom, kde se nachází místo nouzového zastavení (vzdálenost místa od LRBG). Vozidlová část odpoví zprávou 147: Acknowledgement of Emergency Stop. Pokud vlak již přešel místo nouzového zastavení, tak se zpráva ignoruje, ale EVC informuje strojvedoucího. V opačném případě se místo zastavení použije pro definování EoA a SvL (pokud se nenachází až za současným EoA). [9, 3.10.2]

■ **Tabulka 4.14** Struktura zprávy 15: Conditional Emergency Stop

Proměnná	Datový typ	Poznámka
NID_MESSAGE	MessageID (int)	Pevná hodnota 15
L_MESSAGE	unsigned int	Délka zprávy v bytech
T_TRAIN	unsigned int	Časová značka odeslání zprávy
M_ACK	unsigned short	Značí, jestli zpráva musí být potvrzena vozidlovou částí
NID_LRBG	unsigned int	Identifikátor LRBG
NID_EM	unsigned short	Identifikátor zprávy nouzového zastavení
Q_SCALE	Q_Scale (int)	Určuje měřítko vzdálenosti (10 cm, 1 m, 10 m)
Q_DIR	unsigned short	Specifikuje, pro který směr jízdy je informace platná
D_EMERGENCYSTOP	unsigned short	Vzdálenost mezi LRBG a místem pro nouzové zastavení

4.2.5.3 Zpráva 16: Unconditional Emergency Stop

V případě, že traťová část systému přijme tuto zprávu, tak ihned přejde do módu TR. [9, 3.10.2]

■ **Tabulka 4.15** Struktura zprávy 16: Unconditional Emergency Stop

Proměnná	Datový typ	Poznámka
NID_MESSAGE	MessageID (int)	Pevná hodnota 16
L_MESSAGE	unsigned int	Délka zprávy v bytech
T_TRAIN	unsigned int	Časová značka odeslání zprávy
M_ACK	unsigned short	Značí, jestli zpráva musí být potvrzena vozidlovou částí
NID_LRBG	unsigned int	Identifikátor LRBG
NID_EM	unsigned short	Identifikátor zprávy nouzového zastavení

4.2.5.4 Zpráva 18: Revocation of Emergency Stop

Zpráva, která ruší nařízení o nouzovém zastavení. Pro přechod z módu PT je nutné, aby RBC zrušilo veškerá nařízení o nouzových zastaveních. EVC přijímá tuto zprávu v průběhu procedury *Train Trip*.

■ **Tabulka 4.16** Struktura zprávy 18: Revocation of Emergency Stop

Proměnná	Datový typ	Poznámka
NID_MESSAGE	MessageID (int)	Pevná hodnota 18
L_MESSAGE	unsigned int	Délka zprávy v bytech
T_TRAIN	unsigned int	Časová značka odeslání zprávy
M_ACK	unsigned short	Značí, jestli zpráva musí být potvrzena vozidlovou částí
NID_LRBG	unsigned int	Identifikátor LRBG
NID_EM	unsigned short	Identifikátor zprávy nouzového zastavení

4.2.5.5 Zpráva 27: SH Refused

Zpráva se přijímá během procedury *Shunting Initiated by Driver* jako jedna z možných odpovědí na zprávu 130: Request for Shunting.

■ **Tabulka 4.17** Struktura zprávy 27: SH Refused

Proměnná	Datový typ	Poznámka
NID_MESSAGE	MessageID (int)	Pevná hodnota 27
L_MESSAGE	unsigned int	Délka zprávy v bytech
T_TRAIN	unsigned int	Časová značka odeslání zprávy
M_ACK	unsigned short	Značí, jestli zpráva musí být potvrzena vozidlovou částí
NID_LRBG	unsigned int	Identifikátor LRBG
T_TRAIN_RQST	unsigned int	Časová značka požadavku pro přechod do SH

4.2.5.6 Zpráva 28: SH Authorised

Zpráva se přijímá během procedury *Shunting Initiated by Driver* jako jedna z možných odpovědí na zprávu 130: Request for Shunting.

■ **Tabulka 4.18** Struktura zprávy 28: SH Authorised

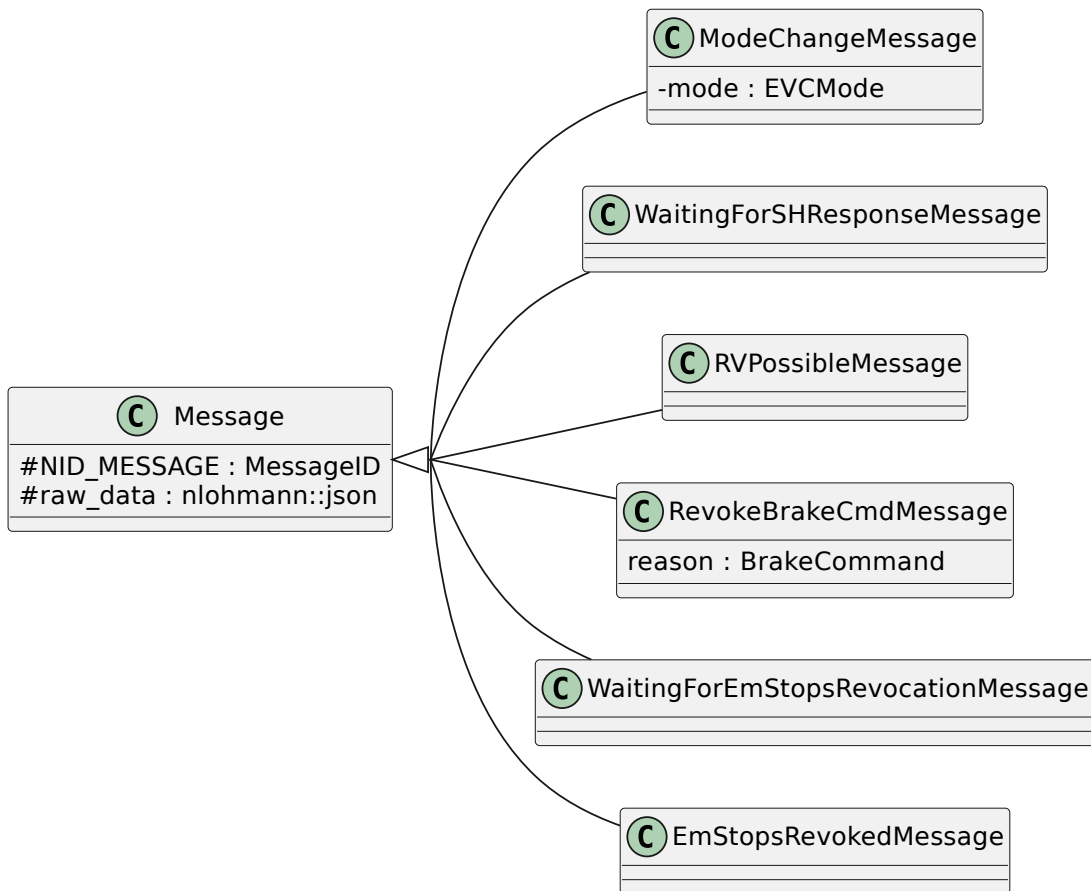
Proměnná	Datový typ	Poznámka
NID_MESSAGE	MessageID (int)	Pevná hodnota 28
L_MESSAGE	unsigned int	Délka zprávy v bytech
T_TRAIN	unsigned int	Časová značka odeslání zprávy
M_ACK	unsigned short	Značí, jestli zpráva musí být potvrzena vozidlovou částí
NID_LRBG	unsigned int	Identifikátor LRBG
T_TRAIN_RQST	unsigned int	Časová značka požadavku pro přechod do SH
m_Packets	vector<Packet>	Volitelné pakety

4.2.6 Reprezentace zpráv v EVC

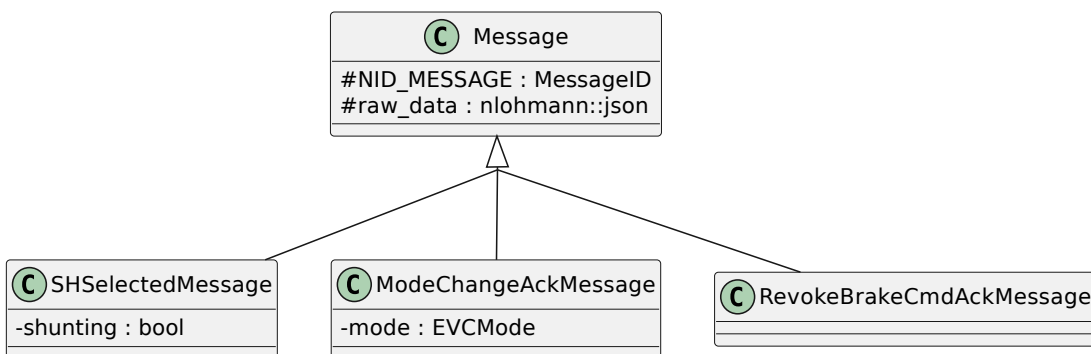
V projektu je již implementována řada tříd, které reprezentují zprávy. Základním stavebním prvkem hierarchie těchto tříd je třída `Message`, která obsahuje proměnnou `NID_MESSAGE`.

Zprávy, které EVC přijímá od DMI, dědí přímo od třídy `Message`.

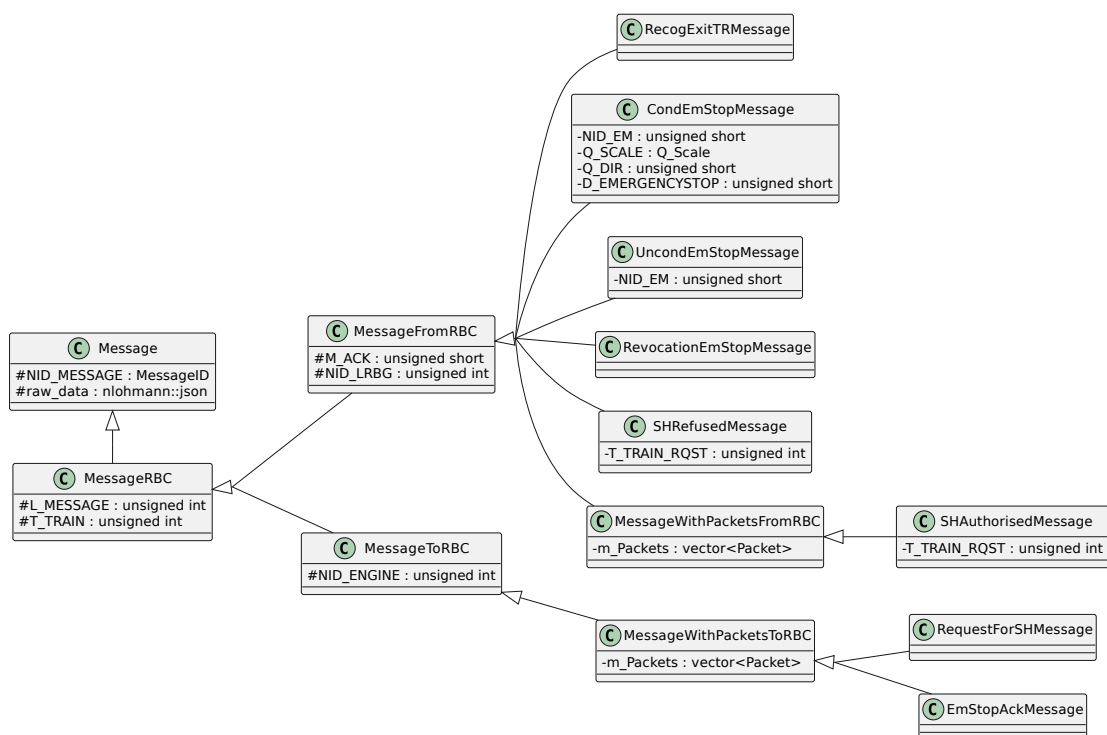
Zprávy, které EVC přijímá od RBC, jsou potomky třídy `MessageFromRBC`, případně třídy `MessageWithPacketsFromRBC`, pokud součástí zprávy mohou být pakety. Zprávy posílané z EVC do RBC dědí od třídy `MessageToRBC`, případně `MessageWithPacketsToRBC`. Zprávy, které budou nově přidány, budou vždy dědit od některé již existující třídy.



■ **Obrázek 4.12** Class diagram tříd reprezentujících zprávy pro komunikaci s DMI - zprávy posílané DMI



■ **Obrázek 4.13** Class diagram tříd reprezentujících zprávy pro komunikaci s DMI - zprávy přijímané od DMI



■ **Obrázek 4.14** Class diagram tříd reprezentujících zprávy pro komunikaci s RBC

4.3 Pakety

Některé zprávy mohou obsahovat pakety. Ty obsahují dodatečné informace, případně mění způsob, jakým se na zprávu reaguje. V této sekci je popsána struktura několika paketů, které se týkají nově přidávaných módů, nebo přechodů mezi těmito módy.

4.3.1 Paket 42: Session Management

Paket se posílá ve zprávě 24: General Message v případě, že traťová část systému si chce vyžádat ukončení mise. K tomu dochází v rámci procedury *End of Mission*.

■ **Tabulka 4.19** Struktura paketu 42: Session Management

Proměnná	Datový typ	Poznámka
NID_PACKET	PacketID (int)	Pevná hodnota 42
Q_DIR	unsigned short	Specifikuje, pro který směr jízdy je informace platná
L_PACKET	unsigned short	Délka paketu v bytech
Q_RBC	unsigned short	Tato proměnná určuje, jestli se spojení má ukončit (hodnota 0), nebo zahájit (hodnota 1)
NID_C	unsigned short	Identifikátor země nebo regionu (není relevantní, pokud NID_RBC má hodnotu <i>unknown</i>)
NID_RBC	unsigned short	Identifikátor RBC
NID_RADIO	unsigned long long	Odběrové číslo RBC (není relevantní, pokud NID_RBC má hodnotu <i>unknown</i>)
Q_SLEEPSESSION	unsigned int	Značí, jestli mají být informace v tomto paketu zohledněny v případě, že vozidlová část je ve stavu <i>Sleeping</i>

4.3.2 Paket 49: List of balises for SH Area

Paket se posílá ve zprávě 3: Movement Authority nebo ve zprávě 28: SH Authorised. Obsahuje seznam balíz, které je možné přejet v módu SH.

