

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra řídicí techniky

Distribuovaný model elektrické motorové jednotky

Martin Kostohryz

Vedoucí: doc. Ing. Hušek Petr, Ph.D.
Obor: Systémy a řízení
Studijní program: Kybernetika a robotika
Květen 2018

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kostohryz** Jméno: **Martin** Osobní číslo: **457183**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra řídicí techniky**
Studijní program: **Kybernetika a robotika**
Studijní obor: **Systemy a řízení**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Distribuovaný model elektrické motorové jednotky

Název bakalářské práce anglicky:

Distributed model of electrical motor unit

Pokyny pro vypracování:

1. Popište zařízení nacházející se v elektrické motorové jednotce, jejich umístění a možnosti redundance.
2. S využitím Petriho sítí vytvořte diskrétní distribuovaný model elektrické motorové jednotky.
3. Analyzujte model z hlediska provozních a poruchových stavů a navrhněte jejich řešení ze strany řídicího systému vozidla.

Seznam doporučené literatury:

- [1] J. L. Peterson: Petri Net Theory and the Modeling of Systems, Prentice-Hall Inc., 1981
- [2] B. Hruz and M. Zhou: Modeling and Control of Discrete-event Dynamic Systems with Petri Nets and Other Tools, Springer, 2007
- [3] Dokumentace k elektrické motorové jednotce

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Petr Hušek, Ph.D., katedra řídicí techniky FEL

Jméno a pracoviště druhého(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **31.01.2018**

Termín odevzdání bakalářské práce: **25.05.2018**

Platnost zadání bakalářské práce: **30.09.2019**

doc. Ing. Petr Hušek, Ph.D.
podpis vedoucí(ho) práce

prof. Ing. Michael Šebek, DrSc.
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Rád bych poděkoval Ing. Jiřímu Pohlovi a firmě Siemens za poskytnutí mnoho cenných rad při vypracovávání této práce. Dále děkuji svému vedoucímu Bakalářské práce doc. Ing. Petru Huškovi Ph.D. za umožnění pracovat na této práci a za připomínky na zlepšení během ní.

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.

V Praze, 25. května 2018

Podpis:

Abstrakt

Tato práce se zabývá elektrickými motorovými jednotkami, což je typ železničních vozidel. Jsou popsány jednotlivé subsystémy vozidla a na základě tohoto popisu je vytvořen distribuovaný model. Závěr práce je věnován řešení poruchových stavů železničních vozidel. Model vozidla byl vytvořen v programu CPN tools, který umožňuje simulovat Petriho barevné sítě. Vytvořený model byl bezchybně odsimulován a nyní se dá využít pro lepší pochopení funkčnosti elektrických motorových vozidel.

Klíčová slova: Elektrická motorová jednotka, Petriho barevné sítě, CPN tools, Distribuovaný model

Vedoucí: doc. Ing. Hušek Petr, Ph.D.

Abstract

This thesis describes an electrical motor unit, which is the type of railway vehicle. Individually subsystems are described. Based on that knowledge distributed model is created. The thesis focuses on failure conditions of the model and their solution in the end. The model was created in program CPN tools, which allows a user to simulate colored Petri nets. Created model was successfully simulated and it is possible to use it for better understanding features of the electrical motor unit.

Keywords: Electrical motor unit, Coloured Petri nets, CPN tools, Distributed model

Title translation: Distributed model of electrical motor unit

Obsah

1 Úvod	1	4 Nástroje na modelování	13
2 Hierarchie vaku	3	4.1 Petriho síť	13
3 Subsystemy železničních vozidel	5	4.2 Petriho barevné sítě	14
3.1 Rozvod energie	5	4.3 CPN tools	14
3.1.1 Elektrická trakce	6	4.4 Popis bloků v CPNtools	15
3.1.2 Rozvod ve vlaku	7	4.4.1 Datové typy	15
3.2 Brzdy	7	4.4.2 Místa	15
3.3 Trakční pohon	8	4.4.3 Hrany	15
3.4 Informační systémy	9	4.4.4 Přechody	16
3.5 Osvětlení vozidel	9	5 Tvorba modelu	17
3.6 Ventilace	10	5.1 Řídicí systém	17
3.7 Dveřní systém	10	5.1.1 Orientace vlaku	20
3.7.1 Stavby	11	5.2 Sběrnice	21
3.8 Pomocné zařízení	11	5.3 Ovládání vlaku	22
3.8.1 Chlazení transformátoru	11	5.3.1 Ovládání řídicím systémem ..	22
3.8.2 Brzdový kompresor	11	5.3.2 Ovládání strojvedoucím	22
		5.4 Subsystemy	23
		5.4.1 Trakční pohon	23

5.4.2 Brzdy	24
5.4.3 Systém dveří	24
5.4.4 Informační systémy	26
5.4.5 Ventilace	27
6 Simulace	29
7 Poruchové stavy vlaku	33
7.1 Rozdělení poruch	33
7.2 Redundance	34
7.3 Model	34
7.3.1 Poruchy	34
7.3.2 Řešení poruch řídicím systémem	35
7.3.3 Redundance	36
8 Závěr	39
Literatura	41
A Obsah CD	43

Obrázky

2.1 Rozdělení Vlaku A - Vlak, B - Vozidlo, C - Vůz	3	5.8 Model brzd	25
2.2 WTB a MVB	4	5.9 Model vnitřních dveří	25
2.3 WTB a MVB v elektrických motorových jednotkách	4	5.10 Model venkovních dveří.....	26
3.1 Rozvod energie ve vozidle	7	5.11 Model informačního systému ..	26
3.2 Schéma zapojení asynchronního motoru	9	5.12 Model ventilace - první část ...	27
4.1 CPN tools	14	5.13 Model ventilace - druhá část ..	27
4.2 Popis bloků CPNtools	16	6.1 Menu simulace	30
5.1 Řídicí systém - hlavní smyčka ..	18	6.2 Rozšíření control panelu o zpoždění	31
5.2 Řídicí systém - Rozšíření hlavní smyčky o více stavů	18	6.3 Záznam první 25 kroků simulace	32
5.3 Řídicí systém - Přidání seznamu stanic	19	7.1 Ukázka generování chyb u informačního systému	35
5.4 Řídicí systém - Určení směru jízdy	20	7.2 Řešení poruch řídicím systémem	36
5.5 Propojení WTB a MVB	21	7.3 Redundance trakčních motorů ..	37
5.6 Hotový řídicí systém	22		
5.7 Model trakčního pohonu	24		

Tabulky

3.1 Provozní stavy brzd 1-brzdí	
0-nebrzdí	8



Kapitola 1

Úvod

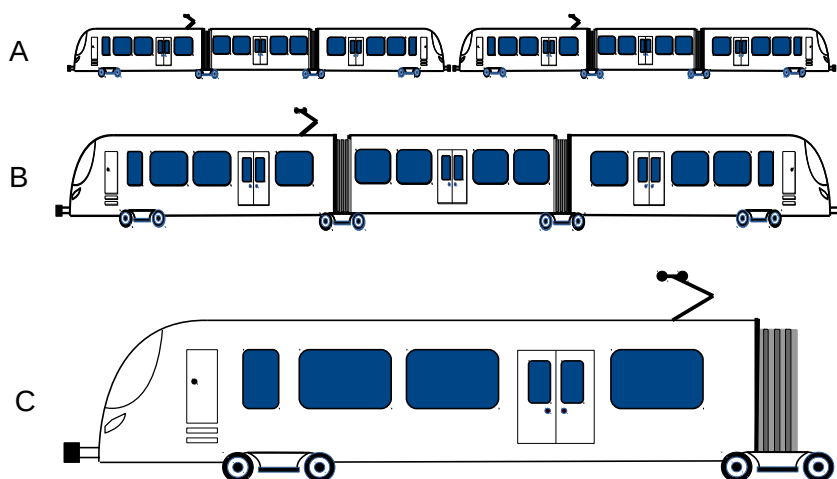
Kolejová vozidla se dělí na trakční (mají pohon) a netrakční (nemají pohon). Dále se dají rozdělit na užitková (přepravují osoby nebo věci) a neužitková (nepřepravují osoby nebo věci). Z hlediska této systematiky je elektrická trakční jednotka (EMU) kolejové vozidlo, složené z více vozů a zajišťuje jak trakční pohon tak i přepravu osob nebo věcí. Tento koncept představuje mnoho výhod a tak je v dnešní době stále více využíván. Spojením více vozů dohromady do ucelené jednotky se naskytne možnost rozložit trakční a pomocná zařízení do různých částí vozidla a docílit tak rovnoměrné rozdělení hmotnosti a lepších trakčních vlastností. Na příměstských tratích je tento typ vozidel žádaný díky možnosti přizpůsobení k přepravní poptávce jejich přepravní kapacitou a velkých rozjezdových a brzdných zrychlení. Na vysokorychlostních tratích se uplatňují elektrické motorové jednotky pro jejich dobrou aerodynamiku a rovnoměrné rozložení hmotnosti.

V této práci se zabývám funkcí elektrické trakční jednotky pro regionální dopravu. V první části je rozebráno jaké systémy musí v rámci vlaku fungovat, aby byla zajištěna celková funkčnost. Druhá část je zaměřená na to, jak jednotlivé subsystémy modelovat a spojit do jednoho systému. Poslední část je zaměřena na poruchové stavy a jejich řešení.

