

Programmeerproject 2

Specificatie

Abdullah Sabaa Allil

Rolnummer: 0575278

abdullah.sabaa.allil@vub.be

Academiejaar 2021-2022

Introductie

Dit document beschrijft de specificaties van mijn toepassing van de derde fase voor het vak programmeerproject 2. Deze toepassing bevat de volgende componenten:

- Infrabel die met de simulator communiceert om de informatie over de verschillende elementen te kunnen opvragen of aanpassen.
- NMBS die met Infrabel communiceert om gegevens aan Infrabel op te vragen of aan te passen.
- De GUI die deel uitmaakt van NMBS. De GUI laat het toe om de verschillende functionaliteiten van de toepassing te kunnen gebruiken op een gebruiksvriendelijke manier.

Zowel NMBS als Infrabel houden hun eigen model van de spooropstelling bij. De spooropstelling van Infrabel kan rechtstreeks met de simulator communiceren en laat dus toe om informatie aan de simulator op te vragen of aan te passen.

Ik heb de volgende extra vereisten voor deze fase gekozen: **Andere opstellingen, tijdstabellen uitlezen uit een bestand** en **Infrabel die op de Raspberry Pi draait**.

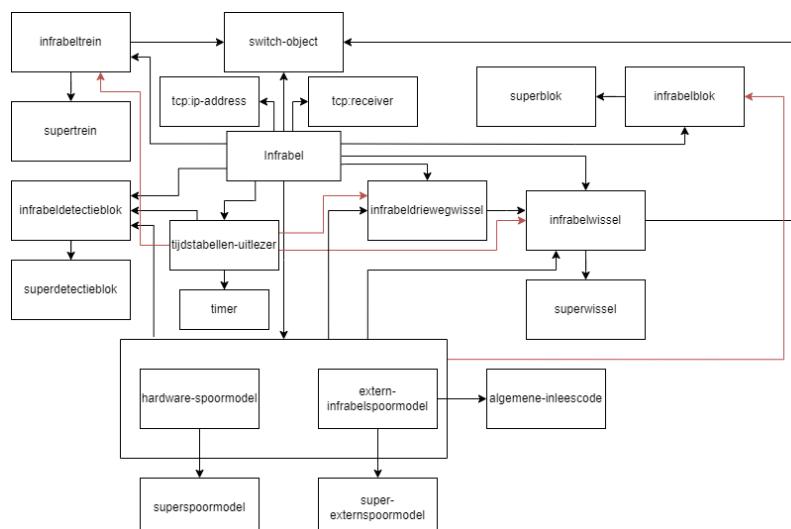
Om plaats te besparen en om niet te veel in herhaling te vallen zullen een aantal zaken die in de specificatie van de tweede fase uitgelegd zijn hier niet opnieuw vermeld worden, aangezien ze niet veranderd zijn.

Beschrijving van gehele software-architectuur

Er wordt gebruikgemaakt van superklassen om codeduplicatie te vermijden. Deze superklassen worden gebruikt om de verschillende elementen bij NMBS en Infrabel voor te stellen. In Infrabel is het bijvoorbeeld de bedoeling dat de verschillende componenten met de simulator communiceren, daarom worden procedures van die superklassen daar overriden door een andere implementatie die dan met de simulator communiceert.

Een pijl tussen twee ADT's betekent dat het ADT waarvan de pijl vertrekt, afhankelijk is van het ADT waar de pijl naartoe gaat