■ **Tabulka 4.20** Struktura paketu 49: List of balises for SH Area

Proměnná	Datový typ	Poznámka
NID_PACKET	PacketID (int)	Pevná hodnota 49
Q_DIR	unsigned short	Specifikuje, pro který směr jízdy je informace platná
L_PACKET	unsigned short	Délka paketu v bytech
N_ITER	unsigned short	Počet data setů, které následují ve zprávě. V tomto paketu odpovídá počtu balíz.
Q_NEWCOUNTRYs	std::vector<unsigned short>	Seznam indikátorů, který značí, jestli je balízová skupina ve stejné zemi jako předchozí v tomto paketu (0 - Stejná země, 1 - Jiná země).
NID-Cs	std::vector<unsigned short>	Seznam identifikátorů země, ve které se balízová skupina nachází. Validní pouze pokud se indikátor Q_NEWCOUNTRY pro danou balízovou skupinu rovná 1.
NID-BGs	std::vector<unsigned int>	Seznam identifikátorů balízových skupin.

4.3.3 Paket 132: Danger for Shunting Information

Pokud se tento paket vyskytuje v balízovém telegramu, který byl přijat z balízy, kterou vlak právě přejel, a ETCS je v módu SH, pak vlak přejde do módu TR.

■ **Tabulka 4.21** Struktura paketu 132: Danger for Shunting Information

Proměnná	Datový typ	Poznámka
NID_PACKET	PacketID (int)	Pevná hodnota 132
Q_DIR	unsigned short	Specifikuje, pro který směr jízdy je informace platná
L_PACKET	unsigned short	Délka paketu v bytech
Q_ASPECT	unsigned short	Určuje, jestli je možné jet dál v případě, že ETCS je v módu SH (0 - zastavit, pokud je mód SH (<i>stop if in SH mode</i>), 1 - jet, pokud je mód SH)

4.3.4 Paket 138: Reversing area information

Jeden z volitelných paketů, který dává EVC informaci o oblasti, kde je dovolené couvat.

■ **Tabulka 4.22** Struktura paketu 138: Reversing area information

Proměnná	Datový typ	Poznámka
NID_PACKET	PacketID (int)	Pevná hodnota 138
Q_DIR	unsigned short	Specifikuje, pro který směr jízdy je informace platná
L_PACKET	unsigned short	Délka paketu v bytech
Q_SCALE	Q_Scale (int)	Určuje měřítko vzdálenosti (10 cm, 1 m, 10 m)
D_STARTREVERSE	unsigned short	Vzdálenost do začátku oblasti, kde je dovolené couvání
L_REVERSEAREA	unsigned short	Délka oblasti, kde je dovolené couvání

4.3.5 Paket 139: Reversing supervision information

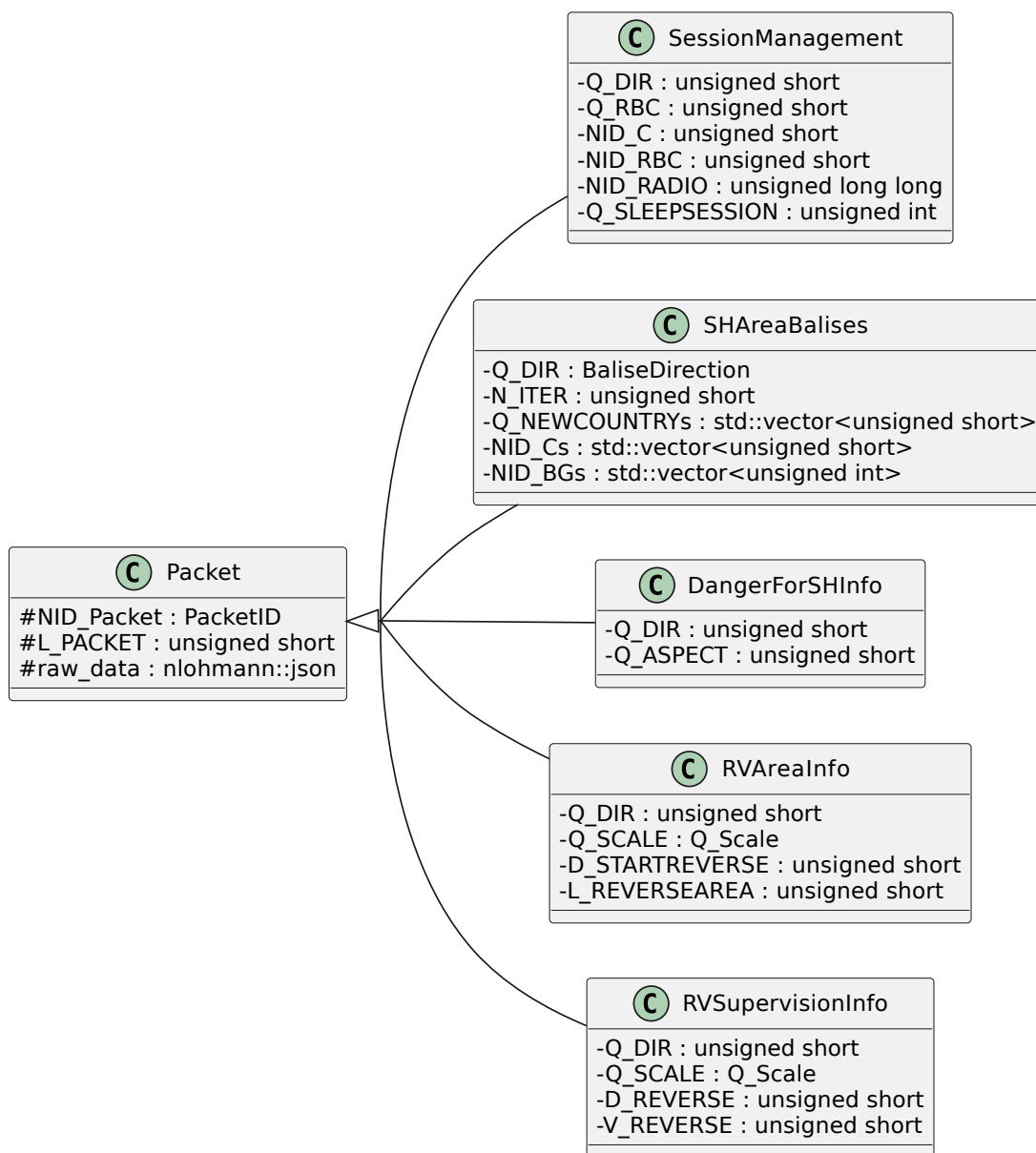
Jeden z volitelných paketů, který dává EVC informace o omezeních, která platí v módu RV.

■ **Tabulka 4.23** Struktura paketu 139: Reversing supervision information

Proměnná	Datový typ	Poznámka
NID_PACKET	PacketID (int)	Pevná hodnota 139
Q_DIR	unsigned short	Specifikuje, pro který směr jízdy je informace platná
L_PACKET	unsigned short	Délka paketu v bytech
Q_SCALE	Q_Scale (int)	Určuje měřítko vzdálenosti (10 cm, 1 m, 10 m)
D_REVERSE	unsigned short	Maximální vzdálenost, kterou je možné ujet v módu RV
V_REVERSE	unsigned short	Maximální rychlost, kterou je možné jet v módu RV

4.3.6 Repräsentace paketů v EVC

Obdobně jako zprávy, i pakety jsou reprezentovány pomocí tříd, které dědí od třídy Packet. Hierarchie tříd, které reprezentují pakety, je výrazně jednodušší než hierarchie zpráv. Pakety se dále nečlení do žádných logických skupin, tudíž není potřeba vytvářet složitější strukturu.



■ **Obrázek 4.15** Class diagram tříd reprezentujících pakety

4.4 Podmínky přechodu nepokryté procedurami

Velká část podmínek přechodu nastává během procedur, které byly popsány (4.1). Návrh implementace těch podmínek, které se procedur netýkají, je v této sekci.

4.4.1 Komunikace s TIU

Od komponenty TIU dostává EVC pravidelně zprávy, které obsahují informace o tom, jestli je zařízení napájené (proměnná *battery_power*) a jestli je stanoviště aktivní (proměnná *cab*). Při zpracování zprávy od TIU bude případně docházet k přechodům do módů SB a NP podle podmínek přechodu 28 a 29.

4.4.2 Komunikace s BTM

V případě, že balízový telegram obsahuje paket *132: Danger for Shunting information* a ETCS se nachází v módu SH, vlak přechází do módu TR. Tato kontrola bude probíhat při zpracování zprávy od BTM. Pokud je vlak v módu SH, tak se také bude kontrolovat, že každá balíza, kterou přejede, je obsažena v seznamu očekávaných balíz. Seznam bude uložen v *ShuntingService*.

Podmínka přechodu 65, která se týká verze jazyka ETCS nebude implementována. Nejedná se o přechod, který by byl klíčový pro běžné fungování systému. Tato funkcionality je alespoň prozatím mimo rámec projektu.

4.4.3 Exit Shunting

Pokud se systém nachází v módu SH a od DMI přijde zpráva 308: *Shunting selected*, kde je proměnná *shunting* nastavená na *false*, nastává přechod do módu SB.

4.5 Služby

Nově přidaná výpočetní logika a s ní spojené ukládání dat budou součástí nových služeb, aby se dodržel princip oddělení spolu nesouvisejících funkcionalit. Jedná se hlavně o data spojená s módy SH a RV, kontroly pohybu vlaku a zpracování zpráv o nouzovém zastavení.

4.5.1 ShuntingService

Nová service, která bude ukládat data spojená s módem SH. Hlavně se jedná o seznam balíz, které je možné projet v módu SH. Tento seznam se bude kontrolovat pokaždé, když EVC přijme balízový telegram, který se zpracovává v metodě třídy *BaseBTMMessageHandler*.

4.5.2 ReversingService

Bude obsahovat informace, které se týkají módu RV. Jedná se o informace o oblastech, kde je dovolené přejít do módu RV a omezení pro rychlost a ujetou vzdálenost, které platí v módu RV.

Pro reprezentaci oblasti, kde je povolené couvání bude vytvořena třída *ReversingArea*. Ta bude obsahovat metodu *TrainIsInside* na zjištění informace, jestli se vlak v této oblasti nachází a metodu *UpdateDistanceTo*, která v parametru přijme vzdálenost, kterou vlak ujel a aktualizuje vzdálenost oblasti od vlaku. Tato metoda se bude volat při každé změně pozice vlaku v metodě *HandleMessage* třídy *ODOMeasurementsHandler*.

4.5.3 EmergencyStopsService

Bude ukládat informace, které se týkají zpráv o nouzovém zastavení. Metody, které bude obsahovat, slouží pro práci s uloženými daty.

- **void AddEmStop(const EmergencyStop & stop)** Uloží informace o zprávě o nouzovém zastavení, která je předaná v parametru.
- **void RevokeEmStop(const unsigned int & id)** Odvolá zprávu o nouzovém zastavení s daným identifikačním číslem.
- **bool EmStopsToBeRevoked()** Vrací boolovskou hodnotu podle toho, jestli je potřeba odvolat nějaká nouzová zastavení.
- **void UpdateStops(const int & traveledDistance)** Metoda, která aktualizuje vzdálenosti jednotlivých oblastí od vlaku.

4.5.4 ModeProfileService

Služba, která se stará o uchovávání dat, které vlak obdrží v paketu *Mode profile*. Jedná se o informace o oblastech, kde je nařízený mód SH nebo OS. Informace jednotlivých oblastech budou reprezentovány třídou *ModeArea*.

Bude obsahovat následující metody:

- **bool IsValid()** Vrací boolovskou hodnotu, která nese informaci o tom, jestli jsou uložená data platná.
- **void SetData(ModeProfile & packet)** Uloží data z přijatého paketu.
- **void DeleteData()** Vymaže data.
- **void UpdateDistance(const double & traveledDistance)** Metoda, která aktualizuje vzdálenosti jednotlivých oblastí od vlaku.
- **std::optional<ModeArea> GetCurrentArea()** Vrací strukturu *optional<ModeArea>*. Ta obsahuje informace o oblasti, ve které se vlak nachází, případně je prázdná, pokud se vlak právě v žádné oblasti nenachází.
- **std::optional<ModeArea> GetNextArea()** Vrací strukturu *optional<ModeArea>*. Obsahuje informace o následující oblasti. Pokud již není před vlakem žádná oblast, je prázdná.

4.5.5 MonitoringService

Pomocná service obsahující metody, které budou provádět kontroly spojené s rychlostí a pozicí vlaku. Tyto metody se budou volat v metodě *ODOMeasurementsHandler::HandleMessageBody*, která zpracovává zprávy od komponenty ODO. Tyto zprávy obsahují informace o rychlosti a pohybech vlaku. Díky přidání této služby bude možné zpřehlednit a strukturalizovat kód, který se nachází v metodě *HandleMessageBody* třídy *ODOMeasurementsHandler*, a zároveň přidat nové funkcionality.

MonitoringService bude obsahovat následující metody:

- **void CheckDistance()** Zajišťuje kontrolu maximální ujeté vzdálenosti (v módu RV). Informace o ujeté vzdálenosti se v metodě přepočítá, porovná s maximem a uloží. Veškerá data týkající se módu RV se ukládají v *ReversingService*.