Kapitola 2

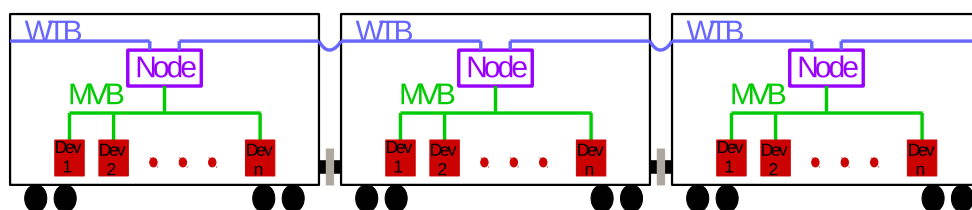
Hierarchie vaku

Vlakem se rozumí skupina vozidel, která plní přepravní funkci. Vlak může být složen z jednoho či více vozidel. Vozidla se dále dělí na jednotlivé vozy. Toto dělení je znázorněno na obrázku 2.1. Takovéto dělení je pro nás podstatné zejména kvůli komunikaci. Při automatizaci železničních vozidel začalo docházet k potřebě standartizovat komunikační síť mezi jednotlivými subsystemy a částmi vlaku. V roce 1999 byl tento problém vyřešen přijmutím TCN (Train communication network) jako standart IEC 61375 [Com99].



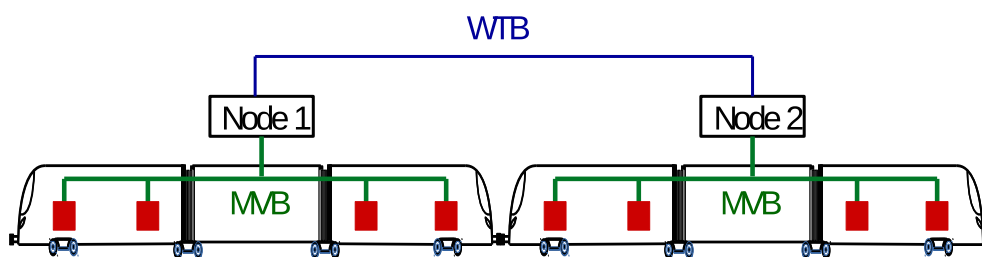
Obrázek 2.1: Rozdělení Vlaku
A - Vlak, B - Vozidlo, C - Vůz

TCN je složena z dvou sběrnic. První je WTB (Wire Train Bus). WTB je sběrnice, která zajišťuje komunikaci mezi vozy a případně i vozidly, které tvoří dohromady vlak. V každém voze se nachází komunikační uzel. K těmto uzlům je připojena druhá sběrnice a to MVB (Multifunction Vehicle Bus). MVB má na starosti komunikaci mezi subsystémy v rámci vozidla. Informace může také posílat pomocí komunikačního uzlu a WTB do ostatních vozidel a řídicí jednotce [HK01].



Obrázek 2.2: WTB a MVB

V případě ucelené elektrické motorové jednotky může být sběrnice MVB natažena přes celou délku vozidla. Sběrnice WTB pak slouží pouze při komunikaci s případnými dalšími připojenými vozidly, které společně tvoří vlak.



Obrázek 2.3: WTB a MVB v elektrických motorových jednotkách

Kapitola 3

Subsystémy železničních vozidel

Železniční vozidla jsou velmi komplikované stroje, které se skládají z mnoha dílčích zařízení. Popisovat tento systém jako celek by bylo příliš komplikované a neefektivní. Proto si systém rozdělíme na subsystémy. Každý subsystém bude řešit svůj specifický úkol. Subsystémy budou navzájem spolu komunikovat pomocí sběrnic, které jsme si popsali v předchozí kapitole.

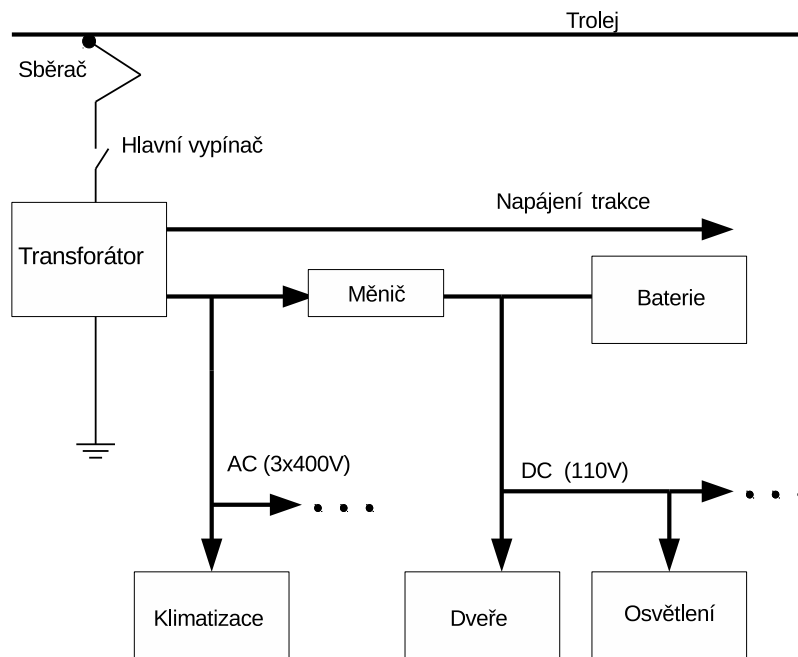
Rozdělením subsystémů do skupin se zabývá norma EN 15380-2 [ČS06]. Normu využijeme a nadefinujeme si subsystémy podle ní v dalších podkapitolách.

3.1 Rozvod energie

Prvním subsystémem, který si popíšeme, je rozvod energie. Jde o nejdůležitější subsystém jelikož na něm závisí fungování všech ostatních subsystémů. Tento subsystém můžeme rozdělit na dvě části: Elektrická trakce a Rozvod elektřiny ve vlaku [Pin12]

3.1.2 Rozvod ve vlaku

Nyní následuje popis distribuce elektrické energie v rámci vozidla. Uvažujme, že vozidlo čerpá elektrický proud z vrchního vedení. K troleji je vozidlo připojeno pomocí sběrače. Ten musí být dostatečně lehký, aby měl dobré dynamické vlastnosti a dobře přiléhal k troleji při jakékoliv rychlosti. Za sběračem následují ochranná zařízení mezi které patří například bleskojistky a hlavní vypínač. Dalším zařízením je transformátor, který snižuje napětí na potřebnou hodnotu. Transformátor má několik sekundárních vinutí a tak odděluje od sebe různé napájecí větve. Samostatná vinutí jsou určena pro napájení trakčních motorů a celé jejich napájení je podrobněji popsáno v samostatné kapitole. Další vinutí jsou určena pro napájení pomocné palubní sítě. Některé subsystemy jsou napájeny přímo střídavým napětím a pro stejnosměrné je za transformátorem umístěn měnič. Celé zapojení je vidět na obrázku 3.1.



Obrázek 3.1: Rozvod energie ve vozidle

3.2 Brzdy

Vozidla jsou vybaveny mnoha typy brzd. Prvním typem brzd jsou samočinné tlakové brzdy. Tyto brzdy využívají stlačený vzduch k vytvoření přítlačné síly

potřebné k vyvolání třecí síly na brzdné kotouče na všech vozidlech soupravy. Druhým způsobem brždění jsou kolejnicové elektromagnetické brzdy. Tyto brzdy jsou vybaveny elektromagnetem, který po spuštění přitáhne brzdný trámec ke kolejnici. Nejúčinnějším způsobem je třetí způsob elektromagnetického brždění, při kterém se jako brzdy využívá trakčních motorů. Trakční motory generují elektrický proud, který lze využít na napájení jiných částí vlaku nebo vrátit do sítě k tzv. rekuperaci.

Vzhledem k různým vlastnostem jednotlivých brzd se každá brzda využívá v jiné situaci nebo dochází k použití kombinace více brzd [JP12]. Tyto kombinace při různých situacích jsou přehledně zobrazeny v tabulce 3.1.

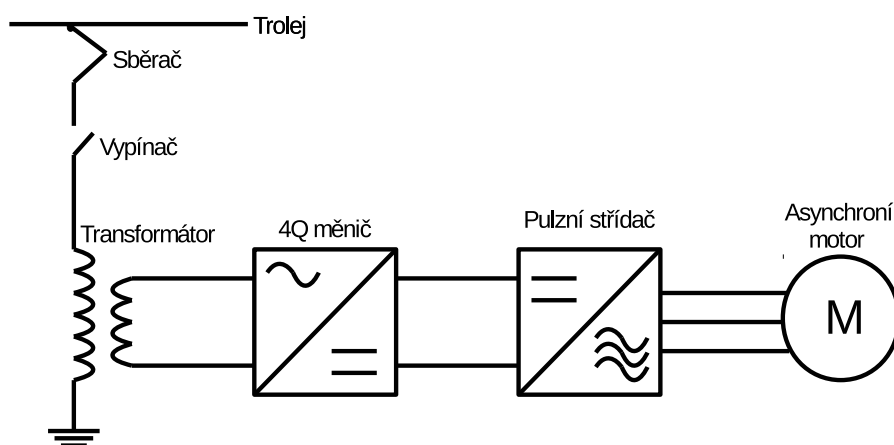
Typ brždění:	Elektrodynamické	elektromagnetické	Třecí
Provozní	1	0	0
Nouzové	1	1	0
Záchrané	0	1	1
Zajišťovací	0	0	1

Tabulka 3.1: Provozní stavy brzd
1-brzdí 0-nebrzdí

3.3 Trakční pohon

Elektrické trakční jednotky jsou vybaveny asynchronními motory s frekvenčním řízením otáček. Přenos energie statorem a rotorem probíhá pomocí elektromagnetické indukce. Díky této vlastnosti nedochází u těchto motorů k takovému opotřebování jako například u stejnosměrných motorů, které používají komutátor. Jsou vhodné pro svojí robustnost a spolehlivost, které je v železničních vozidlech zapotřebí.

Asynchronní motory není v praxi možné napájet pevnou frekvencí jen z jedné fáze rovnou z trakčního vedení. Tento problém se řeší napájení asynchronních tří-fázových trakčních motorů proměnnou frekvencí a proměnným třífázovým napětím zapojením 4Q měniče a pulzního střídače mezi hlavní transformátor a motor. Schéma tohoto zapojení je vidět na obrázku 3.2.