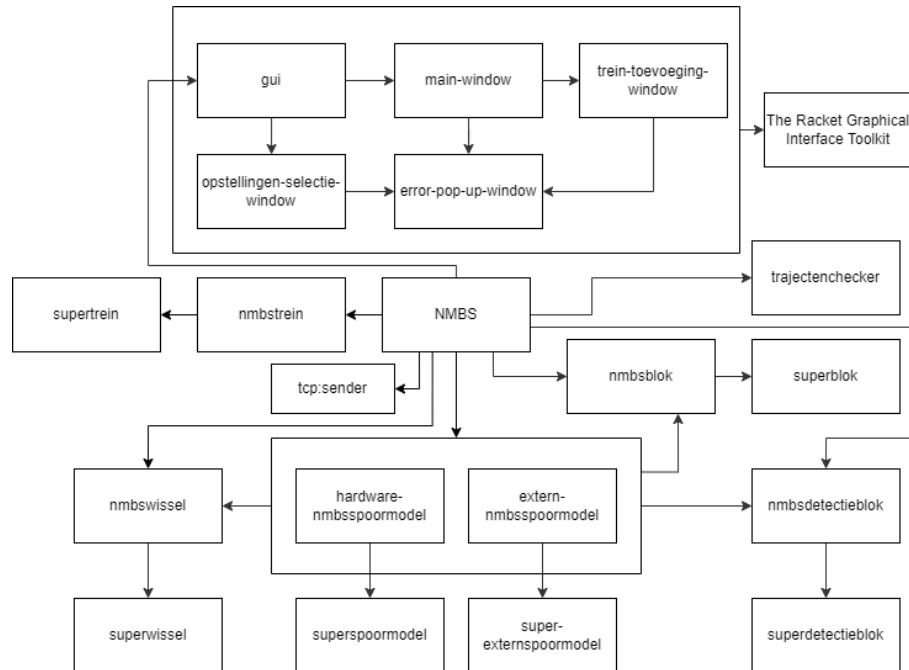
Infrabel



Figuur 1: Het afhankelijkheidsdiagram van het Infrabel component

- Het hardwarespoormodel en het extern-infrabelspoormodel zijn afhankelijk van infrabeldriewegwissel, infrabelwissel en infrabeldetectieblok omdat ze die ADT's gebruiken om het spoormodel aan te maken.
- Het infrabeldriewegwissel ADT is afhankelijk van het infrabelwissel ADT omdat een driewegwissel in Infrabel voorgesteld wordt als een combinatie van 2 gewone wissels.
- Infrabel is afhankelijk van receiver en ip-address omdat Infrabel als server dient waarmee NMBS via het IP adres van Infrabel kan verbinden en dus ook communiceren.

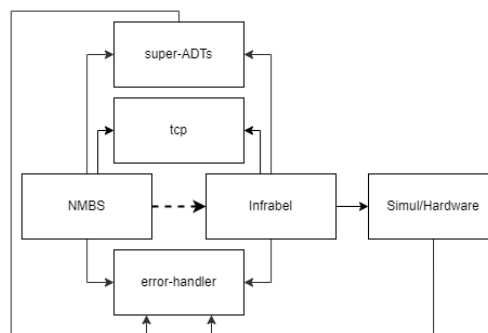
NMBS



Figuur 2: het afhankelijkheidsdiagram van het NMBS component

- Het NMBS ADT is afhankelijk van de GUI module omdat de gui deel uitmaakt van NMBS.
- Het NMBS ADT is afhankelijk van het trajectenchecker ADT om de trajecten de uitgelezen tijdstabellen syntactisch te checken en om na te gaan of de trajecten wel bestaan.
- Het NMBS ADT is afhankelijk van het sender ADT omdat NMBS als klant met Infrabel (server) communiceert.

Algemeen



Figuur 3: algemeen afhankelijkheidsdiagram

- NMBS en Infrabel communiceren met elkaar door middel van TCP. NMBS stuurt berichten naar Infrabel en Infrabel reageert op die berichten.
- NMBS en Infrabel zijn afhankelijk van de super-ADT's. Dit zijn de superklassen waarop de verschillende componenten van infrabel en NMBS zoals treinen en wissels zich baseren.
- NMBS, Infrabel en de super ADTs zijn allemaal afhankelijk van de error-handler. De error handler zorgt voor het afhandelen van verschillende user-side errors. Dit zijn errors die veroorzaakt zijn omdat de gebruiker de toepassing verkeerd heeft gebruikt. Die errors worden dan met de gebruiker gecommuniceerd.
- NMBS en Infrabel zijn afhankelijk van de TCP componenten omdat ze als aparte programma's moeten runnen en met elkaar moeten communiceren via TCP in een server-client model, waarbij Infrabel de server is, en NMBS de client.