- **void RollAwayProtection()** Zajišťuje RAP. Kontroluje pozici směrového ovladače a směr pohybu vlaku. V trenažéru se v případě nechtěného pohybu spustí provozní brzdy.
- **void ResetRAP()** Pokud vlak brzdí z důvodu RAP, pak ukončí brždění a znovu nastaví RAP.
- **void ReverseMovementProtection()** Zajišťuje RMP. Spustí brzdy, pokud zaznamená pohyb ve směru opačném k povolenému pohybu.
- **void ResetRMP()** Pokud vlak brzdí z důvodu RMP, pak ukončí brždění a znovu nastaví RMP.
- **void StandstillSupervision()** Pokud je EVC v módu SB a detekuje se jakýkoliv pohyb vlaku, spustí se brzdy.
- **void ResetStandstillSupervision()** Pokud vlak brzdí z důvodu standstill supervision, pak ukončí brždění a znovu nastaví standstill supervision.
- **void ReversingPossible()** Zajišťuje, že strojvedoucí je notifikován v případě, že je možné couvání.
- **void CheckTrip()** Po úplném zastavení vlaku v případě přechodu do módu TR vyžádá potvrzení od strojvedoucího.
- **void ManageTAFRequest()** Zajišťuje správné zobrazování požadavku TAF strojvedoucímu.
- **void CheckModeArea()** Provede příslušné kroky při přechodu do oblasti, kde je vyžádaný nějaký mód.
- **void CheckNextModeArea()** Nastavuje začátek následující oblasti jako EoA a při přechodu do oblasti, kde je možné potvrzení, zašle oznámení strojvedoucímu.

4.6 Přijímání zpráv

Pro všechny nově vytvořené zprávy bude nutné vytvořit i komplementární handlers, které budou zprávy zpracovávat. Vzniknou tedy třídy:

- src/MessageHandlers/DMI
 - ExitSHMessageHandler
 - ModeChangeAckMessageHandler
 - RevokeBrakeCmdAckMessageHandler
 - SHSelectedMessageHandler
- src/MessageHandlers/RBC
 - CondEmStopMessageHandler
 - RevocationEmStopMessageHandler
 - RecogExitTRMessageHandler
 - SHRefusedMessageHandler
 - SHAuthorisedMessageHandler
 - TripAckMessageHandler
 - UncondEmStopMessageHandler

Implementace

V této kapitole je popis implementace navržených služeb, zpráv a jejich zpracování. Mimo tříd pro zprávy, handlers a služby je také představena třída AsyncProperty, která se používá ve službách, a její metoda Invoke, která se používá pro bezpečnou práci s daty, které služby ukládají, ve více vláknech.

5.1 Implementace zpráv

Pro navržené zprávy ze sekce 4.2 byly v EVC vytvořeny třídy, které je reprezentují. Třídy zpráv vždy obsahují konstruktor, který nastaví proměnné podle vstupních argumentů, metody na získání hodnoty proměnných („getter“) a metody from_json a to_json, které slouží pro konverzi zpráv do formátu JSON a obráceně. Metody pro konverzi zpráv jsou generovány makrem NLOHMANN_DEFINE_TYPE_INTRUSIVE z knihovny nlohmann/json [10], jako je tomu například u třídy ModeChangeAckMessage (viz Výpis kódu 5.1), případně byly napsány bez použití makra, jako je tomu například u třídy SHAuthorisedMessage (viz Výpis kódu 5.2).

```
1 /**
2  *      Message 309 (Mode change acknowledged)
3  */
4 class ModeChangeAckMessage : public Message {
5 public:
6     ModeChangeAckMessage();
7     ModeChangeAckMessage(EVCMode mode, nlohmann::json data);
8
9     EVCMode GetMode() const;
10
11     NLOHMANN_DEFINE_TYPE_INTRUSIVE(ModeChangeAckMessage, NID_MESSAGE, mode)
12 private:
13     EVCMode mode;
14 };
```

■ **Výpis kódu 5.1** Deklarace třídy ModeChangeAckMessage

```

1  /**
2   *      Message 28 (SH Authorised)
3   */
4  class SHAuthorisedMessage : public MessageWithPacketsFromRBC {
5  public:
6      SHAuthorisedMessage();
7      SHAuthorisedMessage(unsigned int length, unsigned int timestamp,
8                          nlohmann::json data, unsigned short mAck,
9                          unsigned int nidLrbg, std::vector<Packet> packets,
10                         unsigned int timestampRqst);
11
12     unsigned int GetTTrainRqst() const;
13
14     friend void to_json(nlohmann::json& j, const SHAuthorisedMessage& m);
15     friend void from_json(const nlohmann::json& j, SHAuthorisedMessage& m);
16 private:
17     /**
18      * Time stamp of the shunting request.
19      *
20      * In the subset called T_TRAIN. Renamed to T_TRAIN_RQST, because
21      * MessageRBC::T_TRAIN already exists.
22      */
23     unsigned int T_TRAIN_RQST;
24 };

```

■ **Výpis kódu 5.2** Deklarace třídy SHAuthorisedMessage

5.2 Implementace paketů

Pro pakety navržené v sekci 4.3 byly vytvořeny třídy, které je reprezentují. Tyto třídy jsou obdobou tříd, které implementují zprávy. Rozdílem samozřejmě je, že třídy pro pakety dědí od třídy Packet. Podobně, jako u zpráv, obsahují konstruktor, gettery a metody pro konverzi do formátu JSON. Vzhledem k podobnosti s třídami, které reprezentují zprávy, není potřeba uvádět výpis kódu.

5.3 Implementace služeb

Byly implementovány služby podle návrhu v sekci 4.5. Pro každou službu bylo vytvořeno rozhraní a třída, která toto rozhraní implementuje. Příkladem toho, jak vypadá rozhraní služby, je výpis kódu 5.3, který ukazuje rozhraní pro ReversingDataService.

```

1 class IReversingDataService : public IService {
2 public:
3     IReversingDataService() : IService() {};
4
5     virtual unsigned short GetRVMaxDistance() const = 0;
6     virtual unsigned short GetRVMaxSpeed() const = 0;
7
8     virtual void SetRVMaxDistance(const unsigned short & distance) = 0;
9     virtual void SetRVMaxSpeed(const unsigned short & speed) = 0;
10
11    virtual void AddRVArea(const ReversingArea & area) = 0;
12    virtual void UpdateAreas(const int & distanceTraveled) = 0;
13    virtual bool TrainIsInsideRVArea() const = 0;
14
15    virtual void DeleteRVAreaInfo() = 0;
16    virtual void DeleteRVSuperInfo() = 0;
17
18    static constexpr ServiceType Type = ServiceType::Reversing;
19 };

```

■ Výpis kódu 5.3 Rozhraní IReversingDataService

Třída ReversingDataService (viz Výpis kódu 5.4) implementuje toto rozhraní. Implementuje jednotlivé metody a obsahuje privátní proměnné, které ukládají potřebná data.

```

1 class ReversingDataService : public IReversingDataService, public IInitializable {
2 public:
3     unsigned short GetRVMaxDistance() const override;
4     unsigned short GetRVMaxSpeed() const override;
5
6     void SetRVMaxDistance(const unsigned short & distance) override;
7     void SetRVMaxSpeed(const unsigned short & speed) override;
8
9     void AddRVArea(const ReversingArea & area) override;
10    void UpdateAreas(const int & distanceTraveled) override;
11    bool TrainIsInsideRVArea() const override;
12
13    void DeleteRVAreaInfo() override;
14    void DeleteRVSuperInfo() override;
15 protected:
16    void Initialize(ServiceContainer& container) override;
17 private:
18    // Distance in cm.
19    AsyncProperty<unsigned short> rv_max_distance;
20    // Speed in km/h.
21    AsyncProperty<unsigned short> rv_max_speed;
22    // Information about reversing areas.
23    AsyncProperty<std::vector<ReversingArea>> rv_areas;
24 };

```

■ Výpis kódu 5.4 Deklarace třídy ReversingDataService

5.3.1 Šablona třídy AsyncProperty

Veškeré proměnné, které služba obsahuje jsou zabalené do šablony třídy AsyncProperty. Protože výpočet běží najednou ve více vláknech, je potřeba zajistit, aby v jeden okamžik k proměnným přistupovalo vždy jen jedno vlákno. Šablona třídy obsahuje jako privátní proměnné hodnotu, kterou obaluje, a zámek (viz Výpis kódu 5.5).

Již dříve byly implementovány metody pro práci se základními typy, ale chyběla možnost používat AsyncProperty pro složitější struktury, jako je například std::vector nebo std::map. Proto byla v rámci této práce implementována metoda AsyncProperty::Invoke (viz Výpis kódu 5.6). Vstupem do metody je funkce a její argumenty. Metoda získá zámek pro danou proměnnou a zavolá funkci, které jako první argument předá vnitřní proměnnou value. Následují další argumenty, které byli vstupem do metody Invoke. Díky této metodě je možné bezpečně volat libovolné funkce s proměnnou, která je zabalená ve třídě AsyncProperty.

```
1 template <typename T>
2 class AsyncProperty {
3 public:
4     ...
5 private:
6     std::mutex own_mutex;
7     std::mutex* mutex;
8     T value;
9 };
```

■ **Výpis kódu 5.5** Privátní proměnné třídy AsyncProperty

```
1 template<typename F, typename ... Param>
2 auto Invoke(const F& func, const Param& ... param) {
3     std::lock_guard<std::mutex> lock(*mutex);
4     return func(value, param...);
5 }
6
7 template<typename F, typename ... Param>
8 auto Invoke(const F& func, const Param& ... param) const {
9     std::lock_guard<std::mutex> lock(*mutex);
10    return func(value, param...);
11 }
```

■ **Výpis kódu 5.6** Implementace metody AsyncProperty::Invoke

5.4 Implementace zpracování zpráv

Zprávy, které EVC obdrží od ostatních komponent, jsou zpracovávány pomocí handlerů. Handler je vždy zodpovědný za zpracování jedné konkrétní zprávy a pokaždé, když EVC tuto zprávu obdrží, se zavolá metoda `HandleMessageBody`, která v argumentu dostane přijatou zprávu. V této metodě se provedou příslušné kroky, podle toho, o jakou zprávu se jedná a jaký je její obsah.

Příkladem třídy, která zpracovává zprávy, je `ModeChangeAckMessageHandler` (viz Výpis kódu 5.7). Tato třída je handlerem pro zprávu 309: Mode change acknowledged. Pro zpracování zprávy využívá tato třída služby. Ukazatele na tyto služby jsou uloženy v privátních proměnných a jsou inicializovány v konstruktoru třídy (viz Výpis kódu 5.8). Metoda `HandleMessageBody` (viz Výpis kódu 5.9) reaguje na přijatou zprávu. V tomto konkrétním příkladě se jedná o potvrzení přechodu do módu od strojvedoucího. Podle toho, jaký mód byl potvrzen, se provedou adekvátní kroky.

```

1 class ModeChangeAckMessageHandler : public MessageHandler {
2 public:
3     ModeChangeAckMessageHandler(ServiceContainer& container);
4 protected:
5     void HandleMessageBody(Message& message) override;
6 private:
7     void SendTrainPositionReport();
8
9     std::shared_ptr<IEVCGeneralDataService> general_data_service;
10    std::shared_ptr<ITIUDataService> tiu_data_service;
11    std::shared_ptr<IMonitoringService> monitoring_service;
12    std::shared_ptr<IBaliseGroupsDataService> balise_groups_data_service;
13    std::shared_ptr<ISpeedDataService> speed_data_service;
14    std::shared_ptr<ITrainDataService> train_data_service;
15    std::shared_ptr<IPublisherService> publisher_service;
16    std::shared_ptr<IDMCommunicationDataService> dmi_service;
17 };

```

■ Výpis kódu 5.7 Deklarace třídy `ModeChangeAckMessageHandler`

```

1 ModeChangeAckMessageHandler::ModeChangeAckMessageHandler(ServiceContainer& container)
2 : MessageHandler(MessageID::ModeChangeAckDMIToEVC, container) {
3     general_data_service = container.FetchService<IEVCGeneralDataService>();
4     tiu_data_service = container.FetchService<ITIUDataService>();
5     monitoring_service = container.FetchService<IMonitoringService>();
6     balise_groups_data_service = container.FetchService<IBaliseGroupsDataService>();
7     speed_data_service = container.FetchService<ISpeedDataService>();
8     train_data_service = container.FetchService<ITrainDataService>();
9     publisher_service = container.FetchService<IPublisherService>();
10    dmi_service = container.FetchService<IDMCommunicationDataService>();
11 }

```

■ Výpis kódu 5.8 Konstruktor třídy `ModeChangeAckMessageHandler`

```

1 void ModeChangeAckMessageHandler::HandleMessageBody(Message& message) {
2     ModeChangeAckMessage msg = message.GetRawData().get<ModeChangeAckMessage>();
3     EVCMode originalMode = general_data_service->GetMode();
4     dmi_service->SetLastReceivedMessageID(msg.GetMessageID());
5
6     if (msg.GetMode() == EVCMode::TRIP) {
7         monitoring_service->SetTripBraking(false);
8         tiu_data_service->SetEmergencyBrake(false);
9         helper_service->SendTIUGeneralUpdate();
10        switch (general_data_service->GetLastActiveLevel()) {
11            case 0: {
12                helper_service->ChangeMode(EVCMode::UNFITTED);
13            }
14            default: {
15                helper_service->ChangeMode(EVCMode::POST_TRIP);
16                monitoring_service->StartCheckDistance();
17            }
18        }
19    }
20    if (msg.GetMode() == EVCMode::ON_SIGHT) {
21        tiu_data_service->SetEmergencyBrake(false);
22        helper_service->ChangeMode(EVCMode::ON_SIGHT);
23    }
24    if (msg.GetMode() == EVCMode::SHUNTING) {
25        tiu_data_service->SetEmergencyBrake(false);
26        helper_service->ChangeMode(EVCMode::SHUNTING);
27        if (general_data_service->GetLastActiveLevel() == 2
28            || general_data_service->GetLastActiveLevel() == 3) {
29            std::vector<Packet> packets;
30            unsigned int currentTime = time_service->GetCurrentTimestamp();
31            EndOfMissionMessage endOfMissionMessage(
32                74, currentTime, nlohmann::json(),
33                train_data_service->GetNIDEngine(), packets
34            );
35            endOfMissionMessage.SetRawData(endOfMissionMessage);
36            publisher_service->PublishMessage(
37                MessageToSend(to_string(MQTTTopics::TORBC), endOfMissionMessage)
38            );
39            rbc_communication_service->SetLastSentMessageID(MessageID::EndOfMission);
40            rbc_communication_service->SetLastSentMessageTime(currentTime);
41        }
42    }
43
44    // when changing a mode, reset RAP, RMP and Standstill supervision starting point
45    if (originalMode != msg.GetMode()) {
46        monitoring_service->ResetRAP();
47        monitoring_service->ResetRMP();
48        monitoring_service->ResetStandstillSupervision();
49    }
50 }

```

■ Výpis kódu 5.9 Metoda HandleMessageBody třídy ModeChangeAckMessageHandler

Tato kapitola popisuje, jakým způsobem bylo implementované řešení testováno. Hlavní složkou testování bylo provedení testovacích scénářů, pomocí kterých bylo zkontrolováno, že EVC reaguje na přijímané zprávy správným způsobem a že stav systému se mění podle očekávání.

