Verkeerssimulatie

Documentsoort:	Behoeftespecificatie
Versie:	2.0
Datum:	24 maart 2022
Auteurs:	Brent van Bladel
Status:	In development

1 Samenvatting

Dit document bevat de specificaties voor een informaticasysteem ter ondersteuning van een verkeerssimulatie. Het is geschreven in het kader van het vak "Project Software Engineering"(1ste bachelor informatica - Universiteit Antwerpen).

2 Context

Op 25 juni 2018 heeft het politiek stuurcomité van Antwerpen 18 projecten geselecteerd die voor een overkapping van de Antwerpse ring zullen zorgen. De eerste werken zijn begonnen in de zomer van 2019 en hebben een hoop verkeershinder met zich mee gebracht. Om deze hinder zo goed mogelijk in te perken in het verdere verloop van het project, heeft het Departement Mobiliteit en Openbare Werken geopteerd om een simulatiemodel te laten ontwikkelen dat het verkeer kan simuleren.

De Universiteit Antwerpen is gevraagd dit systeem te ontwikkelen. In de eerste bachelor informatica zal onder de vakken "Computer Graphics" en "Project Software Engineering" gewerkt worden aan dit project. Tijdens de practica Computer Graphics zal de visualisatie van de simulatie ontwikkeld worden, tijdens de practica Project Software Engineering zal gewerkt worden aan de simulatie applicatie zelf.

3 Legende

De behoeftespecificatie is opgesteld aan de hand van zogenaamde use-cases. Elke use-case beschrijft een klein gedeelte van de gewenste functionaliteit. Het is de bedoeling dat tijdens elke fase van het project verschillende van die use cases geïmplementeerd worden. Een typische use-case bevat de volgende onderdelen:

- **Refertenummer & titel:**
  Wordt gebruikt om naar een bepaalde use-case te verwijzen.

- **Prioriteit:**
  De specificatie van een systeem vraagt meer dan wat binnen de voorziene tijd op te leveren is. Vandaar dat we per use-case aangeven in hoeverre die functionaliteit belangrijk is. In volgorde van belangrijkheid kan hier staan: VERPLICHT (deze use-case moet opgeleverd worden), BELANGRIJK (niet essentieel maar bij voorkeur toch opleveren), NUTTIG (interessant maar kan weggelaten worden).

- **Doel:**
  Summiere beschrijving van het waarom van de use-case, t.t.z. wat de use-case bijdraagt tot de gehele functionaliteit.

- **Preconditie:**
  Summiere beschrijving van de uitgangspunten bij aanvang van de use-case.

- **Succesvol einde:**
  Summiere beschrijving van wat opgeleverd zal worden als er niks fout is gegaan.

- **Stappen:**
  Een sequentiële beschrijving van hoe de use-case precies zal verlopen als alles goed gaat (het zogenaamde "happy day scenario"). De stappen zijn genummerd en kunnen controle instructies (WHILE, IF, ...) bevatten.

- **Uitzonderingen:**
  Een lijst van mogelijke probleemgevallen en hoe die behandeld zullen worden. Een probleem geval (a) verwijst naar het nummer van de stap waar het probleem kan optreden, (b) bevat een conditie die aangeeft wanneer het probleemgeval optreedt, (c) omschrijft heel kort (een lijn) hoe het probleem behandeld zal worden.

- **Voorbeeld:**
  Een voorbeeld van wat in- of uitgevoerd kan worden.

Soms is een use-case een uitbreiding van een andere use-case, en dan zijn volgende onderdelen relevant:

- **Uitbreiding:**
  Een referte naar de use-case waarvan deze een uitbreiding is.

- **Stappen:**
  Een lijst van extra en/of aangepaste stappen t.o.v de use-case waarvan deze een uitbreiding is.
  Een uitbreiding (a) verwijst naar het nummer van de stap die uitgebreid wordt, (b) zegt of de uitbreiding voor, na of tijdens de normale stap zal gebeuren, (c) omschrijft wat precies in de uitbreiding zal gebeuren.

4 Overzicht

Use-Case	**Prioriteit**
1: Invoer	
1.1. Verkeerssituatie inlezen	VERPLICHT
1.2. Voertuiggenerator inlezen	BELANGRIJK
1.3. Voertuig met type inlezen	VERPLICHT
1.4. Bushaltes inlezen	BELANGRIJK
1.5. Kruispunten inlezen	BELANGRIJK
2: Uitvoer	
2.1. Simpele uitvoer	VERPLICHT
2.2. Grafische impressie	BELANGRIJK
2.3. Integratie met graphics	BELANGRIJK
3: Simulatie	
3.1. Rijden van voertuig	VERPLICHT
3.2. Simulatie van verkeerslicht	VERPLICHT
3.3. Automatische simulatie	VERPLICHT
3.4. Simulatie met voertuiggenerator	BELANGRIJK
3.5. Rijden van voertuigen met type	VERPLICHT
3.6. Simulatie van bushaltes	BELANGRIJK
3.7. Simulatie van kruispunten	BELANGRIJK
3.8. Simulatie van kruispunten met verkeerslicht	NUTTIG
3.9. Slimme verkeerslichten	NUTTIG
3.10. Simulatie van oranje verkeerslicht	NUTTIG
4: Gebruikersinterface	
4.1. GUI voor simulatie	NUTTIG
4.2. GUI voor verkeerslichten	NUTTIG
4.3. GUI voor voertuiggenerator	NUTTIG

1.1. Verkeerssituatie inlezen

Prioriteit:
VERPLICHT

Doel:
Inlezen van het schema van de verkeerssituatie: de verschillende wegen en de verschillende voertuigen.