Beschrijving van de componenten

Simulator

In de vorige fase werd de simulator aangepast om externe opstellingen te ondersteunen. De volgende zaken zijn aangepast:

- Een nieuwe module *piko-tracks.rkt* die alle piko tracks bevat.
- Een nieuwe subklasse van *railway*, namelijk *external-railway*. Deze subklasse zorgt voor het inladen van externe opstellingen naar de simulator.
- In *tracks.rkt* werd een nieuwe klasse gemaakt, namelijk *threewayswitch%*. Deze klasse stelt driewegwissels voor.

Algemeen

Error-handler

De error handler zorgt voor het afhandelen van user-side errors.

Procedure	Signatuur	Beschrijving
get-error	$(\emptyset \rightarrow string)$	zal het errorbericht dat overeenkomt met de geraiste error teruggeven.
error-to-CLI	$(\emptyset \rightarrow number)$	zal de error naar het REPL (terminal) printen en het getal van de error teruggeven
raise-error	$(number . any \rightarrow false)$	zal een error raisen gegeven zijn getal en eventuele extra informatie.

Tabel 1: de procedures van de error-handler

Algemene-inleescode

Deze code wordt gebruikt in de spoormodellen en in de simulator om spooropstellingen te kunnen inlezen. Deze code is vooral bedoeld om codeduplicatie te vermijden en om de code van opstellingen inlezen te veralgemenen.

TCP

De tcp module is een abstractie bovenop de ingebouwde racket tcp module. Deze module zorgt voor een duidelijke en eenvoudige setup van een tcp server of klant. Deze module heeft ook handige submodules om het IP adres van de server (ervan uitgaande dat de server op de raspberrypi draait) te zoeken op het netwerk, en om het IP adres van het toestel waarop de server draait op te vragen. Deze module bevat de volgende submodules:

DNS-resolver

Procedure	Signatuur	Beschrijving
-----------	-----------	--------------

resolve-DNS	$(string \rightarrow string)$	Gegeven een domeinnaam (bijvoorbeeld raspberrypi.local) wordt het ip adres van deze domeinnaam <i>geresolved</i> en teruggegeven. Dit wordt gebruikt om het ip adres van de server (de rpi) op te zoeken in het netwerk
-------------	-------------------------------	---

Tabel 2: de procedures van de DNS-resolver

ip-address

Procedure	Signatuur	Beschrijving
get-ip-address	$(\emptyset \rightarrow string)$	Geeft het ip adres van het toestel terug.
check-ipv4	$(string \rightarrow string \cup \#f)$	Checkt de validiteit van een gegeven ipv4 adres.
get-pi-ip	$(\emptyset \rightarrow string)$	Geeft het ip adres van de raspberrypi op hetzelfde netwerk terug.

Tabel 3: de procedures van ip-address

reciever

Procedure	Signatuur	Beschrijving
maak-receiver	$(number\ dispatch \rightarrow receiver)$	Gegeven een poortnummer, waarop gecommuniceerd moet worden, en een dispatch waarnaar berichten gestuurd moeten worden, wordt een receiver (server-side) gemaakt.
check-inkomende-berichten!	$(\emptyset \rightarrow \emptyset)$	Checkt berichten van de klanten die naar de receiver gestuurd zijn, en beantwoordt de berichten. Deze procedure dient in een loop gebruikt te worden zodat er voortdurend geluisterd wordt naar klanten.
sluit-connectie	$(\emptyset \rightarrow \emptyset)$	Sluit de tcp connectie van de listener.