6.1 Testovací scénáře

Testovací scénáře popisují, jaké zprávy se posílají EVC a jak má EVC na tyto zprávy reagovat. Je vysvětleno, jak se mění vnitřní stav v závislosti na přijatých zprávách. Ke každému scénáři přísluší obrázek, na kterém je znázorněná adresářová struktura testovacích zpráv, které se v daném scénáři používají. U každého souboru zprávy je číslo v hranatých závorkách (např. [7]), které určuje pořadí poslané zprávy v rámci testovacího scénáře. U zpráv je podle jejich identifikátoru možné rozlišit, které komponentě se posílají, případně, od které komponenty přišly (viz 4.2.1).

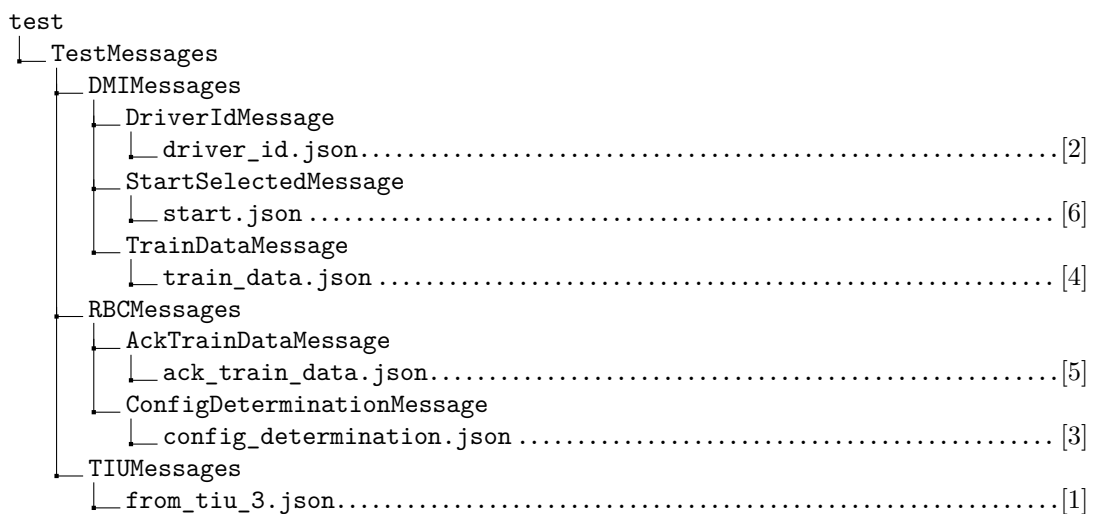
6.1.1 Testovací zprávy

U některých testovacích zpráv je nutné před jejich posláním vyplnit proměnné týkající se časových značek. Jedná se o zprávy z adresáře `test/TestMessages/RBCMessages/MovementAuthorityMessage`, u kterých hodnota proměnné `T_TRAIN` musí být větší než časová značka zprávy, která byla odeslána RBC jako poslední. Dále se to týká zprávy `test/TestMessages/RBCMessages/AckTrainDataMessage/ack_train_data.json`, ve které hodnota proměnné `T_TRAIN` musí být větší než časová značka zprávy, která byla odeslána RBC jako poslední, a hodnota proměnné `T_TRAIN1` musí odpovídat časové značce zprávy, která byla odeslána RBC jako poslední.

6.1.2 Navázání spojení

Velká řada testovacích scénářů vyžaduje, aby bylo nejprve navázané spojení s RBC. Způsob, jakým se spojení naváže, bude popsán v této sekci a ostatní testovací scénáře se na tuto sekci budou odkazovat.

EVC obdrží zprávu 602: From TIU a reaguje přechodem do módu *Stand By*. DMI pošle EVC zprávu 300: Driver id a EVC reaguje zprávu 155: Initiation of communication session. RBC odešle zprávu 32: Configuration determination. EVC potvrdí zahájení spojení zprávu 159: Session established a odešle informaci o zahájení spojení DMI zprávu 202: Session and train data (*session_established = true, train_data_ack = false*). RBC se odešle zpráva 157: SoM Position Report. EVC dostane od DMI zprávu 304: Train data, odpovídá zprávu 202: Session and train data (*session_established = true, train_data_ack = true*) a posílá zprávu 139: Validated train data. Po zmáčknutí tlačítka start (305: Start selected) se odešle RBC zpráva 132: MA Request.



■ Obrázek 6.1 Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.2

6.1.3 Podmínky přechodu 4 a 29

- 4 Vozidlová část ETCS je napájena.
- 29 Vozidlová část ETCS není napájena.

V počátečním stavu je systém v módu *No Power*. V případě, že přijde zpráva z TIU, ve které je proměnná *battery_power* nastavená na *true*, pak systém přejde do módu *Stand By*.

Pokud přijde zpráva, kde je proměnná *battery_power* nastavena na *false*, pak systém opět přejde do módu *No Power*.

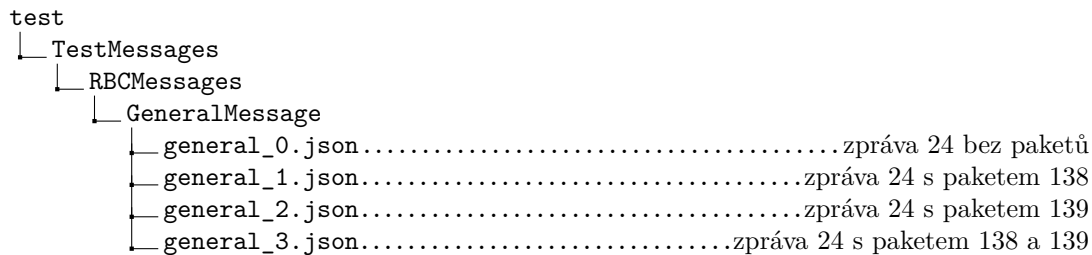


■ Obrázek 6.2 Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.3

6.1.4 Zpracování paketů 138 a 139

- Paket 138: Reversing area information
- Paket 139: Reversing supervision information

Pokud zpráva 24: General message obsahuje paket 138, pak se do ReversingService uloží informace o oblasti, kde je povolené couvání. Pokud zpráva 24: General message obsahuje paket 139, pak se do ReversingService uloží informace o maximální rychlosti a vzdálenosti pro mód *Reversing*.



■ **Obrázek 6.3** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.4

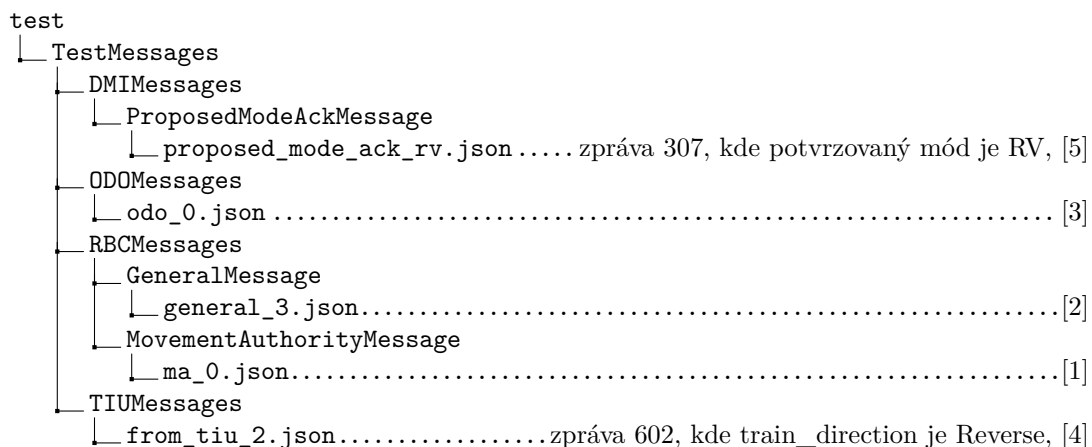
6.1.5 Procedura Reversing

Po zahájení spojení (6.1.2) vlak obdrží zprávu 3: Movement authority, přejde do módu *Full supervision* a obdrží pakety 138: Reversing area information a 139: Reversing supervision information ve zprávě 24: General message.

Vlak přejde do oblasti, kde je povolené couvání. Informace o pohybech vlaku přijdou ve zprávě 481: ODO Measurements. EVC odešle DMI zprávu 211: Reversing possible.

Ve chvíli, kdy od TIU přijde zpráva 602: From TIU, kde je proměnná *train_direction*, která označuje pozici směrového ovladače, nastavena na hodnotu *TrainDirection::Reverse*, EVC odešle DMI zprávu 204: Mode proposed, a tak strojvedoucímu navrhne přechod do módu *Reversing*.

Pokud strojvedoucí přechod potvrdí, EVC přijme zprávu 307: Proposed mode acknowledged a přejde do módu *Reversing*.



■ **Obrázek 6.4** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.5

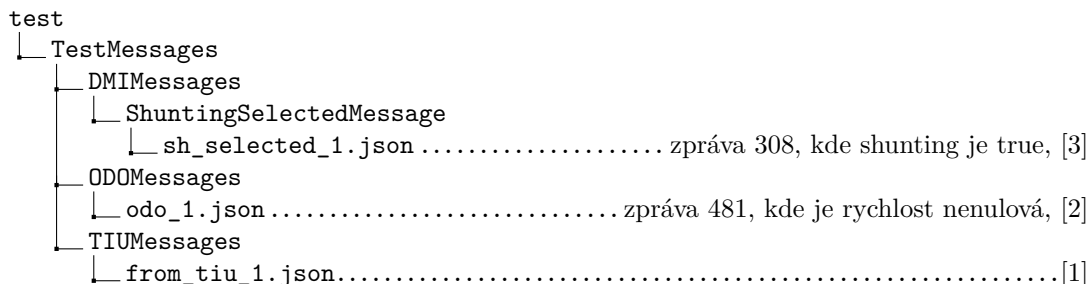
6.1.6 Procedura Shunting Initiated by Driver

6.1.6.1 Vlak nestojí na místě

Vlak se dostane do módu *Stand By* pomocí zprávy 602: From TIU.

Vlak dostane informace o pohybech vlaku ve zprávě 481: ODO Measurements. Hodnota proměnné *V_TRAIN* je větší než nula.

EVC přijme od DMI zprávu 308: Shunting selected s proměnnou *shunting* nastavenou na *true*. Vlak nestojí na místě, tudíž procedura končí.



■ **Obrázek 6.5** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.6.1

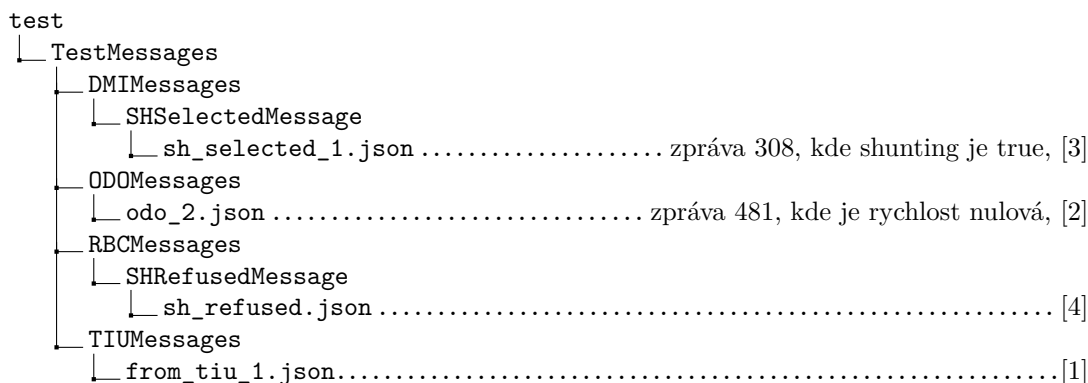
6.1.6.2 Vlak stojí na místě, úroveň ETCS je 2 a RBC nepotvrdí přechod do SH

Vlak se dostane do módu *Stand By* pomocí zprávy 602: From TIU.

Vlak dostane informace o pohybech vlaku ve zprávě 481: ODO Measurements. Hodnota proměnné *V_TRAIN* je nulová.

EVC přijme od DMI zprávu 308: Shunting selected s proměnnou *shunting* nastavenou na *true*. EVC odešle RBC zprávu 130: Request for Shunting a DMI zprávu 210: Waiting for SH request response.

Od RBC přijde zpráva 27: SH Refused. EVC odešle DMI zprávu 208: Paket fixní zprávy a procedura končí s oznámením, že přechod do módu *Shunting* nebyl povolen.



