



VRIJE
UNIVERSITEIT
BRUSSEL



PROGRAMMEERPROJECT 2

Documentatie fase 3

Gérard Lichterttt
gerard.Lichtert@vub.be
0557513

7 mei 2021

Inhoudsopgave

1	Inleiding	2
2	Functionele vereisten	2
3	ADT's	3
3.1	Track ADT	3
3.2	toelichting	3
3.3	ADT switch	3
3.4	ADT detection block	4
3.5	ADT locomotive	4
3.6	ADT railway	4
3.7	ADT infrabel	4
3.8	ADT gui	4
3.9	ADT nmbs	4
4	Afhankelijkheids diagram	4
5	Beschrijving van API tussen infrabel- en NMBS-component	4
6	Planning	4
7	Logboek	4

1 Inleiding

Dit document bevat het verslag van fase 3 van het vak "Programmeerproject 2". Het behandelt de eerste functionele vereisten en de 3 uitgebreide vereisten. Vervolgens zal het over de gebruikte datastructuren gaan en het afhankelijkheids diagram. Deze datastructuren zullen er voor zorgen dat alle functionaliteiten werken en vanuit de GUI aangeroepen kunnen worden. Verder staat in dit verslag een beschrijving van de API tussen de infrabel- en NMBS-component, de planning en het logboek.

2 Functionele vereisten

In dit verslag worden de functionele vereisten per fase besproken. In de eerste fase werd de GUI gemaakt alsook de command and control. De code van de command en control moeten we zelf niet schrijven maar de code die de hardware aanstuurt wel. De code moet locomotieven kunnen laten starten en stoppen, hun snelheid en rijrichting aflezen en veranderen. Verder moeten we ook de de stand van de wissels kunnen uitlezen en verzetten. Via de detectieblokken detecteren we waar een trein zich bevindt.

De GUI laat de eindgebruiker de toestand van het spoornetwerk en de locomotieven zien. Het laat ook interactie toe met de wissels en locomotieven.

In de tweede fase was het de bedoeling dat we de eerste twee uitgebreide vereisten implementeren. Voor dit project werd gekozen voor botsingpreventie aan de hand van een reservatie en bezettingssysteem. Dit zal ervoor zorgen dat een trein zijn snelheid op nul gezet wordt als het pad dat die gaat afleggen al gereserveerd of bezet is. Aangezien de stand van de wissels op eender welk moment gewijzigd kan worden, worden er veel spoorsegmenten gereserveerd tot de aanliggende detectieblokken. Dit is wel met de voorwaarde dat ze op het pad zijn van de richting van de trein. Een andere uitgebreide vereiste waar voor gekozen werd is het automatisch trajectbeheer. De manier waarop dit geïmplementeerd is dat de eindbestemming ingevoerd wordt en het pad berekend word. Dit wordt slechts gedaan als de trein detecteerbaar is. Anders zal het pad nog eens herberekend worden wanneer die detecteerbaar is. Er wordt bij het trajectbeheer ook rekening gehouden met het reservatiesysteem om botsingen te voorkomen.

In de 3e fase is het de bedoeling dat we de laatste uitgebreide vereiste implementeren. Bij dit project is dit de Raspberry Pi vereiste. Dit houdt in dat het mogelijk moet zijn om infrabel te runnen op de Raspberry Pi en NMBS op de computer. Dit wordt behaald door het externe IP adres (of lokaal) mee te geven bij het aanmaken van het NMBS object.

3 ADT's

3.1 Track ADT

Het track ADT houdt bij welke spoorsegmenten verbonden zijn. Het houdt ook bij of het spoorsegment al dan niet gereserveerd is. De reservatiestand kan ook aangepast worden.

Naam	Signatuur	Beschrijving
new	(symbol, list \rightarrow track%)	Maakt een track object aan. Verwacht de ID van het spoorsegment en een lijst met de ID's van de verbonden spoorsegmenten.
get-track-id	(/ \rightarrow symbol)	Geeft de ID van het spoorsegment terug.
track-links	(/ \rightarrow vector)	Geeft de vector terug met de ID's van de verbonden spoorsegmenten.
set-links!	(list \rightarrow /)	verandert de verbonden spoorsegmenten.
links-map	((symbol \rightarrow any) \rightarrow vector)	Voert een procedure uit op de ID's van de verbonden spoorsegmenten en geeft de opgespannen vector terug.
reserved?	(/ \rightarrow symbol \cup false)	Geeft de ID van de locomotief die het spoorsegment gereserveerd heeft of false.
reserve!	(symbol \rightarrow /)	Verandert de reservatiestatus naar het meegegeven ID.
cancel-reservation!	(/ \rightarrow /)	Zet het reservatiestatus op false
links-map!	((symbol \rightarrow any) \rightarrow /)	Voert destructief een procedure uit op de ID's van de verbonden spoorsegmenten en verandert de vector naar de opgespannen vector terug.
nr-of-links	(/ \rightarrow integer)	geeft het aantal verbonden spoorsegmenten terug

Tabel 1: Signaturen van track%

3.2 toelichting

Het track ADT is eigenlijk een superklasse van de volgende ADT's. Dit is zodat de volgende ADT's de methoden erven van het track ADT. New maakt een nieuwe track ADT aan. Dit kan ook vervangen worden door (make-object track% < argumenten >). *get-track-id* dient om de ID op te vragen. *track-links* dient om de vector met verbonden spoorsegmenten op te vragen. *set-links!* wijzigt de verbonden spoorsegmenten. *links-map* voert een procedure uit op de vector-elementen. *reserved?* geeft of de ID van het locomotief terug die het spoorsegment gereserveerd heeft of false wanneer

3.3 ADT switch

Dit bevat alle informatie omtrent het ADT switch. Dit houdt de stand van de wissels bij, laat toe om deze stand te wijzigen en doet laat hetzelfde

Naam	Signatuur	Beschrijving
new	t5	t6

Tabel 2: banana

3.4 ADT detection block

3.5 ADT locomotive

3.6 ADT railway

3.7 ADT infrabel

3.8 ADT gui

3.9 ADT nmbs

4 Afhankelijkheids diagram

5 Beschrijving van API tussen infrabel- en NMBS-component

6 Planning

7 Logboek