Tabel 4: de procedures van de receiver

sender

Procedure	Signatuur	Beschrijving
maak-sender	$(string\ number \rightarrow sender)$	Gegeven een ip adres en een poortnummer wordt een verbinding met een server aangemaakt.
send	$(symbol \rightarrow any)$	Een bericht sturen naar de server gegeven een symbool. De server kan ook iets terugsturen, anders wordt het symbool 'geen-inkomend' teruggestuurd.
get-ip-adres	$(\emptyset \rightarrow string)$	Het ip adres van de sender opvragen.

Tabel 5: de procedures van de sender

Timer

Een object dat een countdown timer voorstelt.

Procedure	signatuur	beschrijving
start!	$(number \rightarrow \emptyset)$	De timer starten gegeven hoeveel seconden de timer moet aftellen.
time-passed?	$(\emptyset \rightarrow boolean)$	een boolean die aangeeft of de tijd gepasseerd is.

Tabel 6: de procedures van het timer ADT

Superklasses

Superklasses worden gebruikt om de gedeelde functionaliteit van ADT's die in Infrabel en NMBS gebruikt worden bij te houden. Op die manier wordt codeduplicatie vermeden en is de code onderhoudsvriendelijk en uitbreidbaar.

supertrein

het supertrein ADT bevat de volgende operaties:

Procedure	Signatuur	Beschrijving
maak-supertrein	$(symbol\ symbol\ symbol \rightarrow supertrein\ ADT)$	maakt, gegeven zijn ID, de ID van het vorig segment en de ID van het beginsegment waaruit de trein vertrekt, een supertreinobject aan.
snelheid pas-snelheid-aan!	$(\emptyset \rightarrow number)$ $(number \rightarrow emptyset)$	
locatie pas-locatie-aan!	$(\emptyset \rightarrow symbol)$ $(symbol \rightarrow \emptyset)$	
id	$(\emptyset \rightarrow symbol)$	
vooruit? rijd-vooruit! rijd-achteruit!	$(\emptyset \rightarrow boolean)$ $(\emptyset \rightarrow \emptyset)$ $(\emptyset \rightarrow \emptyset)$	passen vooruit? aan naar resp. #t en #f en roepen elk pas-rijrichting-aan! op om de richting te veranderen.
omgekeerd? pas-rijrichting-aan!	$(\emptyset \rightarrow boolean)$ $(\emptyset \rightarrow \emptyset)$	zorgt ervoor omgekeerd? niet omgekeerd? wordt. Op die manier rijdt de trein in de "omgekeerde" richting van de huidige richting.

Tabel 7: procedures van het supertrein adt

Er zijn vier variabelen: snelheid, omgekeerd?, vooruit? En detectieblok (locatie). Deze worden aangepast door de verschillende procedures.

Superblok

Procedure	Signatuur	Beschrijving
maak-blok	$(symbol \rightarrow \emptyset)$	Aanmaakoperatie. Verwacht een ID.
set-buren!	$(list \rightarrow \emptyset)$	De burens van het blok zetten (syntax: (list begseg eindseg))
begseg	$(\emptyset \rightarrow symbol)$	Begin en eindburens opvragen
eindseg	$(\emptyset \rightarrow symbol)$	De ids buursegmenten (begseg en eindseg) als een lijst van strings opvragen.
get-naburige-segmenten-ids	$(\emptyset \rightarrow list)$	

Tabel 8: procedures van het superblok ADT

Superdetectieblok

Een superblok dat weet welke trein zich daarop bevindt

Procedure	Signatuur	Beschrijving
maak-superdetectieblok	$(symbol \rightarrow \emptyset)$	Aanmaakoperatie. Verwacht een ID.
trein-aanwezig?	$(\emptyset \rightarrow symbol \cup \#f)$	Geeft #f terug indien geen trein aanwezig, de ID van de trein indien die aanwezig is.
set-aanwezige-trein!	$(symbol \rightarrow \emptyset)$	

zelfde procedures als superblok

Tabel 9: procedures van het superdetectieblok ADT

Superwissel

Procedure	Signatuur	Beschrijving
-----------	-----------	--------------

maak-superwissel	$(symbol \rightarrow superwissel\ ADT)$	maakt, gegeven een ID, een superwisselobject aan
wisselstand	$(\emptyset \rightarrow number)$	Geeft de wisselstand terug.
pas-wisselstand-aan!	$(number \rightarrow \emptyset)$	Past de wisselstand aan.
set-buren! buur?	$(list \rightarrow \emptyset)$ $(symbol \rightarrow \emptyset)$	Buren van de wissel initialiseren (syntax: (list begseg eindseg1 eindseg2))