■ **Obrázek 6.6** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.6.2

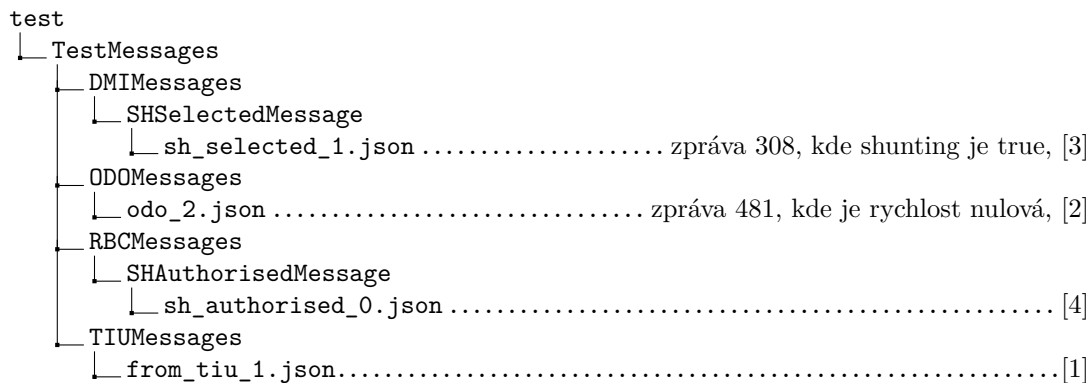
6.1.6.3 Vlak stojí na místě, úroveň ETCS je 2 a RBC potvrdí přechod do SH

Vlak se dostane do módu *Stand By* pomocí zprávy 602: From TIU.

Vlak dostane informace o pohybech vlaku ve zprávě 481: ODO Measurements. Hodnota proměnné *V_TRAIN* je nulová.

EVC přijme od DMI zprávu 308: Shunting selected s proměnnou *shunting* nastavenou na *true*. EVC odešle RBC zprávu 130: Request for Shunting a DMI zprávu 210: Waiting for SH request response.

Od RBC přijde zpráva 28: SH Authorised. EVC přejde do módu *Shunting* a odešle RBC zprávu 150: End of Mission. Dále se provádí procedura End of Mission.



■ **Obrázek 6.7** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.6.3

6.1.7 Procedura End of Mission

Nejprve je potřeba zahájit komunikační spojení (6.1.2).

EVC přijme zprávu 24: General message s paketem 42: Session management, kde je proměnná *Q_RBC* nastavená na hodnotu 0 (to značí žádost o ukončení spojení). EVC odpoví zprávou 156: Termination of communication session. Poté, co přijme zprávu 39: Acknowledgement of a termination of a communication session, je spojení ukončeno.



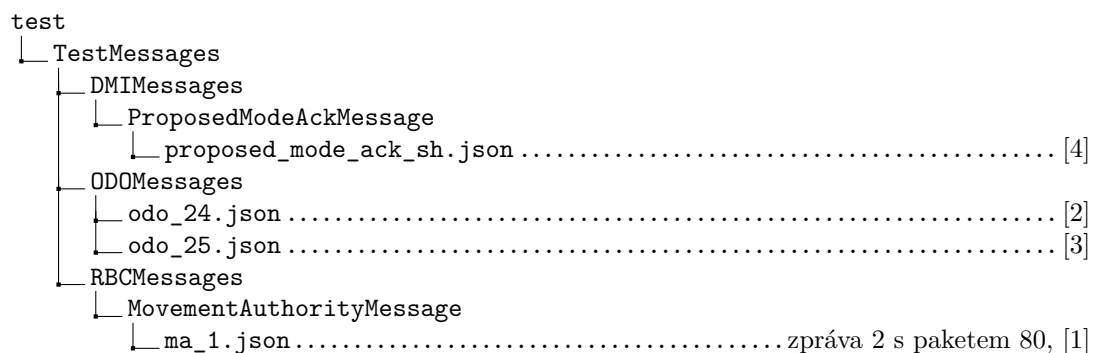
■ **Obrázek 6.8** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.7

6.1.8 Procedura Entry in Shunting

Nejprve je potřeba zahájit spojení (6.1.2).

6.1.8.1 Strojvedoucí potvrdí mód v oblasti pro potvrzení

EVC přijme jako odpověď na zprávu 132: MA Request od RBC zprávu 3: Movement Authority, která obsahuje paket 80: Mode profile. Paket obsahuje informaci o vzdálenosti oblasti, kde je vyžadován mód SH, a délce potvrzovací oblasti, která se nachází před ní. Systém přejde do módu FS. Vlak ujede kus vzdálenosti (odo_24.json) a vstoupí do oblasti pro potvrzení (odo_25.json). DMI se odešle zpráva 204: Mode proposed. DMI odpoví zprávou 307: Proposed mode acknowledged a systém přejde do módu SH. Tento přechod je považován za ukončení mise, tudíž se odešle zpráva 150: End of Mission.



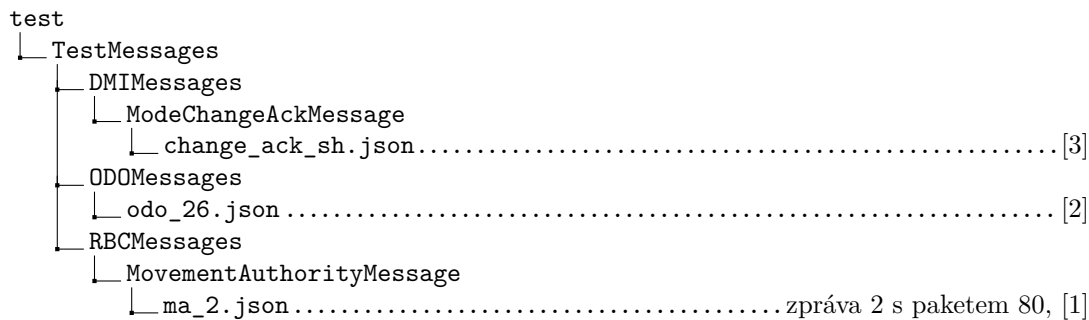
■ **Obrázek 6.9** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.8.1

6.1.8.2 Vlak přejde do oblasti pro mód SH

EVC přijme jako odpověď na zprávu 132: MA Request od RBC zprávu 3: Movement Authority, která obsahuje paket 80: Mode profile. Paket obsahuje informaci o vzdálenosti oblasti, kde je vyžadován mód SH. Vlak se nachází v této oblasti. Protože je systém v módu SB, odešle se zpráva 204: Mode proposed. Vlak se posune dopředu (odo_26.json). Odešle se zpráva 209: Mode change.

6.1.8.2.1 Potvrzení přijde do 5 sekund Do 5 sekund od odeslání zprávy 209 přijde zpráva 309: Mode change acknowledged. Přechod do módu SH je považován za ukončení mise, tudíž se odešle zpráva 150: End of Mission.

6.1.8.2.2 Potvrzení nepřijde do 5 sekund Po 5 sekundách od odeslání zprávy 209 se spustí nouzové brzdy. Nouzové brždění se ukončí, když přijde zpráva 309: Mode change acknowledged. Přechod do módu SH je považován za ukončení mise, tudíž se odešle zpráva 150: End of Mission.



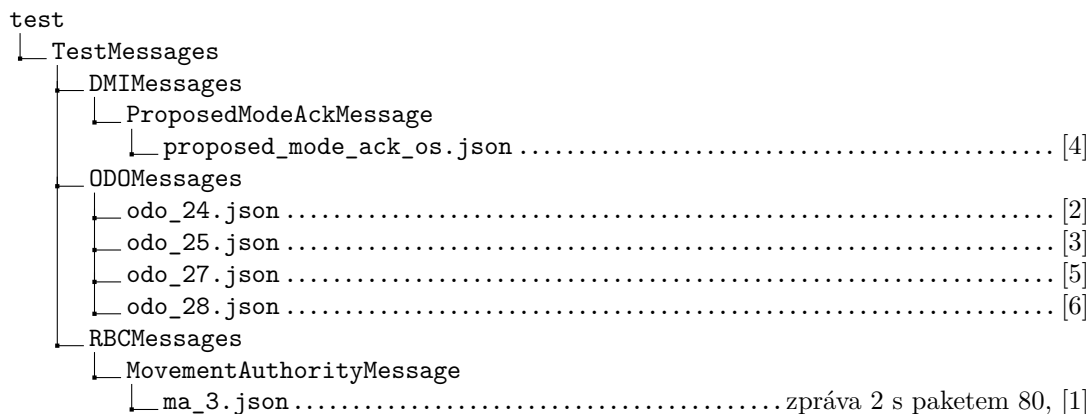
■ **Obrázek 6.10** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.8.2

6.1.9 Procedura On-Sight

Nejprve je potřeba zahájit spojení (6.1.2).

6.1.9.1 Strojvedoucí potvrdí mód v oblasti pro potvrzení

EVC přijme jako odpověď na zprávu 132: MA Request zprávu 3: Movement Authority, která obsahuje paket 80: Mode profile. Paket obsahuje informaci o vzdálenosti oblasti, kde je vyžadován mód OS, a délce potvrzovací oblasti, která se nachází před ní. Systém přejde do módu FS. Vlak ujede kus vzdálenosti (odo_24.json) a vstoupí do oblasti pro potvrzení (odo_25.json). DMI se odešle zpráva 204: Mode proposed. DMI odpoví zprávou 307: Proposed mode acknowledged a systém přejde do módu OS. Vlak dojde na konec oblasti pro mód OS (odo_27.json) a ve chvíli, kdy vlak oblast opustí (odo_28.json) se přejde zpět do módu FS.



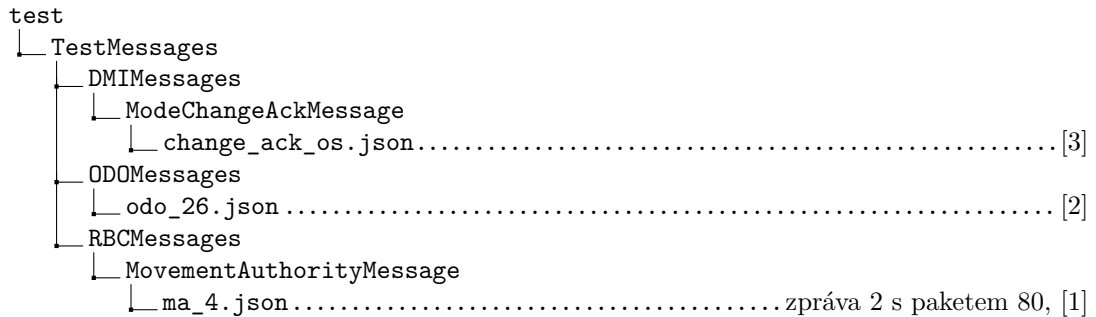
■ **Obrázek 6.11** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.9.1

6.1.9.2 Vlak přejde do oblasti pro mód OS

EVC přijme jako odpověď na zprávu 132: MA Request zprávu 3: Movement Authority, která obsahuje paket 80: Mode profile. Paket obsahuje informaci o vzdálenosti oblasti, kde je vyžadován mód OS. Vlak se nachází v této oblasti. Protože je systém v módu SB, odešle se zpráva 204: Mode proposed. Vlak se posune dopředu (odo_26.json). Odešle se zpráva 209: Mode change.

6.1.9.2.1 Potvrzení přijde do 5 sekund Do 5 sekund od odeslání zprávy 209 přijde zpráva 309: Mode change acknowledged.

6.1.9.2.2 Potvrzení nepřijde do 5 sekund Po 5 sekundách od odeslání zprávy 209 se spustí nouzové brzdy. Nouzové brždění se ukončí, když přijde zpráva 309: Mode change acknowledged.



■ **Obrázek 6.12** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.9.2

6.1.10 Procedura Train trip

Nejprve je potřeba zahájit spojení (6.1.2).

6.1.10.1 Nejsou nařízená nouzová zastavení

EVC přijme jako odpověď na zprávu 132: MA Request zprávu 2: SR Authorisation (proměnná $D_SR = 0$) a 307: Proposed mode acknowledged. Vzdálenost, kterou je možné ujet v módu SR je rovna nule, tudíž systém při přijetí zprávy 481: ODO Measurements s proměnnou D_TRAIN nastavenou na nenulovou hodnotu přejde do módu *Trip*.

Ve chvíli, kdy vlak zastaví, tak se odešle zpráva 209: Mode change ($mode = EVCMODE::TRIP$). Na tu DMI odpoví zprávou 309: Mode change acknowledged. Vlak přejde do módu *Post Trip*. Ukončí se nouzové brždění a odešle se zpráva 136: Train position report. Na tu RBC odpoví zprávou 6: Recognition of exit from TRIP mode. EVC odešle DMI zprávu 213: Waiting for emergency stops revocation, a protože nejsou nařízena žádná nouzová zastavení, odešle zprávu 214: Emergency stops revoked.

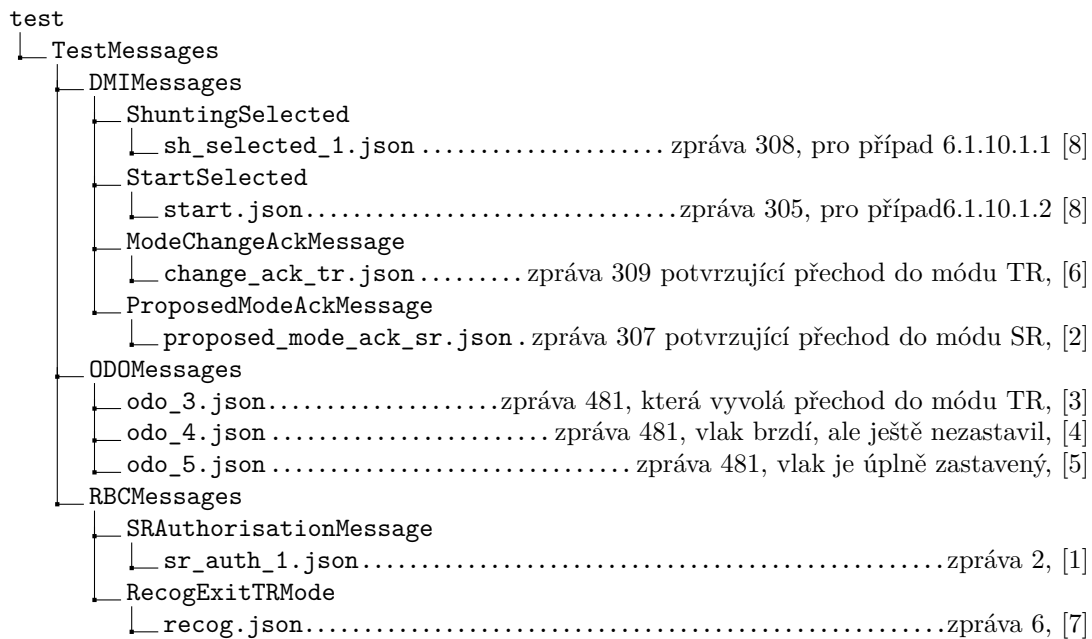
6.1.10.1.1 Strojvedoucí zvolí Shunting Pokud nyní EVC přijme zprávu 308: Shunting selected, zahájí se procedura *Shunting Selected by Driver*.

