WETENSCHAPPEN &
BIO-INGENIEURSWETENSCHAPPEN

VOORSTUDIE
PROGRAMMEERPROJECT 2

Gérard Lichtert

Computerwetenschappen

Brussel, Elsene 1050

Inhoudsopgave

Inleiding

Dit document bevat de voorstudie van het opleidingsonderdeel Programmeerproject 2. Het behandelt eerst de functionele vereisten van het project en vervolgens documenteert het de abstracte datastructuren die ik denk te gebruiken. Deze datastructuren zullen ervoor zorgen dat alle functionaliteiten werken en vanuit de GUI aangeroepen kunnen worden. Verder staat in dit verslag ook het afhankelijkheid diagram van de ADTS , een beschrijving van de API tussen de Infrabel- en NMBS-component en nog een planning die projecteert wanneer ik welk component implementeer.

Functionele vereisten

In deze voorstudie worden de functionele vereisten per fase besproken. In de eerste fase wordt de GUI gemaakt alsook de command en control. De code van de command en control moeten we zelf niet maken maar de code die de hardware aanstuurt wel. De code moet locomotieven kunnen laten starten en stoppen, en hun snelheid en rijrichting aflezen en veranderen. Verder moeten we ook 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 zien. Dit houdt in dat we altijd de snelheid van locomotieven kunnen zien alsook de stand van de wissels en welke detectieblokken bezet zijn of gereserveerd. De GUI zal de gebruiker ook toelaten om functionaliteiten van Infrabel te gebruiken. Hiervoor zullen knoppen of dropdownmenu's zijn die bijhorende functies uitvoeren. Tenslotte zal de GUI ook een logboek bijhouden van de laatste evenementen alsook de stand van actieve detectieblokken (gereserveerd of in gebruik).
In de tweede fase beginnen we met de uitgebreide vereisten maar nog steeds alleen met de simulator. Als twee vereisten heb ik gekozen voor het automatische trajectberekening en het voorkomen van botsingen. Voor het voorkomen van botsingen ga ik met een reservatiesysteem werken. Een locomotief kan een detectieblok in gebruik stellen en reserveert op het zelfde moment het volgend detectieblok. Zodra een ander locomotief een gereserveerd stuk wil reserveren of een gebied betreedt dat al in gebruik is zullen beiden locomotieven stilvallen.
Voor de automatische trajectberekening ga ik gebruik maken van de graffen van algoritme en datastructuren 2. Ik heb met het ADT tracks al een artisanaal graaf gemaakt maar dit zal uit eindelijk veel beter zijn met de implementatie van de graffen.
In de derde fase zal ik mijn laatste uitgebreide vereisten implementeren. Ik kies voor het gebruiken van een Raspberry Pi die dan infrabel draait terwijl mijn computer de NMBS draait. De bedoeling is dan dat ik de NMBS op mijn computer gebruik om te communiceren met het infrabel op de Raspberry Pi die vervolgens communiceert met de hardware.

ADT's

ADT Switch

Dit ADT bevat alle informatie over de switches op het railnetwerk. De switches houden een bepaalde stand aan die afgelezen kan worden en veranderd kan worden. Ook kunnen we de 2 tracks waarbij deze switch uitkomt hiervan aflezen eenmaal dat we deze erbij hebben gezet.

Naam:	Signatuur:	Beschrijving:
New-switch	(symbol, track or null, track or null →/)	Maakt een nieuwe switch aan.
Id	(/ → symbol)	Geeft de id terug van de switch.
Switch?	(/ → Boolean)	Geeft terug of het een switch is.
Read-status	(/ → symbol)	Geeft de status van de switch terug, dit kan 'open of 'closed zijn.
Set-status-closed!	(/ → /)	Zet de status van de switch op 'closed.
Set-status-open!	(/ → /)	Zet de status van de switch op 'open.
Current-linked-track	(/ → track)	Geeft de de huidige track terug waar een locomotief naartoe zou gaan afhankelijk van de status van switch.
Left-track	(/ → track)	Geeft de de linkertrack terug.
Right-track	(/ → track)	Geeft de rechtertrack terug.
Set-left-track!	(track → /)	Verandert de linkertrack.
Set-right-track!	(track → /)	Verandert de rechtertrack.

ADT Detection-block

Dit ADT bevat alles omtrent het gedrag van detectieblokken, van het reserveren tot nagaan welke trein zich als laatst dat spoorsegment bevindt.