Obrázek 3.2: Schéma zapojení asynchronního motoru

3.4 Informační systémy

Informační systémy se nachází ve všech vozech a mají za úkol informovat cestující o aktuální poloze vlaku, následující stanici, rychlosti, času atd. Informační systémy můžeme rozdělit podle umístění na vnitřní a vnější a podle způsobu sdělení informace na optické a akustické.

3.5 Osvětlení vozidel

Další subsystém má za úkol osvětlení interiéru vozidel. Jednotlivé vozy jsou vybaveny různými typy světel tak, aby byl zajištěn dostatečný komfort pro cestující i bezpečnostní předpisy. Velká část všech světel je ovládaná centrálně buď automaticky nebo průvodčím. V městské a příměstských dopravě jsou toto obvykle jediná svítidla umístěná ve vozidle. V dálkových vlacích navíc přibývá individuální osvětlení ke každému sedadlu.

■ 3.7.1 Stavy

Dveře se mohou nacházet ve 4 stavech:

- **Otevřené**
- **Zavřené**
 - **Povolené otevřít**
Dveře je možné otevřít stiskem tlačítka
 - **Zakázané otevřít**
Dveře není možné otevřít
 - **Blokované**
Dveře jsou porouchány a není možné je otevřít

■ 3.8 Pomocné zařízení

Některé z výše uvedených subsystémů potřebují ke své funkci další pomocná zařízení. Tato zařízení se starají o bezpečný a efektivní provoz daných subsystémů.

■ 3.8.1 Chlazení transformátoru

Hlavním transformátorem prochází veliký výkon a vlivem ztrát se jeho část mění na tepelnou energii. Při větší teplotě dochází k zvýšení ztrát a navíc hrozí poškození. Proto je za potřebí transformátor chladit. K tomu se používá olej, do kterého je transformátor ponořen. Olej je uváděn do pohybu čerpadlem do výměníku, kde se ochladí okolním vzduchem za pomoci ventilátorů. Následně je ochlazený olej vrácen k transformátoru.

■ 3.8.2 Brzdový kompresor

V části, kde jsme se věnovali brzdám, jsme si řekli, že jeden z typu brzd, který se používá, je Samočinná tlaková brzda. Tato brzda potřebuje ke svému chodu

stlačený vzduch. Pro tento účel jsou železniční vozidla vybavena brzdovým kompresorem a zásobním vzduchojemem. Ve voze navíc musí být potrubí, kterým se tento stlačený vzduch bude distribuovat.

Kompresor dodává stlačený vzduch do hlavního vzduchojemu odkud je pomocí potrubí distribuován i do dalších zařízení.

Kapitola 4

Nástroje na modelování

Poté co jsme si popsali jednotlivé subsystémy a zařízení, které se nacházejí v kolejových vozidlech, můžeme se posunout k vytváření modelu. V této kapitole si popíšeme jaké nástroje máme k dispozici.

4.1 Petriho síť

Petriho sítě jsou nástroj určený ke studiu systémů. Teorie Petriho sítí nám umožňuje modelovat Petriho síť pomocí tohoto nástroje a vytvořit matematický popis systému. Analýza Petriho sítě nám poté umožňuje odhalit důležité informace o struktuře a dynamickém chování modelovaného systému. Tato informace může být použita k vyhodnocení modelovaného systému a k návrhnutí zlepšení nebo změně. Díky tomu je vývoj teorie Petriho sítí založen na aplikaci Petriho sítí v modelování a vytváření systémů [Pet81].

Princip Petriho sítí vymyslel v roce 1962 Carle Adame Petri ve své disertační práci. Síť je složena z míst, hran a přechodů. Místa a přechody jsou propojeny orientovanými hranami. Nikdy nemohou být spojeny dvě místa nebo přechody hranou. V každém místě se mohou nacházet tokeny, které představují společně s místem, ve kterém se nachází, určitou informaci o stavu systému. Naopak v přechodech se tokeny nemohou vyskytovat. Přechody reprezentují nějakou událost, ke které může dojít v systému. Pokud všechna místa z kterých vede hrana do daného přechodu mají token, jsou splněny podmínky pro odpálení přechodu. V tuto chvíli se ze všech vstupních míst odebere token a do všech

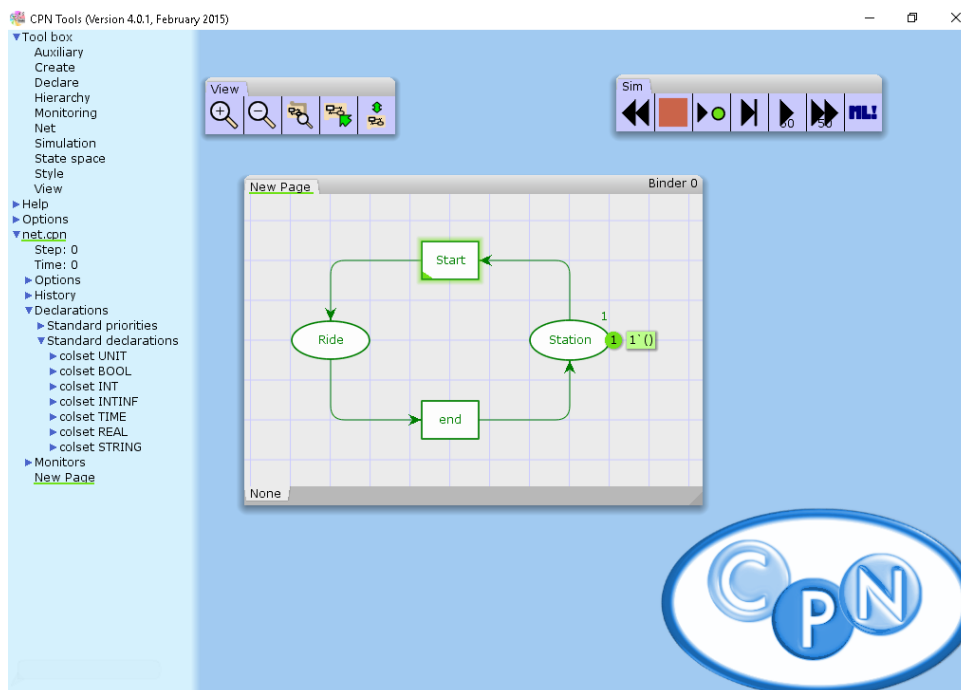
výstupních míst se jeden token dá.

4.2 Petriho barevné sítě

Rozšířením Petriho sítí jsou Petriho barevné sítě. Tyto sítě se liší v tom, jakou informaci nesou v sobě tokeny. Zatím co v Petriho sítích jsou všechny tokeny stejné, v Petriho barevných sítích má každý token svůj datový typ a hodnotu. Díky tomu jsme schopni namodelovat mnohem složitější systémy [citb].

4.3 CPN tools

CPN tools je program, který je určen pro návrh Petriho barevných sítí [citc]. Program byl vytvořen na Aarhus Universitě v Německu. Program nabízí grafický editor pro návrh Petriho sítí a podporuje i simulaci. To jsou důvody proč pomocí tohoto programu bude vytvořen model celého vozidla.



Obrázek 4.1: CPN tools

■ 4.4 Popis bloků v CPNtools

■ 4.4.1 Datové typy

CPNtools obsahuje několik základních datových typů jako Unit, Boolean, Integer, Large Integer, Real, Time, String, Enumerated a Index. Program nám však umožňuje kombinací těchto typů si vytvořit svůj vlastní. Další možnost jak vytvořit datový typ je Výčtový typ. U toho jednoduše vypíšeme všechny hodnoty, kterých může nabývat.

Dále můžeme tvořit proměnné a konstanty, které nám umožní tvořit komplikovanější funkce a podmínky. Těchto možností budeme využívat při tvorbě modelu.

■ 4.4.2 Místa

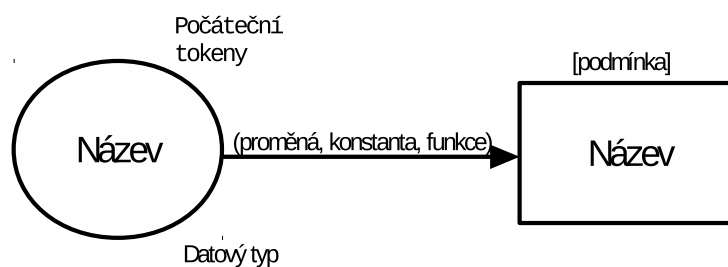
Místa jsou v programu reprezentovaná elipsou. Mohou mít libovolný počet vstupů a výstupů. Uprostřed je napsané jméno. Dále mají místa svůj datový typ a seznam počátečních tokenů. V místě se nemůže nacházet jiný datový typ než je jeho datový typ.

■ 4.4.3 Hrany

Hrany jsou spoje mezi přechody a místy. Můžou být jednosměrné nebo obousměrné. Obsahují navíc popis jaké tokeny mohou přenášet. Tokeny musí vždy být stejného typu jako je místo do kterého přechází nebo z kterého vychází. Mohou přenášet jeden nebo více tokenů. Popis může obsahovat proměnné konstanty nebo funkce. Pokud je hrana popsána proměnnou, mohou po ní projít všechny tokeny splňující daný datový typ. V případě konstant musí přesně souhlasit data v konstantě s daty v tokenu. Funkce pak mohou obsahovat proměnné i konstanty a definovat složitější podmínky.

4.4.4 Přechody

Přechody jsou zobrazeny jako obdélníky. Taktéž pro ně platí, že mají libovolný počet vstupů a výstupů. Vstupní i výstupní data mohou mít libovolný datový typ. Přechody navíc můžou obsahovat podmínky za jakých jsou proveditelné. Základní podmínka je, že všechna místa ze kterých vedou hrany do přechodu, mají v sobě token, který může projít do přechodu přes danou hranu.