Tabel 10: procedures van het superwissel ADT

Superspoormodel

Het superspoormodel houdt de verschillende spoorelementen bij zoals de detectieblokken en de wissels. Momenteel worden ze bijgehouden in vectorobjecten, maar later zal er gebruikgemaakt worden van grafen om de relaties tussen de verschillende spoorcomponenten voor te stellen.

Procedure	signatuur	Beschrijving
maak-superspoormodel	$(\emptyset \rightarrow \emptyset)$	
get-detectieblokken-ids	$(\emptyset \rightarrow list)$	geven lijsten van strings van de ids terug.
get-wissels-ids	$(\emptyset \rightarrow list)$	
get-detectieblokken-ids	$(\emptyset \rightarrow list)$	geven lijsten van de ids terug.
get-wissels-ids	$(\emptyset \rightarrow list)$	
get-detectieblok-buren	$(number \rightarrow list)$	
vind-detectieblok	$(number \rightarrow detectieblok)$	
update-detectieblokken!	$(symbol\ symbol\ symbol \rightarrow \emptyset)$	
is-begseg-van-detectieblok?	$(number\ symbol \rightarrow boolean)$	
setup-segmenten	$(list \rightarrow \emptyset)$	
vind-wissel	$(number \rightarrow wissel)$	
pas-wissel-aan!	$(symbol\ any \rightarrow \emptyset)$	
initialiseer	$(list\ lambda\ list\ lambda\ list \rightarrow \emptyset)$	

Tabel 11: procedures van het superspoormodel ADT

- setup-segmenten neemt een lijst van lijsten aan als argument. Elke sublijst heeft de vorm (list id (begid) (eindids)) waarbij eindids ofwel 1 id (bij gewone (detectie)blokken is ofwel meerdere (bij wissels). Deze procedure zal dan de verschillende segmenten updaten zodat ze hun buren kennen.
- pas-wissel-aan! zal een wissel in de opslagruimte van de wissels aanpassen naar een ander object dat meegegeven wordt bij oproep van deze procedure. Deze procedure wordt in Infrabel gebruikt om driewegwissels die voorgesteld worden als een tweewegwissel te vervangen door een driewegwisselobject.
- initialiseer zal het spoormodel initialiseren en de segmenten instellen. De procedure neemt een lijst van detectieblokken Id's mee en een aanmaakoperatie voor een detectieblok, en zal vervolgens alle detectieblokken aanmaken. initialiseer doet ook hetzelfde voor wissels. De procedure neemt ook een lijst van connecties mee (eentje zoals die van setup-segmenten) en zal de verschillende segmenten met elkaar verbinden.

Externspoormodel

Een subklasse van superspoormodel dat gebruikt wordt bij externe opstellingen. Dit ADT begrijpt dezelfde procedures als superspoormodel.

Procedure	Signatuur	Beschrijving
maak-externspoormodel	$(list\ list\ list\ lambda\ lambda \rightarrow externspoormodel\ ADT)$	neemt een lijst van definities van type segmenten, een lijst van definities van spoorsegmenten, een lijst van

		de connecties en 2 lambda's aan. De eerste lambda bepaalt de argumenten voor de aanmaakoperaties van de verschillende segmenten, en de tweede lambda bepaald welke aanmaakoperatie (welke klasse) nodig is.
--	--	---

Tabel 12: procedures van het superspoormodel ADT

Infrabel

Switch-object

Het switch-object is een singleton-object dat zorgt ervoor dat het gemakkelijk is om tussen de simulator en de hardware te switchen. Dit object bevat de volgende operaties:

Procedure	Signatuur
switch-tussen-simulator/hardware	$(boolean \rightarrow switch - object)$
De resterende procedures hebben dezelfde naam en signatuur als die van de interface van de simulator/hardware.	