Name:	Signature:	Beschrijving:
New-detection-block	(Symbol → detection-block)	Maakt een nieuwe detectieblok aan.
Id	(/ → symbol)	Geeft de id van de detectieblok terug.
occupied?	(/ → boolean)	Geeft terug of het spoor bezet is.
Last-train-passed	(/ → symbol)	Geeft de id terug van de laatste trein die op het spoor gepasseerd is.
Set-to-occupied!	(symbol → /)	Zet het spoor naar bezet, slaagt de id van de trein op en wijzigt de status van reserved? naar niet gereserveerd.
Set-to-vacant!	(/ → /)	Zet occupied? Naar #f.
Reserved?	(/ → boolean)	Geeft terug of het huidige spoor gereserveerd is.
Reserve!	(/ → /)	Reserveert het spoorsegment en zet reserved? Naar #t

ADT Locomotive

Dit ADT staat in van het aanmaken en beheren van het gedrag van locomotieven.

Name:	Signature:	Beschrijving:
New-locomotive	(Symbol, symbol→ locomotive)	Maakt een nieuw locomotief aan: verwacht de id van de locomotief en de id van het spoor waar de locomotief begint
Id	(/ → symbol)	Geeft de id van de locomotief terug
Get-speed	(/ → integer)	Geeft de snelheid van de locomotief terug
Set-speed!	(integer → /)	Verandert de snelheid van de locomotief
Get-direction	(/ → symbol)	Geeft de richting terug
Set-direction!	(symbol → /)	Verandert de richting van de locomotief
Get-destination	(symbol → track)	Geeft de destinatie van de locomotief terug
Set-destination!	(symbol → track)	Zet de destinatie van de locomotief naar de meegegeven spoor-id.
Set-location!	(symbol → track)	Veranderd de locatie van de trein, dit zal samenwerken met detectieblokken om de laatst gekende locatie terug te kunnen krijgen.
Get-last-known-location	(/ → symbol)	Geeft het spoor-id terug van de laatst gekende locatie.

ADT Track

Het ADT track is een koppeling tussen alle spoorsegmenten. Het kan een detectieblock, wissel of geen van beiden zijn. Dit ADT zal mogelijks vervangen worden door een graf maar dit is hoe dat ik denk dat ik een netwerk kan maken.

Name:	Signature:	Beschrijving:
new-track	(symbol, detection-block or switch or #f → /)	Maakt een nieuwe track aan met een id en 1 van de volgende: detection-block, switch of #f.
id	(/ → symbol)	Geeft de id van de track terug.
interact-with-adt	(/ → detection-block or switch or #f)	Geeft het ADT terug die opgeslagen is in de track of #f als de track geen speciaal track is.
next-track	(/ → track)	Geeft de volgende track terug.
set-next-track!	(track → /)	Verandert de volgende track naar de meegegeven track.
previous-track	(/ → track)	Geeft de vorige track terug.
set-previous-track!	(track → /)	Verandert de vorige track naar de meegegeven track.
has-next-track?	(/ → boolean)	Geeft terug of er een volgende track.
has-previous-track?	(/ → boolean)	Geeft terug of er een vorige track is.
type	(/ → symbol)	Geeft het type track terug als symbool.

Toelichting:

Als een track een switch is zal het géén next-track hebben. De bedoeling is dan dat we met het adt van de switch moeten interageren om de verbonden tracks te krijgen. Deze tracks zijn wel verbonden met de switch als vorige track om de verbinding niet te doorbreken.

ADT Railway

Bij het ADT Railway zal het compleet spoor gedefinieerd worden en worden de spoorsegmenten gelinkt met elkaar. Via het ADT Railway kunnen we nu interageren met het spoornetwerk