Obrázek 4.2: Popis bloků CPNtools

Kapitola 5

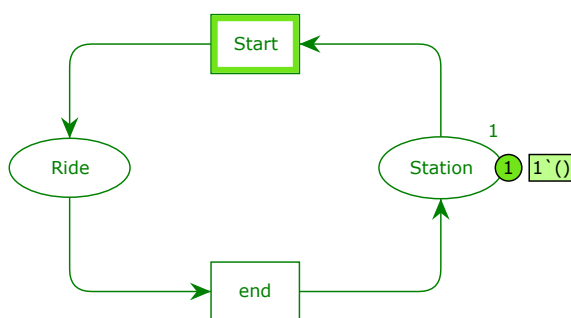
Tvorba modelu

V třetí kapitole jsme si popsali subsystémy a zařízení nacházející se v železničních vozidlech. Ve čtvrté kapitole pak jaké máme nástroje na modelování. Nyní tyto vědomosti spojíme a začneme tvořit diskrétní model železničního vozidla. Model bude rozdělen na menší části tak, aby jsme dosáhli větší přehlednosti a lepšího grafického znázornění. Tyto části budou reprezentovat jednotlivé subsystémy. Všechny subsystémy následně budou spojené v jeden systém.

5.1 Řídicí systém

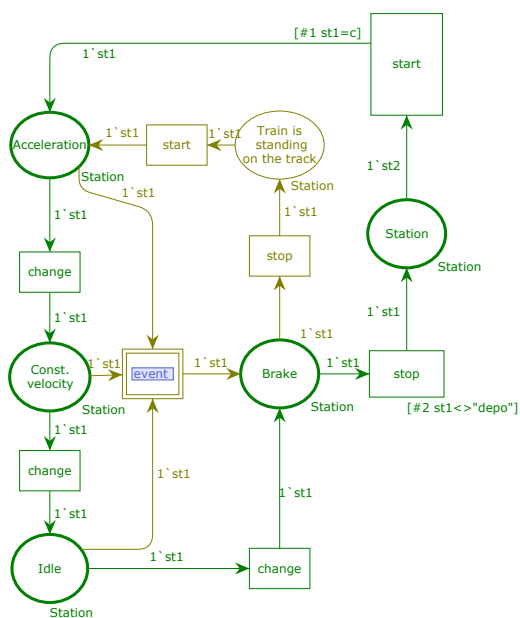
Modelováním začneme s řídicím systémem. Řídicí systém si bude udržovat informaci o aktuálním stavu vlaku, stanovišti strojvedoucího a bude posílat příkazy subsystémům.

První část, kterou se tedy budeme zabývat, je hlavní smyčka. Hlavní smyčka bude popisovat stav vlaku. Stav budou z počátku dva a to *Station* a *Ride*. Tato situace bude znázorněna tak, že budem mít dvě místa, která odpovídají stavům a dva přechody, kterými jsou spojené. Vlak bude reprezentován jedním tokenem. Místo ve kterém se token nachází nám pak říká v jakém stavu je vlak. Celá situace je znázorněna na obrázku 5.1.



Obrázek 5.1: Řídicí systém - hlavní smyčka

Nyní se zaměříme na samotnou jízdu. Budeme rozlišovat 4 fáze jízdy: *Acceleration*, *Const velocity*, *Idle* a *Brake* a nahradíme jím stav *Ride*. Tyto fáze pujdou v pořadí jak byly napsány za sebou a tak je propojíme přechody. Takto by probíhala jízda ideálně. Pokud ale během jízdy dojde k události kvůli které bude muset vlak zastavit nebo přibrzdit, budou se fáze opakovat znovu od začátku. Tyto události budeme simulovat tak, že z každého místa které představuje jednu z fází povede přechod do místa brždění. Dále si vytvoříme místo *Train is on the track* do kterého vlak přejde až dobrzdí. Z tohoto místa pak povede přechod opět do první fáze. Célé schéma je na obrázku 5.2.

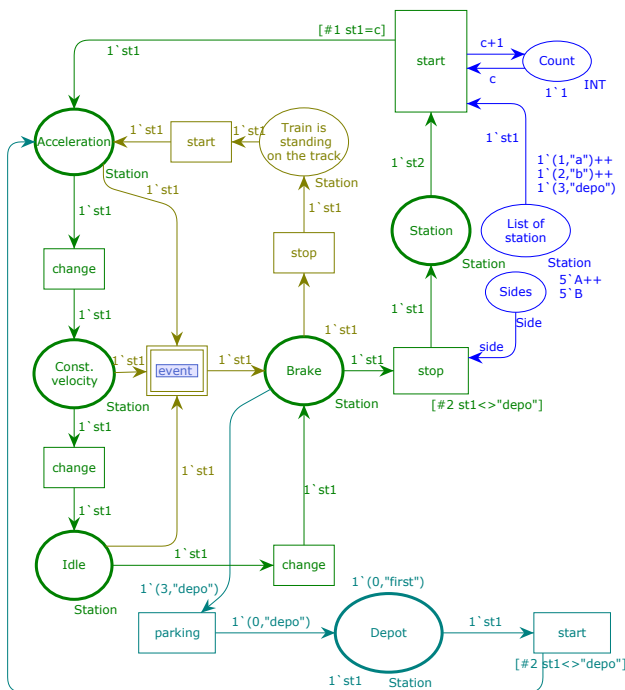


Obrázek 5.2: Řídicí systém - Rozšíření hlavní smyčky o více stavů

Model řídicího systému si rozšíříme o informaci, ve které nebo do které zastávky vlak jede. Toho docílíme tak, že si navrhne vlastní datový typ. Tento typ nazveme `Station`. Bude se skládat z typu `int` a `string`. Hodnota `Integeru` nám říká v kolikáté stanici jsme a hodnota `String` název této stanice. Dále vytvoříme proměnou, kterou použijeme na hrany mezi místy a přechody v hlavní smyčce:

```
Station product: int * string;
var st1, st2: Station;
```

Nyní umístíme nové místo `List of Stations` a uložíme do něj seznam všech zastávek a připojíme ho k přechodu `Start`. Pokaždé když se vlak bude rozjíždět ze stanice, vezme token s následující stanicí a token znázorňující vlak a předá token do místa `jízda` s novou zastávkou. Aby jsme zajistili, že stanice budou odebírány ve správném pořadí, přidáme ještě místo `count`, ve kterém budeme počítat kolikátá stanice je na řadě. Toto místo spojíme s přechodem `Start`. Dále vyřešíme situaci, kdy vlak už projel všechny stanice a jede do depa. Přidáme přechod `Parking`, do kterého povede hrana z místa `Brake`. Poté si vytvoříme místo `Depot` a přechod `start` a vše propojíme.

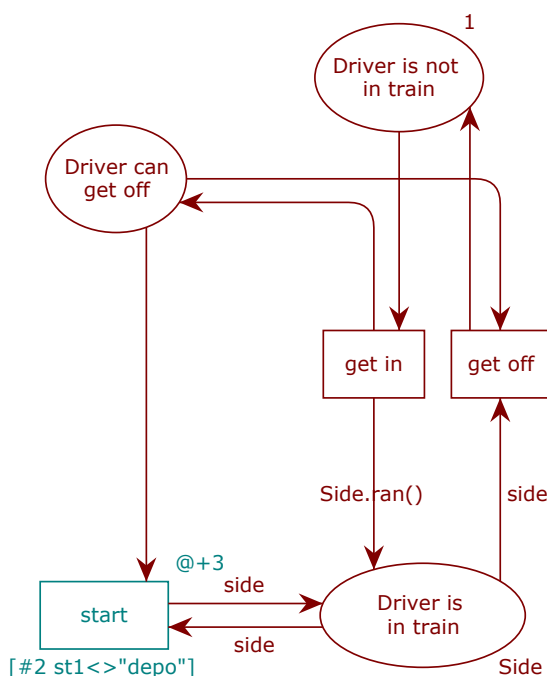


Obrázek 5.3: Řídicí systém - Přidání seznamu stanic

5.1.1 Orientace vlaku

Většina dnešních elektrických trakčních jednotek je vybavena více řídicími stanovišti. Výhoda je v tom, že si strojvedoucí pouze přejde na opačný konec. Poté co nastoupí strojvedoucí do vlaku, je zapotřebí určit, z kterého stanoviště bude vlak řízen. Zejména pokud je vlak sestaven z více vozidel, která mohou být různě orientovaná je nutno jednotně určit, která strana je levá a která pravá a který směr je vpřed a který vzad.

Tuto situaci vyřešíme pomocí zapojení níže. Nasedajícího strojvedoucího popisuje přechod *get in*. Tato akce pošle token do míst *Driver is in train* a *Driver can get off*. Místo *Driver is in train* bude typu *side* a token na tomto místě nám bude říkat jestli je obsazeno stanoviště A nebo B. Místo můžeme spojit obousměrně s přechodem *Start*, neboť vozidlo nemůže vyjet ze stanice bez strojvedoucího. Místo *Driver can get off* nám naopak při spojení s přechodem *get off* zajistí, že nemůže strojvedoucí vystoupit během jízdy.