6.1.10.1.2 Strojvedoucí zvolí Start Pokud EVC přijme zprávu 305: Start selected, EVC odešle zprávu 132: MA Request.

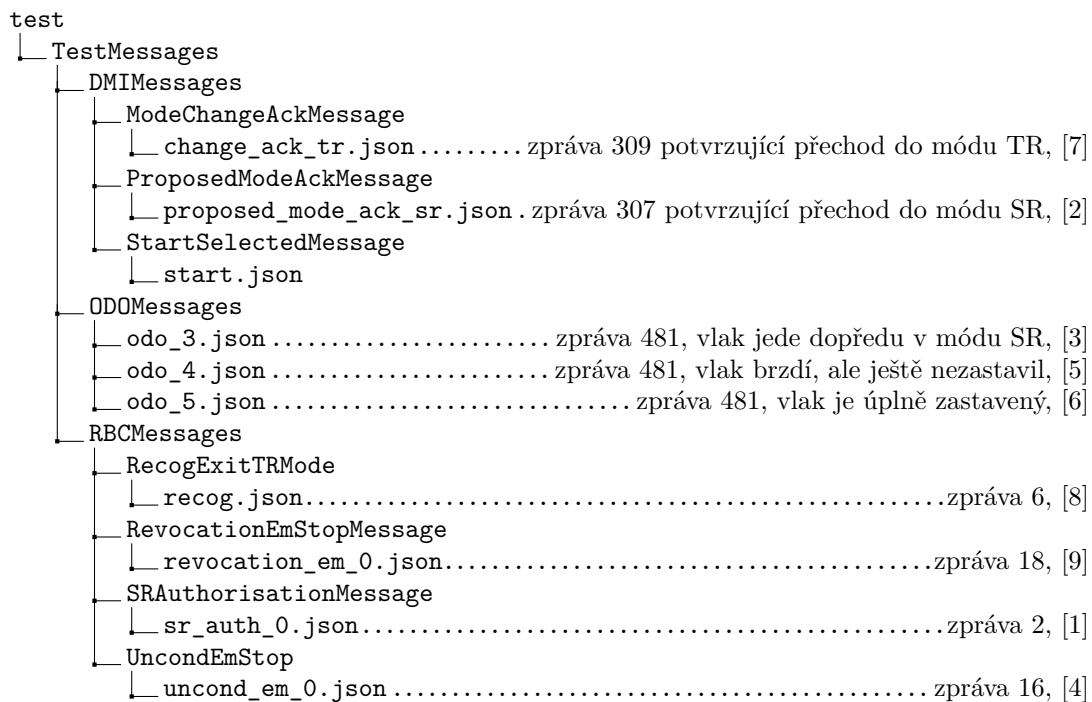
6.1.10.2 Nařízení nouzového zastavení

Vlak se pohybuje směrem dopředu (odo_3.json). Vlak obdrží zprávu 16: Unconditional Emergency Stop přejde do módu *Trip* a spustí nouzové brzdy.

Ve chvíli, kdy vlak zastaví, se odešle zpráva 209: Mode change ($mode = EVCMODE::TRIP$). Na tu DMI odpoví zprávou 309: Mode change acknowledged. Vlak přejde do módu *Post Trip*. Ukončí se nouzové brždění a odešle se zpráva 136: Train position report. Na tu RBC odpoví zprávou 6: Recognition of exit from TRIP mode. EVC odešle DMI zprávu 213: Waiting for emergency stops revocation. Od RBC obdrží zprávu 18: Revocation of Emergency Stop, která zruší zprávu o nouzovém zastavení. Protože nejsou nařízena žádná další nouzová zastavení, odešle zprávu 214: Emergency stops revoked.



■ **Obrázek 6.13** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.10



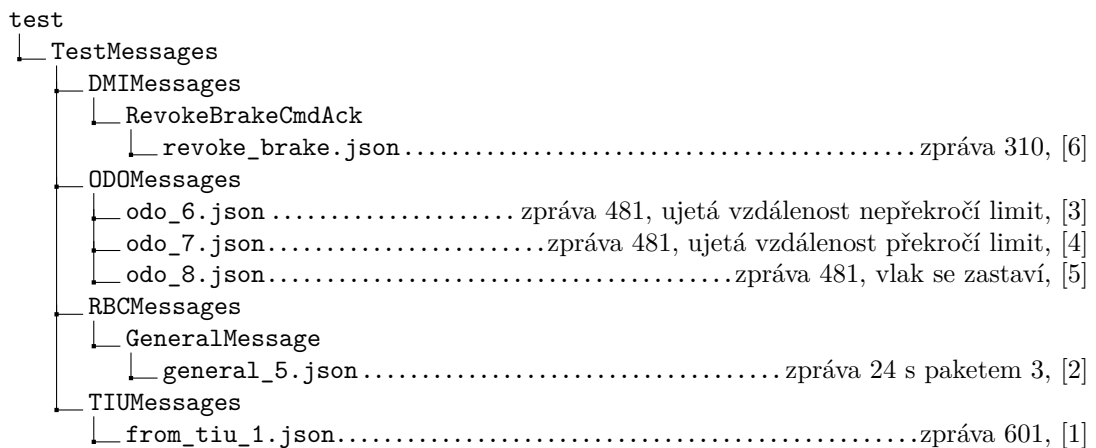
■ **Obrázek 6.14** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.10.2

6.1.11 Standstill supervision

6.1.11.1 Vlak ujíždí směrem dopředu

Vlak přejde do módu *Stand By* pomocí zprávy 601: From TIU a obdrží paket 3: National values ve zprávě 24: General message. Díky tomu se nastaví národní hodnota *D_NVROLL*, která určuje ujetou vzdálenost, kterou vlak nemůže překročit.

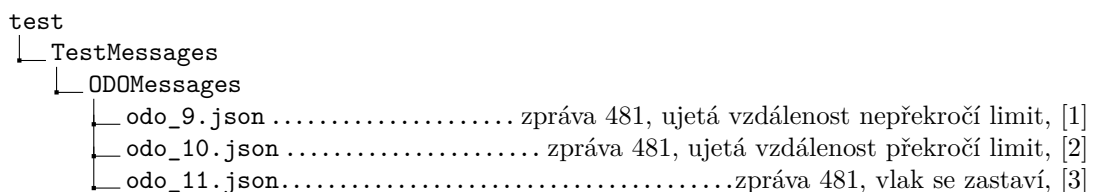
Ve chvíli, kdy vlak v módu *Stand By* ujede vyšší vzdálenost, než tu, která je definovaná proměnnou *D_NVROLL* se spustí provozní brzdy a strojvedoucímu se oznámí důvod brždění zprávou 208: Paket fixní zprávy. Jakmile vlak zastaví, odešle se strojvedoucímu zpráva 212: Revoke brake command, kterou potvrdí zprávou 310: Revoke brake command acknowledged. Ukončí se brždění a Standstill supervision se resetuje.



■ **Obrázek 6.15** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.11.1

6.1.11.2 Vlak ujíždí směrem dozadu

Po resetu Standstill supervision vlak couvá o větší vzdálenost, než je limit. Opět se spustí provozní brzdy a strojvedoucímu se oznámí důvod brždění zprávou 208: Paket fixní zprávy. Jakmile vlak zastaví, odešle se strojvedoucímu zpráva 212: Revoke brake command, kterou potvrdí zprávou 310: Revoke brake command acknowledged. Ukončí se brždění a Standstill supervision se resetuje.



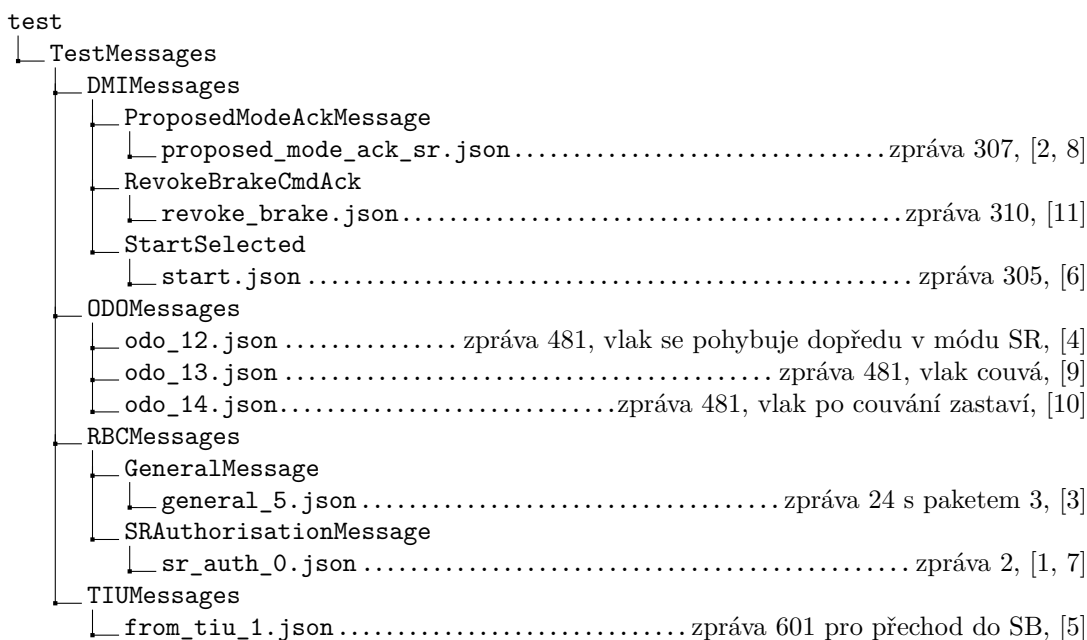
■ **Obrázek 6.16** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.11.2

6.1.12 Reverse movement protection

Nejprve je potřeba zahájit spojení (6.1.2).

Systém přejde do módu SR pomocí zpráv 2: SR Authorisation a 307: Proposed mode acknowledged. EVC obdrží paket 3: National values ve zprávě 24: General message. Vlak ujede nějakou vzdálenost v módu SR (odo_12.json) a poté přejde do módu *Stand By* (from_tiu_1.json).

Strojvedoucí zmáčkne start (305: Start selected). Systém opět přejde do módu SR pomocí zpráv 2 a 307 a ve chvíli, kdy vlak odjede směrem dozadu vzdálenost překračující limit definovaný národní hodnotou (odo_13.json), spustí se servisní brzdy. Ve chvíli kdy vlak zastaví (odo_14.json) se odešle DMI zpráva 212: Revoke brake command. Strojvedoucí odpoví zprávou 310: Revoke brake command acknowledged. Uvolní se servisní brzdy a pozice pro RMP se resetuje.



■ Obrázek 6.17 Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.12

6.1.13 Roll away protection

Nejprve je potřeba zahájit spojení (6.1.2).

Systém přejde do módu SR pomocí zpráv 2: SR Authorisation a 307: Proposed mode acknowledged. EVC obdrží paket 3: National values ve zprávě 24: General message. Vlak ujede nějakou vzdálenost v módu SR (odo_12.json) a poté přejde do módu *Stand By* (from_tiu_1.json).

6.1.13.1 Směrový ovladač v pozici dopředu

Strojvedoucí zmáčkne start (305: Start selected). Systém opět přejde do módu SR pomocí zpráv 2 a 307. Směrový ovladač je v pozici, která značí směr kupředu a ve chvíli, kdy vlak odjede směrem dozadu vzdálenost překračující limit definovaný národní hodnotou (odo_13.json), se spustí servisní brzdy. Poté, co vlak zastaví (odo_14.json), se odešle DMI zpráva 212: Revoke brake command. Strojvedoucí odpoví zprávou 310: Revoke brake command acknowledged. Uvolní se servisní brzdy a pozice pro RAP se resetuje.