Tabel 13: de procedures van het switch-tussen-simulator/hardware ADT

Infrabeltrein

Infrabeltrein is een subklasse van supertrein. Infrabeltrein gaat verschillende operaties overriden om ervoor te zorgen dat hij met de simulator communiceert en dat hij dan informatie krijgt of aanpast.

Infrabelblok

Het infrabelblok is een subklasse van superblok en gebruikt dezelfde operaties als superblok.

Infrabeldetectieblok

Het infrabeldetectieblok is een subklasse van het superdetectieblok en maakt gebruik van de superdetectieblokopertes.

Infrabelwissel

Infrabelwissel is een subklasse van superwissel. De overriden operaties zijn: wisselstand en pas-wisselstand-aan!. Er is ook een nieuwe operatie:

Procedure	signatuur	Beschrijving
maak-infrabelwissel	$(symbol \rightarrow infrabelwissel)$	
verander-wisselstand!	$(\emptyset \rightarrow \emptyset)$	zorgt voor het aanpassen van de wisselstand naar de andere wisselstand.

Tabel 14: de extra procedures van infrabelwissel

Infrabeldriewegwissel

Infrabeldriewegwissel is een wissel die uit 2 infrabelwissels bestaat en die drie standen heeft: 1, 2 en 3. Het ADT bevat de volgende operaties:

Procedure	Signatuur
maak-driewegwissel	$(symbol \text{ infrabelwissel } infrabelwissel \rightarrow driewegwissel)$
wisselstand	$(\emptyset \rightarrow number)$
pas-wisselstand-aan!	$(number \rightarrow \emptyset)$
verander-wisselstand!	$(\emptyset \rightarrow \emptyset)$
set-buren!	$(list \rightarrow \emptyset)$
buur?	$(symbol \rightarrow boolean)$

Tabel 15: extra procedures van infrabeldriewegwissel

hardware-spoormodel

Het hardware-spoormodel is een subklasse van het infrabelspoormodel. Het hardware-spoormodel override de initialiseer functie, zodat de initialisatie van het spoormodel gebruikmaakt van objecten uit Infrabel en ook de spoorsegmenten van de hardwareopstelling met elkaar verbindt.

Procedure	Signatuur
Maak-hardwarespoormodel	$(\emptyset \rightarrow \emptyset)$
initialiseer	$(\emptyset \rightarrow \emptyset)$

Tabel 16: procedures van hardware-spoormodel

externe-infrabelspoormodel

een subklasse van externe-superspoormodel dat ervoor zorgt dat aanmaakoperatie van Infrabel componenten gebruikt worden. De aanmaakoperatie is maak-externe-infrabelspoormodel en neemt 3 lijsten aan: definities van de types segmenten, definities van de segmenten zelf, en de connecties tussen de segmenten.

Tijdstabellen-uitlezers

Een ADT dat zorgt voor het uitlezen en runnen van tijdstabellen.

Procedure	Signatuur
maak-tijdstabellen-uitlezers	$(\lambda \lambda \lambda \lambda \lambda \rightarrow \text{tijdstabellen} - \text{uitlezers ADT})$
run-tijdstabellen!	$(\emptyset \rightarrow \emptyset)$
maak-tijdstabellen	$(\text{list} \rightarrow \text{list})$
verwijder-tijdstabel!	$(\text{symbol} \rightarrow \emptyset)$

Tabel 17: procedures van de tijdstabellen-uitlezers

- maak-tijdstabellen-uitlezers verwacht 4 lambdas's: een procedure om een trein te vinden, een procedure om een detectieblok te vinden, een procedure om een wissel te vinden, en een procedure om een trein aan te sturen.
- maak-tijdstabellen neemt een ruw tijdstabel binnen, zet het om naar het gebruikte formaat binnen het ADT, en geeft een lijst terug van de nieuwe treinen en hun beginposities. Deze lijst wordt in NMBS gebruikt om te weten welke treinen aangemaakt worden.
- verwijder-tijdstabel! zal, gegeven een symbool dat de ID van de trein is, het tijdstabel voor deze trein verwijderen.