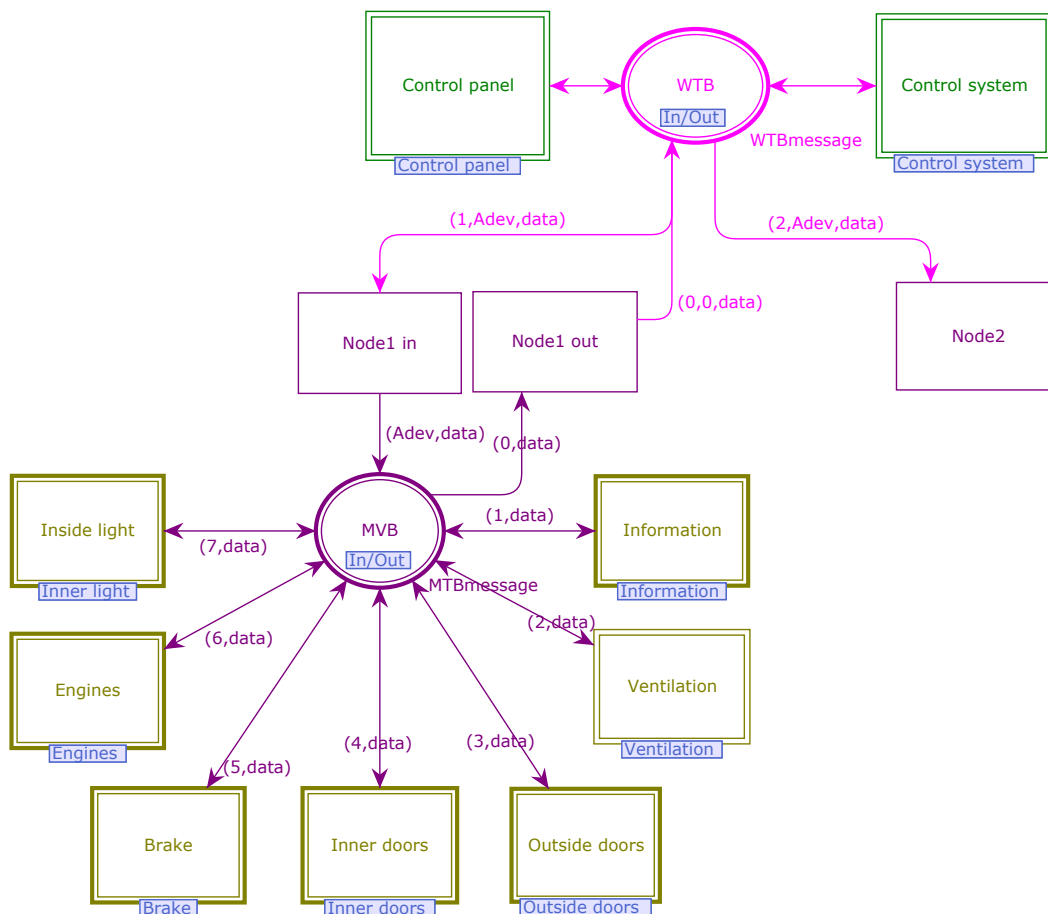


Obrázek 5.4: Řídicí systém - Určení směru jízdy

5.2 Sběrnice

Aby mohly systémy navzájem komunikovat, musíme namodelovat sběrnice. Sběrnice máme ve vlaku dvě (WTB a MVB). Obě budou reprezentovány svým vlastním místem a budou propojené přechodem *Node*. V případě že by jsme měli zapojeno více vozidel dohromady, přibyly by nám i další přechody *Node* připojené k místu WTB. Každý vůz by taktéž měl své vlastní místo MVB. Sběrnice MVB budou propojené hranami s konkrétními přechody v jednotlivých subsystémech.

Pro místa představující sběrnice vytvoříme datové typy, které budou představovat posílané zprávy. Nejdříve si vytvoříme datový typ *data*. Typ *data* bude složen z Integeru a Stringu. Půjde o *data*, která budou přijímána subsystémy. Abychom dokázali rozlišit pro jaký subsystém je zpráva určená, rozšíříme ji o Integer, který bude představovat adresu subsystému. Stejným způsobem rozšíříme *data* o adresu vozidla, pro které je zpráva určená.

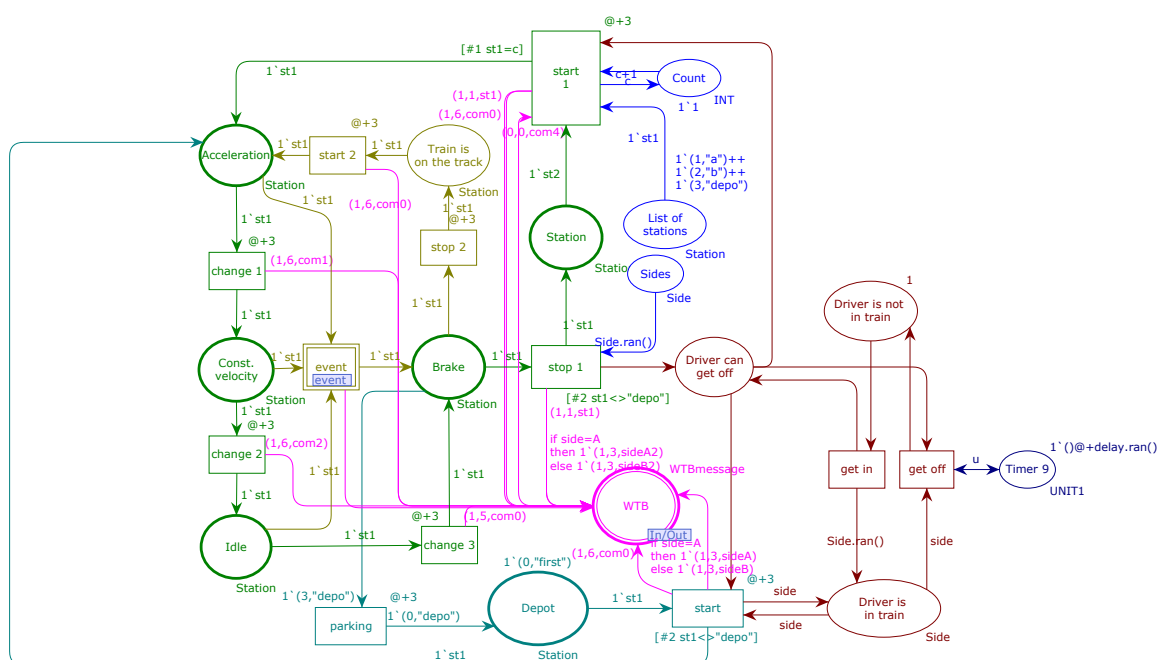


Obrázek 5.5: Propojení WTB a MVB

5.3 Ovladání vlaku

5.3.1 Ovladání řídicím systémem

Vlak bude z velké části ovládán řídicím systémem. Toho docílíme tak, že propojíme hranami přechody, které mění stav vlaku s místem reprezentujícím WTB. Pokaždé když nyní dojde k nějaké události v řídicím systému, bude o tom vyslána zpráva. Tímto způsobem například budeme ovládat motory při změně stavu vozidla z *Acceleration* na *Const. velocity*.



Obrázek 5.6: Hotový řídicí systém

5.3.2 Ovladání strojvedoucím

Část kterou jsme si teď popsali je vykonávána řídicím systémem automaticky. Nyní nastává otázka jak v modelu reprezentovat zásah strojvedoucího do řízení. Situaci vyřešíme pomocí ovládacího panelu. Ovládací panel budeme modelovat pomocí několika přechodů, které reprezentují jednotlivé ovládací prvky. Přechody nebudou mít žádný vstup, pouze výstup (tudíž je bude možno provést kdykoliv). Výstup je vždy jeden a směřuje do místa *WTB*.

■ 5.4 Subsystémy

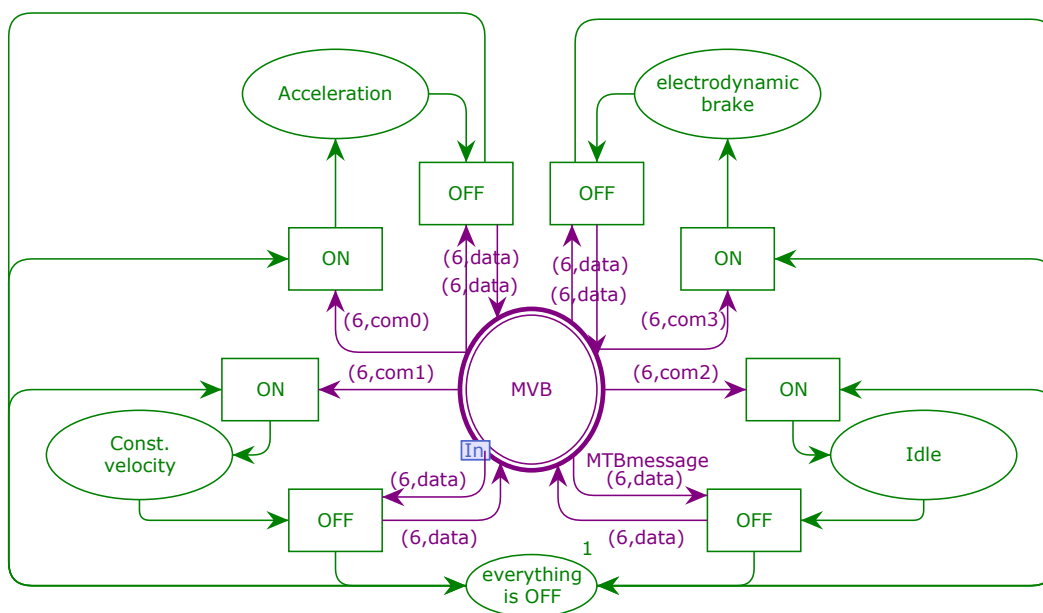
Nyní si do modelu přidáme dílčí subsystémy. Ty budou přijímat informace z MVB a dále je zpracovávat. Také budou moci navzájem komunikovat přes sběrnici.