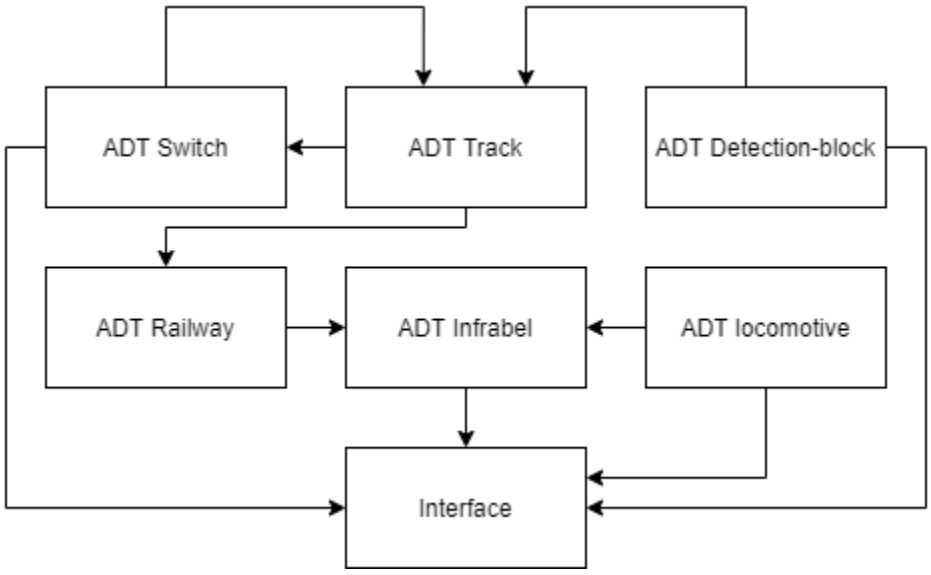
Name:	Signature:	Beschrijving:
New-railway	(/ → railway)	Maakt een nieuw spoornetwerk
Get-adt-track	(symbol → track)	Geeft een de track terug van de ingevoerde id
Add-track!	(track → /)	Voegt een spoorsegment toe aan het netwerk
Remove-track!	(symbol → /)	Verwijdert een spoorsegment van het netwerk

ADT Infrabel

Bij het ADT infrabel komen de treinen en het spoornetwerk samen. Hieruit kunnen we de simulator of hardware aanspreken

Name:	Signature:	Beschrijving:
new-infrabel	(/ → /)	Maakt een nieuw infrabel aan
start	(boolean →)	Start de simulator(#t) of hardware (#f)
stop	(/ → /)	Stopt de simulator of hardware
Get-locomotive-adt	(symbol → locomotive)	Geeft een trein terug van met de gezochte id
Add-locomotive	(symbol, symbol → /)	Maakt een nieuw locomotief aan en slaagt deze op in het databank
Remove-locomotive	(symbol → /)	Verwijdert de locomotief met gegeven id
get-switch-adt	(symbol → switch)	Geeft de gezochte switch terug
Get-active-detection-blocks	(/ → list)	Geeft een lijst van id's van bezette detectieblokken terug
Calculate-path	(symbol, symbol → list)	Geeft een lijst van id's terug als path van spoor-id naar een ander spoor-id
Get-log	(/ → list)	Geeft een lijst terug met recente evenementen

Afhankelijkheidsdiagram



Beschrijving van API tussen Infrabel- en NMBS- component

NMBS naar Infrabel	Infrabel terug naar NMBS
Vraagt informatie over een trein aan	Geeft snelheid, laatst opgeslagen locatie, richting en traject terug alsook de optie om de locomotief te starten of stoppen.
Vraagt de stand van een switch aan	Geef de stand van een switch terug
Vraagt laatste evenementen aan	Geeft laatste evenementen terug
Vraagt bezette detectieblokken	Geeft bezette (en gereserveerd) detectie blokken terug en de id van de trein die deze bezet (of heeft gereserveerd)
Vraag om de snelheid van een trein te veranderen	/
Vraag om de richting van een trein te veranderen	/
Vraag om de stand van een switch te wisselen	/

Planning

Week:	Gepland:
5	18/10 Indienen voorstudie
6	Feedback afwachten, ADT detectieblok en switch implementeren
7	ADT locomotive & track implementeren
8	ADT railway & beginnen aan ADT infrabel
9	ADT infrabel & NMBS component maken
10	Laatste aanpassingen NMBS
11	18/10 Indienen fase 1: code en documentatie
12	Feedback afwachten
13	Implementeren van graffen voor het automatisch trajectberekening
14	Implementeren van graffen voor het automatisch trajectberekening
15	Automatisch trajectberekening implementeren
16	Automatisch trajectberekening implementeren
17	Mogelijk maken dat GUI het traject kan tonen aan de hand van de trajectberekening
18	Botsingen voorkomen implementeren
19	Botsingen voorkomen implementeren
20	GUI aanpassen voor mogelijk extra componenten
21	Feedback vragen
22 +23 + 24 +25	Aanpassingen doorvoeren aan de hand van de feedback
26	14/03 Indienen fase 2: code en documentatie
27	Feedback afwachten
28 + 29 +30 +31	RPI
32	GUI aanpassen indien nodig
33	Hardware film maken/aanpassingen doorvoeren indien nodig
34	Hardware film maken/aanpassingen doorvoeren indien nodig
35	Hardware film maken/aanpassingen doorvoeren indien nodig
36	23/05 Indienen fase 3: code en documentatie