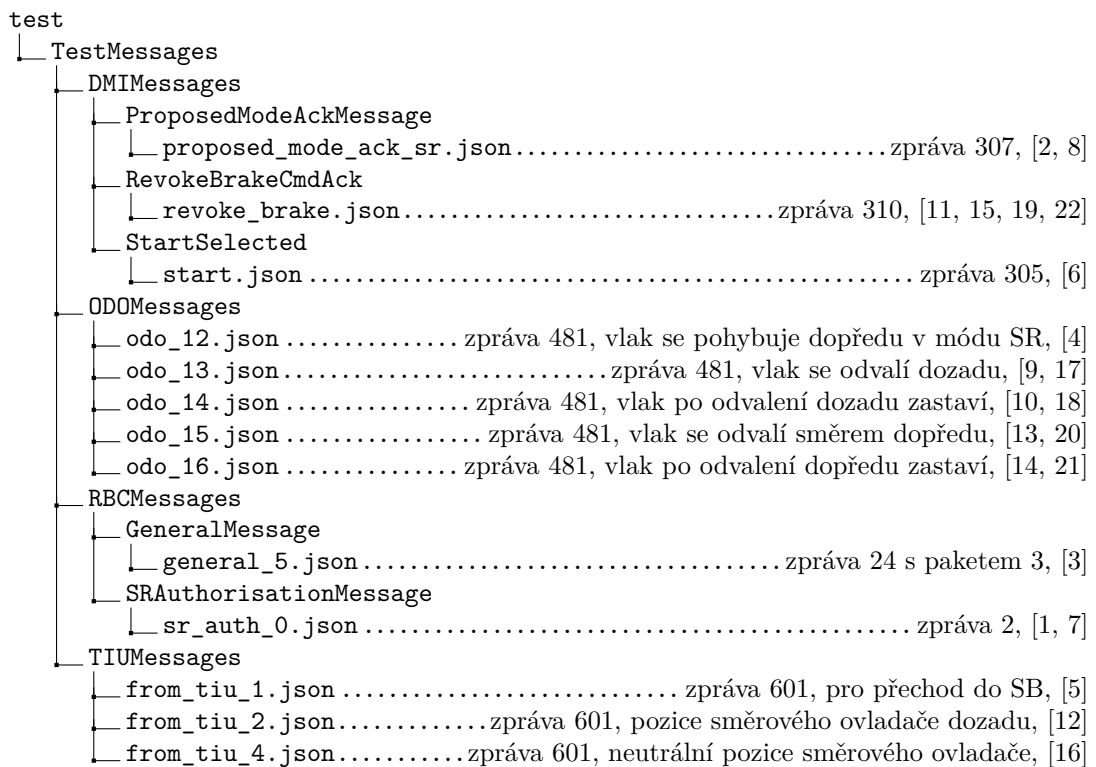
6.1.13.2 Směrový ovladač v pozici dozadu

Strojvedoucí změní pozici směrového ovladače na couvání. Vlak odjede směrem dopředu vzdálenost překračující limit definovaný národní hodnotou (odo_15.json), spustí se servisní brzdy. Ve chvíli kdy vlak zastaví (odo_16.json) se odešle DMI zpráva 212: Revoke brake command. Strojvedoucí odpoví zprávou 310: Revoke brake command acknowledged. Uvolní se servisní brzdy a pozice pro RAP se resetuje.

6.1.13.3 Směrový ovladač v neutrální pozici

Strojvedoucí změní pozici směrového ovladače do neutrální pozice. Vlak odjede směrem dozadu vzdálenost překračující limit definovaný národní hodnotou (odo_13.json), spustí se servisní brzdy. Ve chvíli kdy vlak zastaví (odo_14.json) se odešle DMI zpráva 212: Revoke brake command. Strojvedoucí odpoví zprávou 310: Revoke brake command acknowledged. Uvolní se servisní brzdy a pozice pro RAP se resetuje.

Vlak odjede směrem dopředu vzdálenost překračující limit definovaný národní hodnotou (odo_15.json), spustí se servisní brzdy. Ve chvíli kdy vlak zastaví (odo_16.json) se odešle DMI zpráva 212: Revoke brake command. Strojvedoucí odpoví zprávou 310: Revoke brake command acknowledged. Uvolní se servisní brzdy a pozice pro RAP se resetuje.



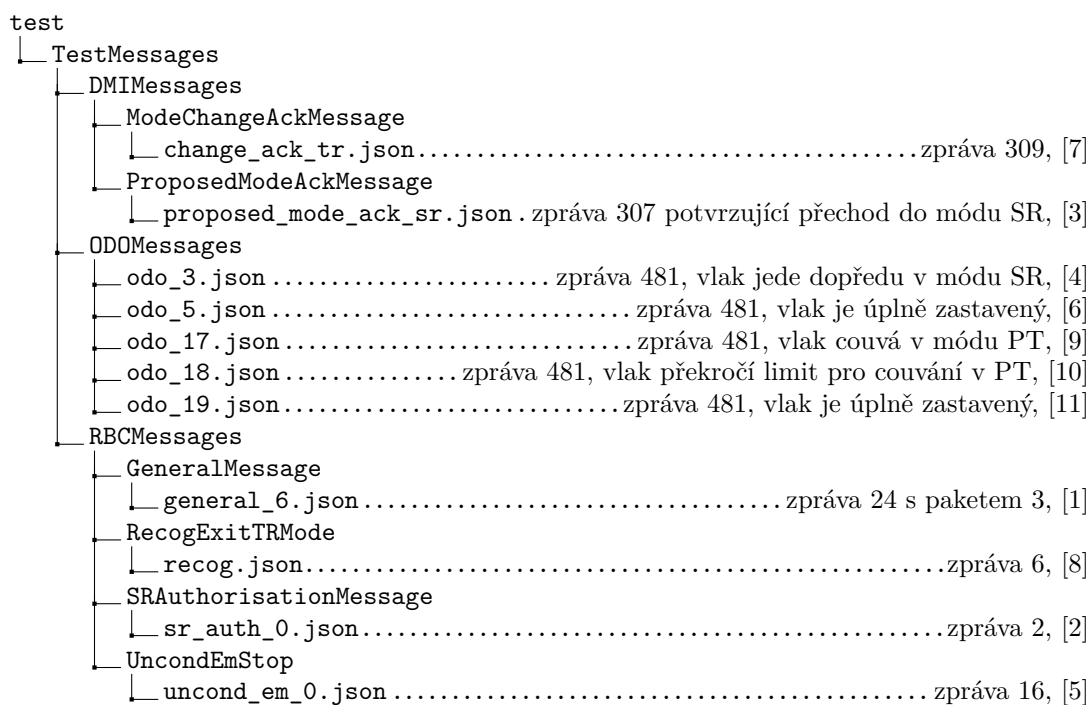
■ Obrázek 6.18 Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.13

6.1.14 Maximální vzdálenost v módu PT

Nejprve je potřeba zahájit spojení (6.1.2).

Vlak dostane ve zprávě 24: General message paket 3: National Values, díky kterému se nastaví limit pro couvání v módu PT. Vlak přejde do módu *Staff responsible* zprávami 2: SR Authorisation a 307: Proposed mode acknowledged. Vlak se pohybuje směrem dopředu (odo_3.json). EVC obdrží zprávu 16: Unconditional Emergency Stop, systém přejde do módu *Trip* a spustí se nouzové brzdy.

Ve chvíli, kdy vlak zastaví, se odešle zpráva 209: Mode change (*mode = EVCMODE::TRIP*). Na tu DMI odpoví zprávou 309: Mode change acknowledged. Vlak přejde do módu *Post Trip*. Ukončí se nouzové brždění a odešle se zpráva 136: Train position report. Na tu RBC odpoví zprávou 6: Recognition of exit from TRIP mode. EVC odešle DMI zprávu 213: Waiting for emergency stops revocation. Vlak se pohybuje směrem dozadu (odo_17.json). Ve chvíli, kdy ujede vzdálenost, která překročí limit pro couvání v módu PT, (odo_18.json) přejde do módu *Trip* a spustí se nouzové brzdy. Po úplném zastavení (odo_19.json) se odešle zpráva 209: Mode change (*mode = EVCMODE::TRIP*).



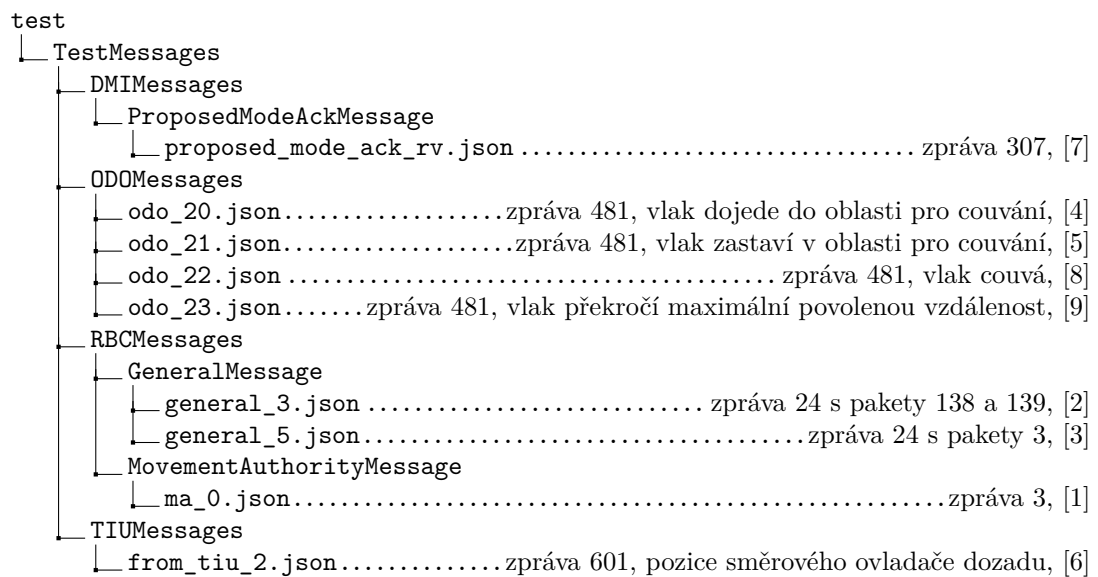
■ **Obrázek 6.19** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.14

6.1.15 Maximální vzdálenost v módu RV

Vlak přejde do módu FS zprávou 3: Movement Authority. EVC obdrží národní hodnoty v paketu 3: National Values ve zprávě 24: General message. EVC obdrží zprávu 24: General message s pakety 138: Reversing area information a 139: Reversing supervision information, dojedě do oblasti pro couvání (odo_20.json) a tam zastaví (odo_21.json). EVC odešle DMI zprávu 211: Reversing possible.

Strojvedoucí nastaví směrový ovladač do pozice pro couvání. Tato informace přijde EVC ve zprávě 602: From TIU. EVC reaguje navržením módu *Reversing* strojvedoucímu: zpráva 204: Mode proposed. Strojvedoucí potvrdí mód zprávou 307: Proposed mode acknowledged a systém přejde do módu *Reversing*.

Vlak couvá (odo_22.json) a při překročení limitu pro couvání (odo_23.json) přejde do módu *Trip* a spustí se nouzové brzdy.

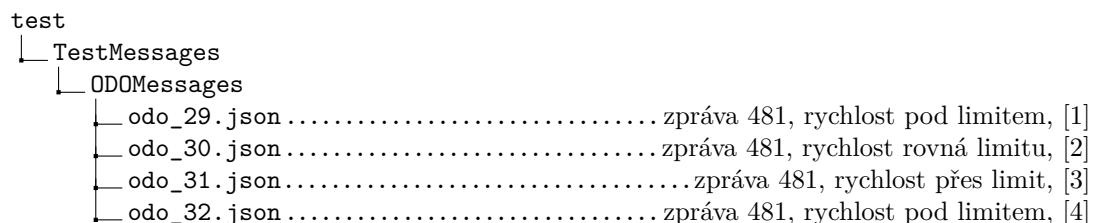


■ Obrázek 6.20 Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.15

6.1.16 Kontrola maximální rychlosti v módu RV

Nejprve se provede přechod do módu RV 6.1.5.

Maximální rychlost v módu RV je nastavená na 20 km/h. Pokud vlak tuto rychlost překročí, spustí se provozní brzdy. Ve chvíli, kdy rychlost klesne pod maximální povolenou, se brzdy uvolní.

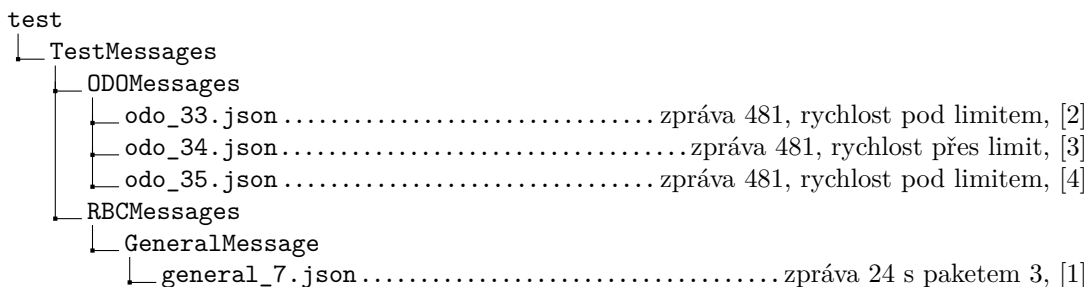


■ Obrázek 6.21 Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.16

6.1.17 Kontrola maximální rychlosti v módu SH

Nejprve se provede přechod do módu SH 6.1.8.1. Před potvrzením módu SH EVC přijme zprávu 24: General Message s paketem 3: National Values.

Maximální rychlost v módu SH je nastavená na 20 km/h. Pokud vlak tuto rychlost překročí, spustí se provozní brzdy. Ve chvíli, kdy rychlost klesne pod maximální povolenou, se brzdy uvolní.

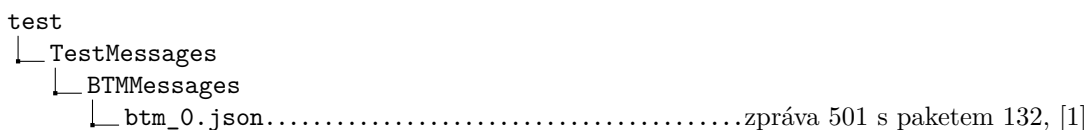


■ **Obrázek 6.22** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.17

6.1.18 Přijetí paketu 132: Danger for Shunting information

Nejprve se provede přechod do módu SH 6.1.6.3.

Ve chvíli, kdy vlak dostane zprávu od BTM, která obsahuje paket 132, přejde systém do módu *Trip*.



■ **Obrázek 6.23** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.18

6.1.19 Exit from Shunting

Nejprve se provede přechod do módu SH 6.1.6.3.

Ve chvíli, kdy vlak dostane od DMI zprávu 308: Shunting selected, kde je hodnota proměnné *shunting* nastavena na *false*, přejde systém do módu *Stand By*.