Pro účely komunikace se subsystémy se nám bude hodit nadefinovat si některé standartní zprávy, které bude posílat řídicí systém. Pro tento účel je v CPNtools možnost definovat konstanty. Vytvoříme tedy 4 příkazy, které budou typu *data*.

```
val com0=(0,"");  
val com1=(1,"");  
val com2=(2,"");  
val com3=(3,"");
```

■ 5.4.1 Trakční pohon

Trakční pohon se bude moci nacházet ve 4 stavech *Acceleration*, *Const. velocity*, *Idle*, *Electrodynamic brake*. Tyto 4 stavy budou reprezentované místy. Navíc bude existovat ještě 5. stav, kdy jsou motory vypnuté a tento stav budeme reprezentovat tím, že v žádném místě není token. Motory se mezi stavy budou přepínat na základě příkazu od řídicího systému. Abychom při příjmu příkazu nastavili daný stav a zároveň odebrali token z předchozího stavu, potřebujeme další místo, které bude reprezentovat jestli jsou motory v jednom ze 4 stavů. Následně propojíme místa s přechody tak jak je naznačeno na obrázku 5.7, abychom dosáhli požadované funkčnosti.



Obrázek 5.7: Model trakčního pohonu

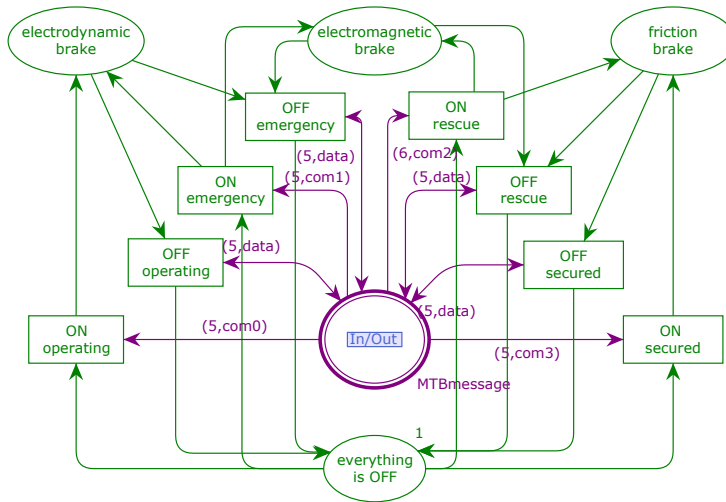
5.4.2 Brzdy

U brzd budeme předpokládat stejný model jako u motorů. Brzdy máme trojího typu. Budeme tedy mít tentokrát pouze 3 místa reprezentující stavy. Dalším rozdílem oproti modelu trakčních motorů je ten, že pokud budeme chtít využívat elektrodynamickou brzdu, musíme zaslat zprávu do subsystému motorů, který samotné brždění zajišťuje. K tomu bude sloužit přechod který bude propojen obousměrně s místem *electrodynamic brake* a výstupní hranou spojen s místem *MVB*. Zapojení je na obrázku 5.8.

5.4.3 Systém dveří

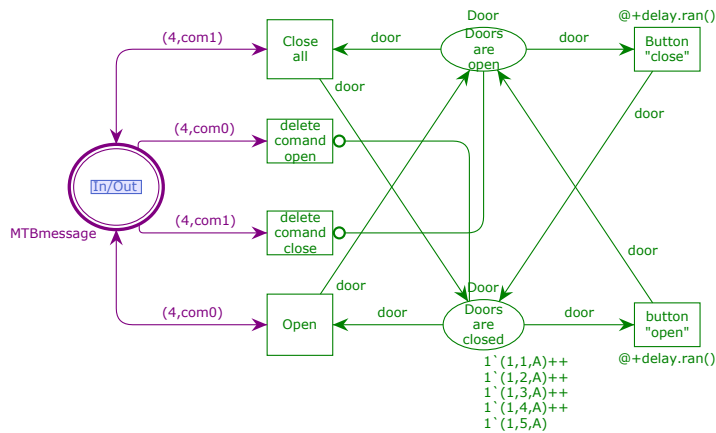
Dveře, jak již bylo popsáno v třetí kapitole, máme dvojího typu: vnitřní a vnější. Pro oba typy si vytvoříme zvlášť schéma, přestože se jejich funkce podobá.

Začneme vnitřními dveřmi, které jsou o něco jednodušší. Vnitřní dveře se mohou dostat do dvou stavů: otevřené nebo zavřené. Pro tyto dva stavy vytvoříme místa, která je budou reprezentovat. Stavy dveří se mohou změnit příkazem od cestujícího. Situace bude modelována propojením míst *open*



Obrázek 5.8: Model brzd

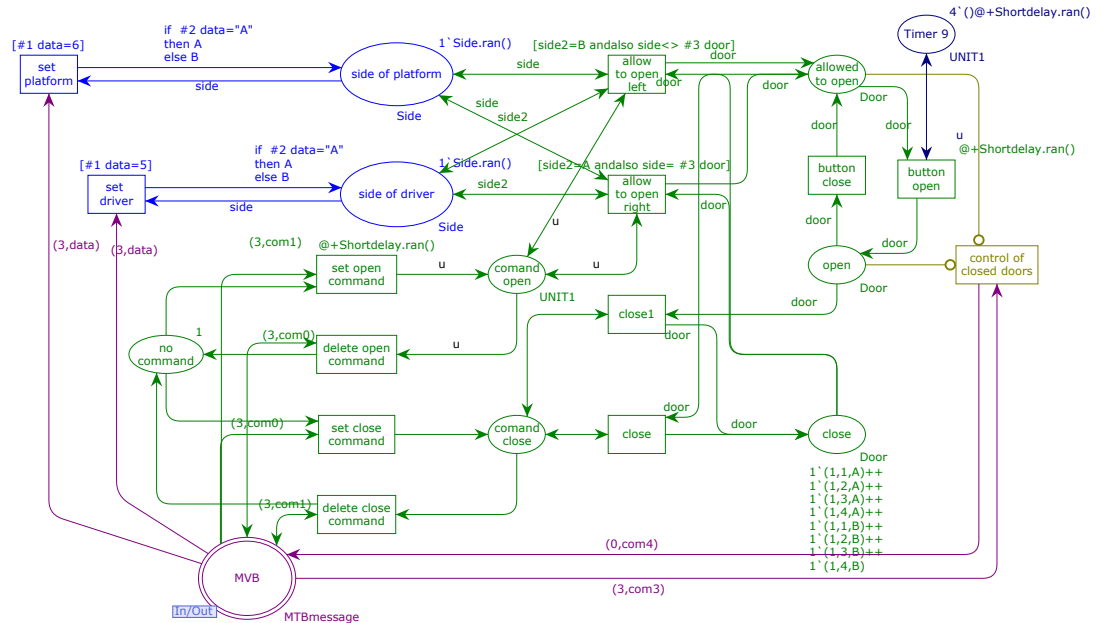
a *close* dvěma přechody, každým v jednom směru. Samotné dveře budou reprezentovány tokeny. K tomuto účelu si vytvoříme nový datový typ. Pojme-nujemeho *Door* a bude mít složky *int* - číslo vozu *int* - číslo dveří. Stejným způsobem budeme modelovat i osvětlení.



Obrázek 5.9: Model vnitřních dveří

Venkovní dveře ovládá strojvedoucí a řídicí systém. Řídicí systém povolí otevření dveří na straně nástupiště. Následně strojvedoucí může otevřít dveře. Dveře se však pouze odemknou a cestující si je otvírají sami. Tento stav bude reprezentován místem *Allow to open*, propojíme přechody stejně jako jsme to udělali u stavů *Open* a *Close*. Navíc ještě tento systém musí zajistit, aby se dveře otvíraly na správné straně. Proto model ještě vybavíme dvěma místy, která reprezentují pozici strojvedoucího a pozici nástupiště. Pozice strojvedoucího se nastaví hned po nástupu strojvedoucího do vozu a strana

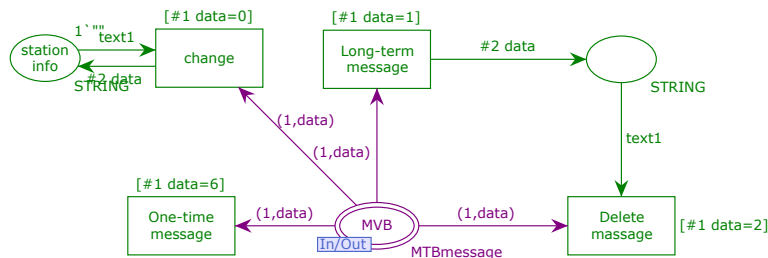
nástupiště je posílána od řídicího systému vždy, když vlak zastaví ve stanici. Těchto informací využijeme a povolíme otevření dveří pouze na správné straně.



Obrázek 5.10: Model venkovních dveří

5.4.4 Informační systémy

Informační systém bude přijímat zprávy od řídicího systému, i od ostatních subsystémů. Tyto zprávy se budou zobrazovat dokud nebude přijata další zpráva s příkazem o odstranění informace. Nebo se zobrazí pouze jednorázově.

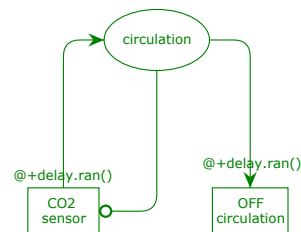


Obrázek 5.11: Model informačního systému

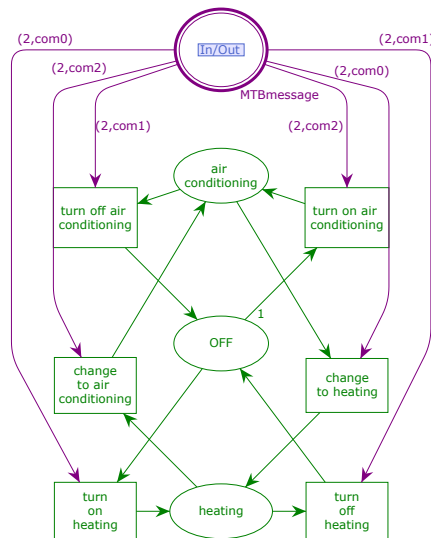
5.4.5 Ventilace

Model ventilace bude rozdělen na dvě části. Část reprezentující cirkulaci bude velmi jednoduchá. Budeme mít místo reprezentující nasávání vzduchu z venku a dva přechody, které buď odeberou nebo pošlou token do místa.