Infrabel

Het Infrabel ADT is een abstractie voor de component die verantwoordelijk is voor de communicatie met de simulator en het doorlopend beheer van de infrastructuur. Dit ADT bevat de volgende operaties:

Procedure	Signatuur
maak-infrabel	$(\emptyset \rightarrow \emptyset)$
start-hardware-opstelling! laad-opstelling-in! activeer! deactiveer!	$(\emptyset \rightarrow \emptyset)$ $(\text{list list list} \rightarrow \text{boolean})$ $(\emptyset \rightarrow \emptyset)$ $(\emptyset \rightarrow \emptyset)$
stuur-trein! verwijder-trein!	$(\text{symbol number number symbol} \rightarrow \text{symbol})$ $(\text{symbol} \rightarrow \emptyset)$
trein-aanwezig?	$(\text{symbol} \rightarrow \text{boolean})$
pas-trein-snelheid-aan!	$(\text{symbol number} \rightarrow \emptyset)$
rijd-vooruit!	$(\text{symbol} \rightarrow \emptyset)$

- een lambda die een lijst van lijsten teruggeeft die de ID's van treinen, detectieblokken en wissels als strings bevat. Deze lijsten worden gebruikt om selectiemenu's voor te stellen.
- Een lambda om opstellingen in te laden.
- Een lambda om de hardware opstelling te activeren.
- Een lambda uit NMBS om te zorgen voor het toevoegen van een trein.
- Een lambda uit NMBS om te zorgen voor het verwijderen van een trein.
- Een lambda die zorgt voor het zoeken naar een trein en het teruggeven van de treininformatie en callbacks.
- Een lambda die zorgt voor het zoeken naar een detectieblok en het teruggeven van de detectieblok informatie.
- Een lambda die zorgt voor het zoeken naar een wissel en het teruggeven van de wisselinformatie en callbacks.
- Een lambda om trajecten in te laden.
- Een lambda om een verbinding te maken en te activeren
- Een lambda om het IP adres van Infrabel (Raspberry PI) in het netwerk te zoeken.

Opmerking: De gui bestaat uit verschillende modules die elk een venster voorstelt, zoals main-window. Deze worden hier niet verder beschreven omdat het anders veel te lang duurt.

NMBS

Het NMBS ADT is verantwoordelijk voor de functionaliteit die geen logisch onderdeel is van de infrastructuur maar bovenop bouwt. Dit ADT bevat de volgende operaties:

Procedure	Signatuur
maak-nmbs	$(\emptyset \rightarrow NMBS)$
start	$(\emptyset \rightarrow \emptyset)$
stop	$(\emptyset \rightarrow \emptyset)$

Tabel 21: procedures van het NMBS ADT.

Communicatieprotocols

Infrabel en NMBS werken volgens een server-client model. Infrabel dient als server die op bijvoorbeeld een Raspberry Pi draait. NMBS is dan een klant die met de Infrabel communiceert. Deze communicatie gebeurt via TCP. Infrabel ontvangt berichten uit NMBS die via TCP gestuurd zijn en kan vervolgens dingen aanpassen, of iets naar NMBS sturen.

Infrabel maakt gebruik van het receiver ADT om de TCP server te creëren. NMBS maakt gebruik van het sender ADT om de TCP klant te creëren. Infrabel aanvaardt dezelfde berichten als het ADT zelf. De procedures van beide ADTs worden hierboven vermeld.

Infrabel stuurt altijd getallen, symbolen, booleans of lijsten van getallen en symbolen terug naar NMBS wanneer nodig. NMBS doet ook hetzelfde wanneer er argumenten doorgegeven moeten worden naar Infrabel.