■ **Obrázek 6.24** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.19

6.1.20 Seznam balíz pro mód SH

Nejprve se provede přechod do módu SH 6.1.6.3, v některých scénářích je potřeba nahradit zprávu `sh_authorized_0.json` za některou ze zpráv, která obsahuje paket 49: List of balises for SH Area.

6.1.20.1 Seznam balíz nebyl poslán

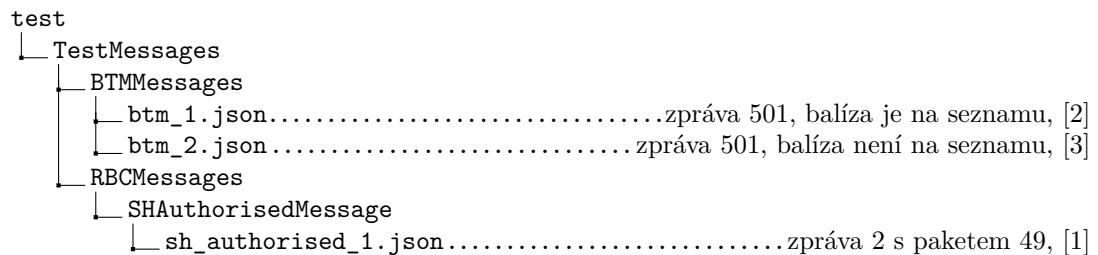
Zpráva `sh_authorized_0.json` se nenahrazuje. Vlak může přejet balízu s libovolným identifikátorem.



■ **Obrázek 6.25** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.20.1

6.1.20.2 Seznam balíz obsahuje nějaké balízy

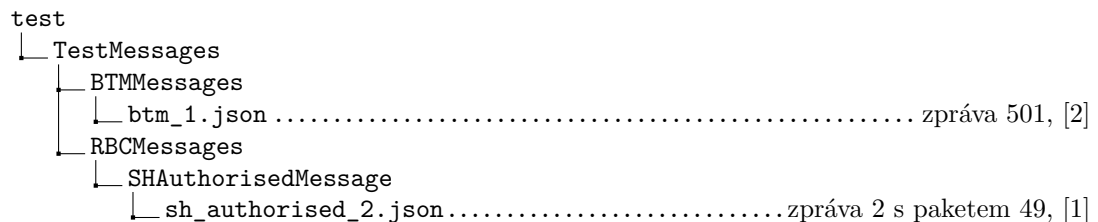
Zpráva `sh_authorized_0.json` se nahradí zprávou `sh_authorized_1.json`. Ve chvíli, kdy vlak dostane zprávu od BTM z balízy, která není na seznamu, přejde systém do módu *Trip*.



■ **Obrázek 6.26** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.20.2

6.1.20.3 Seznam balíz neobsahuje žádné balízy

Zpráva `sh_authorized_0.json` se nahradí zprávou `sh_authorized_2.json`. Ve chvíli, kdy vlak dostane zprávu od BTM z balízy, která není na seznamu (tudíž libovolné balízy), přejde systém do módu *Trip*.



■ **Obrázek 6.27** Adresářová struktura souborů obsahující zprávy pro testovací scénář 6.1.20.3

6.2 Průběh a výsledky testování

Testovací scénáře popsané v předchozí sekci byly prováděny manuálně za použití programu MQTTX [11], což je MQTT klient s grafickým rozhraním, který umožňuje zasílat zprávy a zobrazuje celý průběh komunikace. EVC byly zasílány připravené zprávy a bylo kontrolováno, že EVC odpovídá dle očekávání.

Veškeré testovací scénáře byly provedeny a bylo ověřeno, že EVC reaguje korektně. Posílané zprávy, přechody módů a změny vnitřního stavu odpovídají očekávaným výsledkům. Je tedy možné říct, že implementované řešení splňuje funkční požadavky tak, jak byly definovány v sekci 3.6.

Kód byl předán týmu ETCS, který na projektu pracuje v rámci předmětu *Softwarový týmový projekt 1* a má na starosti komponenty EVC, DMI a RBC. Projekt je v rozpracovaném stavu a do data odevzdání práce nebyly ostatní komponenty ve fázi, kdy by bylo možné otestovat funkčnost výsledků práce společně s ostatními komponentami.



Kapitola 7

Závěr










Cílem této práce bylo navrzení a implementace módů reversing, shunting a post trip v komponentě EVC trenážeru ETCS a následné otestování implementovaného řešení.

Teoretická část práce se zabývala vysvětlením toho, co je to systém ERTMS, popisem jeho součástí a přiblížením jeho využití pro vlakovou dopravu. Dále zde byly popsány módy, ve kterých se systém může nacházet, procedury, které se týkají módů reversing, shunting a post trip, a funkcionality systému, které se v těchto módech používají. Závěrem části byl shrnut současný stav projektu, kterého je tato práce součástí.

Praktická část se týkala návrhu procedur a zpráv a paketů, které se posílají mezi jednotlivými komponentami systému. Byly navrženy služby, které zpracovávají a ukládají data přijatá ve zprávách. Dále bylo popsáno, jak probíhala implementace navržených tříd. Praktická část práce byla uzavřena popisem průběhu testování, které sestávalo z provedení testovacích scénářů. Bylo ověřeno, že implementované řešení odpovídá požadavkům, které byly stanoveny.

V práci se podařilo navrhnout a implementovat zadané módy, přechody mezi jednotlivými módy a funkcionality v těchto módech. Řešení bylo úspěšně otestováno pomocí testovacích scénářů. Cíle práce se tedy povedlo splnit.

Význam symbolů v modelech

	Počáteční událost	Představuje začátek procedury
	Ukončující událost	Představuje konec procedury
	Událost	Událost, která nastane v průběhu procedury
	Odeslání zprávy ¹	Představuje událost odeslání zprávy
	Přijetí zprávy ¹	Představuje událost přijetí zprávy
	Splnění podmínky ¹	Představuje událost, kdy dojde k splnění nějaké podmínky
	Rozhodovací brána – podmínka	Procedura pokračuje větví, která splňuje podmínku
	Rozhodovací brána – událost	Procedura pokračuje větví, ve které nastane následující událost jako první
	Aktivita	Představuje činnost vykonanou nějakým aktérem v proceduře

■ **Tabulka A.1** Význam symbolů v modelech

¹Tato událost může být událostí, které nastane v průběhu procedury (dvojitě ohraničení), ale i počáteční (jednoduché ohraničení) a ukončující událostí (tučné ohraničení).

Změny v souborech

Příložený kód obsahuje všechny soubory v projektu. Pouze některé z těchto souborů byly vytvořeny nebo pozměněny v rámci této práce.

Tato příloha obsahuje seznam souborů, které byly pozměněny nebo vytvořeny v průběhu práce. Pokud je uveden adresář, pak všechny soubory v něm byly pozměněny nebo vytvořeny v průběhu práce. Název zakončený znakem * zastupuje dva soubory, které se liší pouze příponou (.h a .cpp).

```
src
├── Configurations
│   ├── EVCData
│   │   ├── EmergencyStopsConfiguration.h
│   │   ├── EVCGeneralConfiguration.h
│   │   ├── MACConfiguration.h
│   │   ├── ModeProfileConfiguration.h
│   │   ├── ReversingConfiguration.h
│   │   └── ShuntingConfiguration.h
│   └── MonitoringConfiguration.h
├── Core
│   ├── Application.cpp
│   ├── AsyncDeque.h
│   ├── AsyncProperty.h
│   └── HelperFunctions.h
├── Enums
│   ├── BrakeCommand.h
│   ├── MessageID.h
│   ├── PacketID.h
│   ├── ServiceType.h
│   └── TextID.h
└── Jobs
    └── ModeChangeTimeoutJob.*
```

```

src
├── MessageHandlers
│   ├── DMI
│   │   ├── DriverIdMessageHandler.cpp
│   │   ├── ExitSHMessageHandler.*
│   │   ├── ModeChangeAckMessageHandler.*
│   │   ├── ProposedModeAckMessageHandler.*
│   │   ├── RevokeBrakeCmdAckMessageHandler.*
│   │   ├── RVSelectedMessageHandler.*
│   │   ├── SHSelectedMessageHandler.*
│   │   └── StartSelectedMessageHandler.*
│   ├── RBC
│   │   ├── AckTerminationMessageHandler.cpp
│   │   ├── CondEmStopMessageHandler.*
│   │   ├── GeneralMessageHandler.*
│   │   ├── MovementAuthorityMessageHandler.*
│   │   ├── RecogExitTRMessageHandler.*
│   │   ├── RevocationEmStopMessageHandler.*
│   │   ├── SHAuthorisedMessageHandler.*
│   │   ├── SHRefusedMessageHandler.*
│   │   ├── SRAuthorisationMessageHandler.*
│   │   └── UncondEmStopMessageHandler.*
│   ├── BaseBtmMessageHandler.*
│   ├── FromTIUMessageHandler.*
│   ├── ODOMeasurementsHandler.*
│   └── TestMessageHandler.h
├── Models
├── Messages
│   ├── FromDMI
│   │   ├── ExitSHMessage.*
│   │   ├── ModeChangeAckMessage.*
│   │   ├── RevokeBrakeCmdAckMessage.*
│   │   ├── RVSelectedMessage.*
│   │   └── SHSelectedMessage.*
│   ├── FromRBC
│   │   ├── CondEmStopMessage.*
│   │   ├── RecogExitTRMessage.*
│   │   ├── RevocationEmStopMessage.*
│   │   ├── SHAuthorisedMessage.*
│   │   ├── SHRefusedMessage.*
│   │   └── UncondEmStopMessage.*
│   ├── ToDMI
│   │   ├── EmStopsRevokedMessage.*
│   │   ├── FixedMessage.*
│   │   ├── ModeChangeMessage.*
│   │   ├── RevokeBrakeCmdMessage.*
│   │   ├── RVPossibleMessage.*
│   │   ├── WaitingForEmStopsRevocationMessage.*
│   │   └── WaitingForSHResponseMessage.*
│   └── ToRBC
│       ├── EmStopAckMessage.*
│       └── RequestForSHMessage.*

```

```
impl
├── src
│   ├── Models
│   │   ├── Other
│   │   │   ├── EmergencyStop.*
│   │   │   ├── ModeArea.*
│   │   │   └── ReversingArea.*
│   │   └── Packets
│   │       ├── DangerForSHInfo.*
│   │       ├── ModeProfile.h
│   │       ├── RVAreaInfo.*
│   │       ├── RVSupervisionInfo.*
│   │       ├── SessionManagement.*
│   │       └── SHAreaBalises.*
│   └── Services
│       ├── EVCData
│       │   ├── EmergencyStops
│       │   ├── GradientProfile
│       │   ├── MA
│       │   ├── ModeProfile
│       │   ├── NationalValues
│       │   ├── ODO
│       │   │   └── ODODataService.cpp
│       │   ├── PositionReport
│       │   ├── Reversing
│       │   ├── Shunting
│       │   ├── SpeedProfile
│       │   ├── StaffResponsible
│       │   ├── TAF
│       │   └── Train
│       ├── Helper
│       ├── Message
│       │   └── MessageResolverService.cpp
│       └── Monitoring
└── test
    ├── Models
    │   └── Other
    └── TestMessages
```


Bibliografie

1. *ERTMS/ETCS* [online]. 2021. [cit. 2023-01-25]. Dostupné z: <https://www.azd.cz/cs/o-spolecnosti/ertmsetcs>.
2. *Implementační Plán ETCS* [online]. 2016. [cit. 2023-04-23]. Dostupné z: <https://www.ducr.cz/cs/legislativa/implementacni-plan-etcs>.
3. *ERTMS in brief* [online]. 2023. [cit. 2023-01-25]. Dostupné z: <https://www.ertms.net/about-ertms/ertms-in-brief/>.
4. *Radio Communication* [online]. 2018. [cit. 2023-01-25]. Dostupné z: https://www.era.europa.eu/domains/infrastructure/european-rail-traffic-management-system-ertms/radio-communication_en.
5. *Subsystems and constituents of the ERTMS* [online]. 2020. [cit. 2023-01-25]. Dostupné z: https://transport.ec.europa.eu/transport-modes/rail/ertms/how-does-it-work/subsystems-and-constituents-ertms_en.
6. STANLEY, Peter. *ETCS for engineers*. Hamburg: Eurail Press, 2011. ISBN 9783777104164.
7. EUROPEAN UNION AGENCY FOR RAILWAYS. *SUBSET-026 System Requirements Specification Chapter 4 Modes and Transitions* [online]. 2006. Ver. 2.3.0 [cit. 2023-01-25]. Dostupné z: <https://www.era.europa.eu/era-folder/set-specifications-1-etcs-b2-gsm-r-b1>.
8. EUROPEAN UNION AGENCY FOR RAILWAYS. *SUBSET-026 System Requirements Specification Chapter 5 Procedures* [online]. 2006. Ver. 2.3.0 [cit. 2023-01-25]. Dostupné z: <https://www.era.europa.eu/era-folder/set-specifications-1-etcs-b2-gsm-r-b1>.
9. EUROPEAN UNION AGENCY FOR RAILWAYS. *SUBSET-026 System Requirements Specification Chapter 3 Principles* [online]. 2006. Ver. 2.3.0 [cit. 2023-01-25]. Dostupné z: <https://www.era.europa.eu/era-folder/set-specifications-1-etcs-b2-gsm-r-b1>.
10. LOHMANN, Niels. *JSON for Modern C++*. 2022. Ver. 3.10.5. Dostupné také z: <https://github.com/nlohmann/json>.
11. EMQ TECHNOLOGIES INC. *MQTTX: Cross-platform MQTT 5.0 Desktop Client*. 2023. 1.9.2-6. Dostupné také z: <https://mqttx.app/>.

Obsah přiloženého archivu

	readme.txt	stručný popis obsahu média
	exe	adresář se spustitelnou formou implementace
	src	
	impl	zdrojové kódy implementace
	thesis	zdrojová forma práce ve formátu L ^A T _E X
	text	text práce
	thesis.pdf	text práce ve formátu PDF