Druhá část je ovládání topení. To budeme reprezentovat třemi stavy: *Air conditioning*, *Heating* a *OFF*. Mezi těmito stavy se budeme přepínat pomocí příkazů přijímaných od strojvedoucího. Využijeme k tomu příkazů nadefinovaných na začátku této kapitoly. Zapojení jsou vidět na obrázcích 5.12 a 5.13.



Obrázek 5.12: Model ventilace - první část



Obrázek 5.13: Model ventilace - druhá část

Kapitola 6

Simulace

V předchozí kapitole jsme dokončili kompletní model vozidla a nyní je nařadě vyzkoušet simulaci. Simulací si ověříme funkčnost modelu a přijdeme na případné chyby. Program CPNtools nám nabízí několik možností jak simulovat model[cite]. K simulaci slouží položka *Simulation*. Přetáhneme jí na pracovní plochu a zobrazí se nám okno s nabídkou 6.2. Označíme si nástroj, který chceme využívat a přetáhneme ho na přechod nebo síť, kterou chceme odsimulovat.

- **Rewind**
Vrátí simulaci do počátečního stavu.
- **Stop**
Zastaví probíhající simulaci.
- **Bind manually**
Má mnoho využití podle toho kde je použita[cita]
- **Single step**
Provede přechod, na který je použita (pokud je proveditelný) nebo vybere náhodně z přechodů v dané části sítě.
- **Play**
Provede předem určený počet přechodů. Simulace je zobrazována tak, aby bylo vidět co se přesně děje.
- **Fast Forward**
Funguje stejně jako předchozí možnost, jen simulaci nezobrazuje průběžně, ale ukáže až výsledek.

V souboru je uloženo pořadí tak i čas ve kterém byly přechody vykonávány. Záznam prvních 25 kroků je vidět na obrázku 6.3. V záznamu si můžeme ověřit jestli se všechny kroky udávají ve správném pořadí.

```
CPN Tools simulation report for:  
/bak.cpn  
Report generated: Wed May 23 13:41:07 2018
```

```
step time transition  
1 0 get_in @ (1:Control_system)  
2 0 start @ (1:Control_system)  
3 0 Node1_in @ (1:Vehicle)  
4 0 Node1_in @ (1:Vehicle)  
5 0 set_driver @ (1:Outside_doors)  
6 0 ON_Acceleration @ (1:Engines)  
7 0 clear @ (1:Brake)  
8 1 light_on_1 @ (1:Inner_light)  
9 3 CO2_sensor @ (1:Ventilation)  
10 3 Light_On @ (1:Control_panel)  
11 3 change_1 @ (1:Control_system)  
12 3 light_on_1 @ (1:Inner_light)  
13 3 Node1_in @ (1:Vehicle)  
14 3 Light_off @ (1:Inner_light)  
15 3 Node1_in @ (1:Vehicle)  
16 3 OFF_Acceleration @ (1:Engines)  
17 3 ON_Const @ (1:Engines)  
18 4 button @ (1:Inner_doors)  
19 4 Button @ (1:Inner_doors)  
20 5 one @ (1:Control_panel)  
21 5 Node1_in @ (1:Vehicle)  
22 5 Heating_ON @ (1:Control_panel)  
23 5 Node1_in @ (1:Vehicle)  
24 5 One @ (1:Information)  
25 5 turn_on_heating @ (1:Ventilation)
```

Obrázek 6.3: Záznam první 25 kroků simulace

Kapitola 7

Poruchové stavy vlaku

Do této chvíle jsme uvažovali plně funkční vozidlo bez závad. To neodpovídá příliš reálnému provozu a proto se nyní zaměříme na to, co se stane, pokud se nějaká závada vyskytne. Dále se podíváme jak danou situaci můžeme řešit.

7.1 Rozdělení poruch

Nejprve si rozdělíme poruchy do skupin. Skupiny budeme mít 3 a budou dané závažností poruchy. Pokud je porucha pouze malého rozsahu a nemá vliv na provoz vozidla, můžeme jí v průběhu jízdy ignorovat a nahlásit až po ukončení provozu a příjezdu do depa. Obvykle jsou tyto poruchy malého rozsahu a je možné je opravit během rutinní údržby. Druhou možností jsou situace, kdy porucha má větší rozsah. Jedná se o poruchu subsystému, který má vliv na pohyb nebo bezpečnost vozidla. Jelikož tato porucha představuje riziko pro cestující, vlak pouze dokončí jízdu do nejbližší stanice, kde se bude porucha řešit. Poslední možností je situace, kdy vozidlo nemůže dále pokračovat v jízdě a ihned začne brzdit. Tato situace nastane pouze ve výjimečných případech.

7.2 Redundance

Z rozdělení poruch vyplývá, že pokud nebudeme chtít narušovat jízdní řád, můžeme si dovolit pouze poruchy v první skupině. Bohužel nemůžeme zajistit, aby se zařízení ve vlaku nikdy neporouchala. Můžeme ovšem do vozidla umístit více stejných zařízení. Pokud jedno přestane fungovat, jednoduše přepneme na druhé. Porouchané zařízení je poté opravené až vozidlo dokončí jízdu.

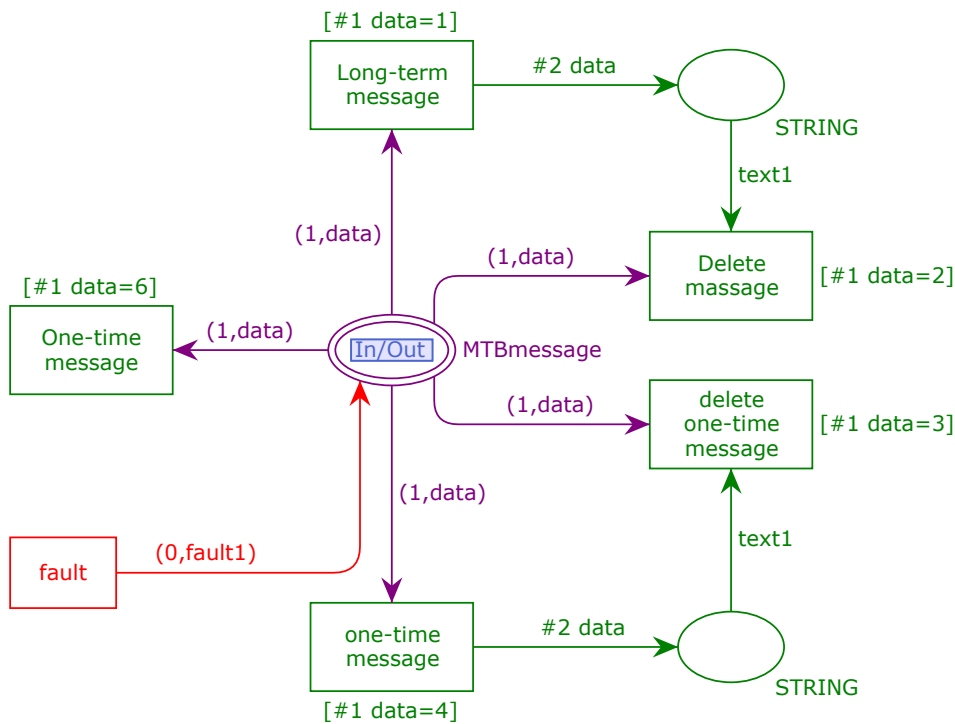
Výhodou tohoto přístupu je jak už bylo řečeno větší spolehlivost vozidla. Nevýhodou je celkový počet zařízení ve vozidle. Více zařízení znamená vyšší celkovou váhu a cenu vozidla. Taktéž musíme nadbytečná zařízení někde do vozidla umístit, což nám zmenšuje velikost přepravního prostoru. Tento princip je používán u zvláště důležitých subsystémů (Trakční pohon, napájení palubní sítě).

7.3 Model

7.3.1 Poruchy

Poruchy přidáme do modelu tak, že každý subsystém vybavíme zapojením, které je bude hlásit řídicímu systému. Řídicí systém tyto zprávy zpracuje a rozhodne jak na ně bude reagovat.

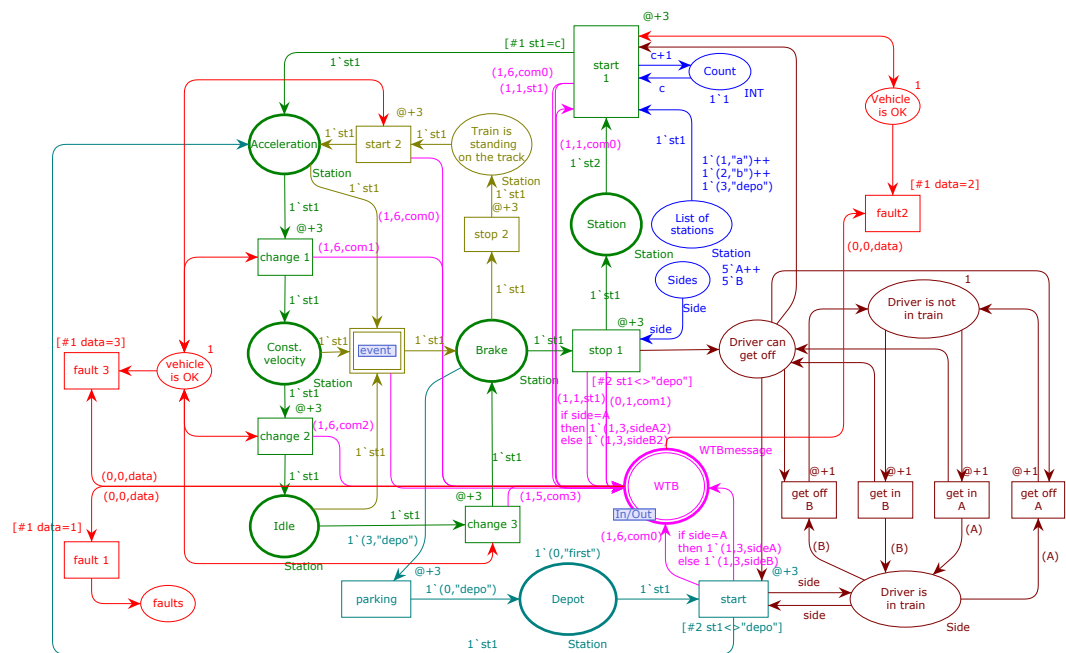
V každém subsystému přidáme přechod, který bude spojen se sběrnici a nebude mít žádný vstup. Pro zprávy, které tyto přechody budou vysílat, si vytvoříme konstanty. Konstanta budou typu data. Do Integeru napíšeme číslo reprezentující vážnost poruchy (1 - 3 viz Rozdělení poruch) a string bude obsahovat řetězec *fault-name of subsystem*.



Obrázek 7.1: Ukázka generování chyb u informačního systému

7.3.2 Řešení poruch řídicím systémem

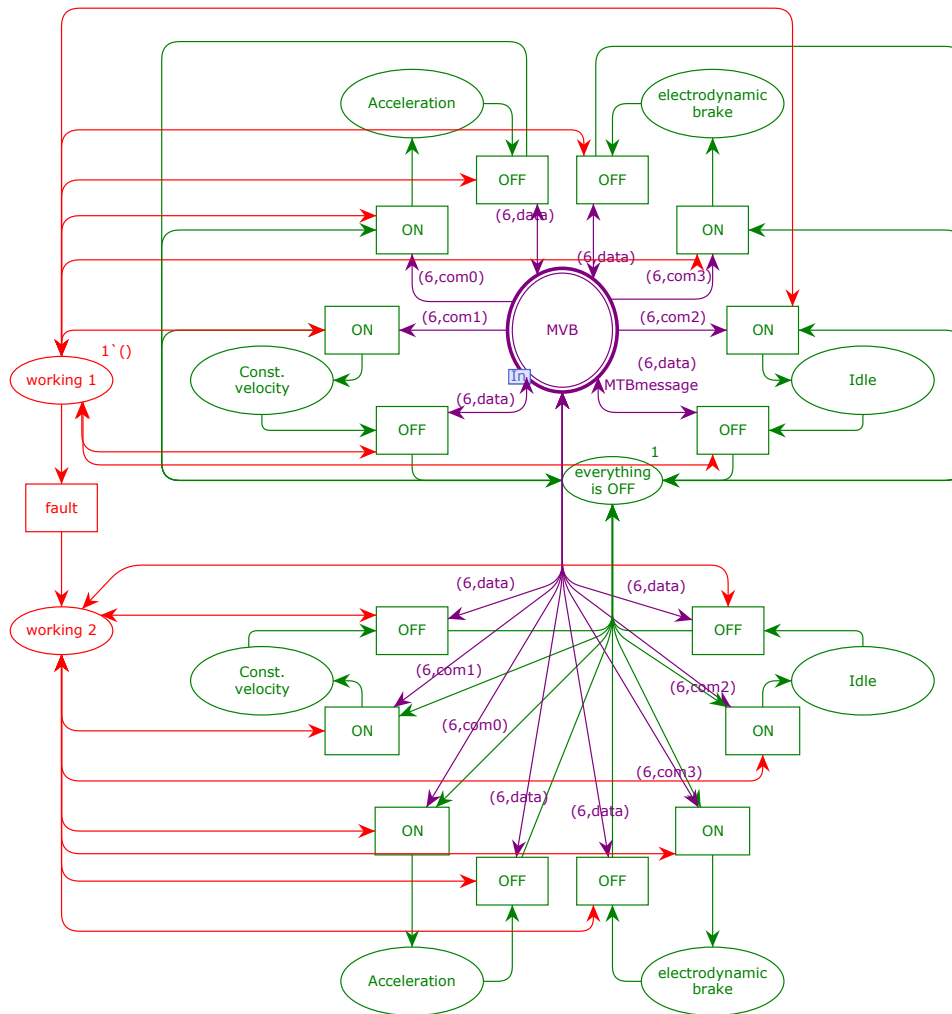
V řídicím systému musíme přidat část, která bude zpracovávat přijaté zprávy a patřičně na ně reagovat. Toho docílíme tak že zprávy roztrídíme podle kategorie. V případě první kategorie pouze zprávy uložíme tím, že token se zprávou pošleme do místa *Save faults*. Na druhý typ zpráv musíme reagovat tak, že nepovolíme odjezd ze zastávky, jakmile se tam vlak dostane. U posledního typu zpráv bude řídicí systém reagovat spuštěním brždění.



Obrázek 7.2: Řešení poruch řídicím systémem

7.3.3 Redundance

Nyní si tuto problematiku přidáme do našeho modelu. Začneme s redundancí. Konkrétněji si zdvojíme subsystém trakce. Subsystém tedy zkopírujeme a vložíme vedle stávajícího. U původního motoru nyní musíme vyřešit jak signalizovat poruchu a realizovat přepnutí na druhý subsystém. To realizujeme tak, že všechny stávající přechody, které jsou propojené se sběrnici, propojíme obousměrně s novým místem *Working 1*. V tomto místě bude token signalizující, že subsystém je funkční. Poruchu subsystému budeme simulovat tak, že místo *Working 1* připojíme k přechodu *fault*. Pokud dojde k vykonání přechodu, odstraní se token z místa *working 1* a přejde do místa *Working 2*. To je propojené s druhým subsystémem stejným způsobem jako je to v prvním. Tímto docílíme toho, že se všechny příkazy, které přichází po sběrnici, přeměrují na druhý subsystém.



Obrázek 7.3: Redundance trakčních motorů

Po upravení schéma můžeme model znovu simulovat tak jak bylo popsáno v předchozí kapitole. K odzkoušení konkrétních nově přidaných částí využijeme možnosti simulace pomocí *Single step*.

Kapitola 8

Závěr

V této práci jsem se zabýval elektrickými motorovými jednotkami. Mým úkolem bylo popsání tohoto typu železničních vozidel a na základě toho vytvoření distribuovaného modelu. Distribuovaný model se dále musel odsimulovat a vyřešit u něj problém s poruchami zařízení.

Při popisu elektrické motorové jednotky byl kladen důraz na vysvětlení funkce jednotlivých subsystémů, jejich vzájemné komunikace a celkovému významu ve vozidle.

Pro tvorbu modelu jsem využil Petriho barevných sítí. Společně s programem CPN tools mi Petriho barevné sítě poskytly ideální nástroj na vytvoření věrohodného modelu železničního vozidla. Model je rozdělen na mnoho částí, které reprezentují subsystémy vozidla. V poslední části byl model resp. modely subsystémů rozšířeny o další zařízení, která simulovala redundanci důležitých částí vozidla. Při simulaci jsem si poté ověřil funkčnost celého modelu.

Výsledkem této práce je ukázka jak fungují železniční vozidla a především elektrické motorové jednotky. Vytvořený model může sloužit nejen k diskrétní simulaci jízdy, ale také k vzdělávacím účelům a lepšímu pochopení procesů dějících se uvnitř elektrických motorových jednotek. Zadání tímto bylo splněno i když by se model dal v budoucnu rozšířit o další části. Například by byla zajímavá simulace spojení více jednotek dohromady nebo přidání dalších méně podstatných subsystémů.



Literatura

- [cita] *Bind manually – cpn tools. cpn tools – a tool for editing, simulating, and analyzing colored petri nets [online]. dostupné z: <http://cpntools.org/2018/01/11/bind-manually/>.*
- [citb] *Coloured petri nets. department of computer science [online]. copyright © [cit. 22.05.2018]. dostupné z: <http://cs.au.dk/cpnets/>.*
- [citic] *Cpn tools, A tool for editing, simulating, and analyzing Colored Petri nets. CPN Tools – A tool for editing, simulating, and analyzing Colored Petri nets [online]. Dostupné z: <http://cpntools.org/>.*
- [citd] *Create a timed cp-net – cpn tools. cpn tools – a tool for editing, simulating, and analyzing colored petri nets [online]. dostupné z: <http://cpntools.org/2018/01/09/create-a-timed-cp-net/>.*
- [cite] *Run a simulation – cpn tools. cpn tools – a tool for editing, simulating, and analyzing colored petri nets [online]. dostupné z: <http://cpntools.org/2018/01/15/run-a-simulation/>.*
- [Com99] International Electrotechnical Commission, *International standart iec 61375-1*, Part 1: Train Communication Network.
- [Dan08a] Jiří Danzer, *Elektrická trakce 1 přehled problematiky*, 49.
- [Dan08b] ———, *Elektrická trakce 1 přehled problematiky*, 60–62.
- [HK01] Pierre A. Zuber Hubert Kirmann, *The iec/ieee train communication network*, 81–92.
- [JP12] Ivan Rádl Jiří Pohl, *Řízení železničních trakčních vozidel*.

- [Pet81] James L. Peterson, *Petri net theory and the modeling of systems*, Prentice=Hall, Inc, Englewood Cliffs, N.J 07632, 1981.
- [Pin12] Pintscher, *Zařízení vysokého napětí na železničních vozidlech*, 1–20.
- [ČS06] ČSN, *En 15380-2 Železniční aplikace - systém označování kolejových vozidel*.



Příloha A

Obsah CD

- **Model.cpn**
Model vozidla vytvořený v CPN tools.
- **Výsledky simulací**
Textové soubory se záznamem průběhu simulace.
 - **SimulationReport1.txt**
 - **SimulationReport2.txt**
 - **SimulationReport3.txt**
- **Elektronická verze této práce**