Preconditie:
Een ASCII bestand met daarop een beschrijving van de wegen en voertuigen. (Zie Appendix A voor meer informatie over het XML formaat)

Succesvol einde:
Het systeem bevat een schema met de verschillende wegen, en informatie over alle voertuigen.

Stappen:
1. Open invoerbestand
2. WHILE Bestand niet ingelezen
2.1. Herken het soort element (VOERTUIG, BAAN, VERKEERSLICHT)
2.2. Lees verdere informatie voor het element
2.3. IF Verifieer geldige informatie
2.3.1. THEN Voeg element toe aan de simulatie
2.3.1. ELSE Foutboodschap + positioneer op volgende element in het bestand
3. Verifieer consistentie van de verkeerssituatie
4. Sluit invoerbestand

Uitzonderingen:
2.1. [Onherkenbaar element] Foutboodschap + positioneer op volgende element in het bestand ⇒ verdergaan vanaf stap 2
2.2. [Ongeldige informatie] Foutboodschap + positioneer op volgende element in het bestand ⇒ verdergaan vanaf stap 2
3. [Inconsistente verkeerssituatie] Foutboodschap ⇒ verdergaan vanaf stap 4

Voorbeeld:
Een baan met twee auto's en één verkeerslicht:

```
<BAAN>
    <naam>Middelheimlaan</naam>
    <lengte>500</lengte>
</BAAN>

<VERKEERSLICHT>
    <baan>Middelheimlaan</baan>
    <positie>400</positie>
    <cyclus>20</cyclus>
</VERKEERSLICHT>

<VOERTUIG>
    <baan>Middelheimlaan</baan>
    <positie>20</positie>
</VOERTUIG>

<VOERTUIG>
    <baan>Middelheimlaan</baan>
    <positie>0</positie>
</VOERTUIG>
```

1.2. Voertuiggenerator inlezen

Prioriteit:
BELANGRIJK

Doel:
Inlezen van hoe voertuigen gegenereerd moeten worden.

Preconditie:
Een ASCII bestand met daarop een beschrijving van een voertuiggenerator. (Zie Appendix A voor meer informatie over het XML formaat)

Succesvol einde:
Het systeem bevat een schema met een voertuiggenerator.

Uitbreiding:
Use Case 1.1

Stappen:
[2.1, tijdens] Hou rekening met extra element VOERTUIGGENERATOR

Uitzonderingen:
Geen

Voorbeeld:

```
<VOERTUIGGENERATOR>
    <baan>Middelheimlaan</baan>
    <frequentie>5</frequentie>
</VOERTUIGGENERATOR>
```

1.3. Voertuig met type inlezen

Prioriteit:
VERPLICHT

Doel:
Een specifiek voertuig zal zich anders gedragen afhangend van zijn type. Om dit in de simulatie te kunnen opnemen, moet deze data ook ingelezen worden. Zie Appendix C voor meer informatie over de verschillende soorten voertuigen.

Uitbreiding:
Use Case 1.1 en Use Case 1.2

Stappen:
[2, tijdens] Hou rekening met extra mogelijkheden voor 'type'

Opmerkingen:
Indien je Use Case 1.2 ondersteunt, dan zal elke voertuiggenerator ook een attribuut 'type' bevatten. Dit bepaalt welk type voertuigen er gegenereerd zullen worden.

Voorbeeld:
Voorbeeld van voertuig en voertuiggenerator met verschillende types (auto, bus, brandweerwagen, ziekenwagen, en politiecombi).

```
<VOERTUIGGENERATOR>
    <baan>Middelheimlaan</baan>
    <frequentie>5</frequentie>
    <type>auto</type>
</VOERTUIGGENERATOR>

<VOERTUIG>
    <baan>Middelheimlaan</baan>
    <positie>20</positie>
    <type>bus</type>
</VOERTUIG>

<VOERTUIG>
    <baan>Middelheimlaan</baan>
    <positie>40</positie>
    <type>brandweerwagen</type>
</VOERTUIG>

<VOERTUIG>
    <baan>Middelheimlaan</baan>
    <positie>60</positie>
    <type>ziekenwagen</type>
</VOERTUIG>

<VOERTUIG>
    <baan>Middelheimlaan</baan>
    <positie>80</positie>
    <type>politiecombi</type>
</VOERTUIG>
```

1.4. Bushaltes inlezen

Prioriteit:
BELANGRIJK

Doel:
Een baan kan bushaltes bevatten. Om dit in de simulatie te kunnen opnemen, moet deze data ook ingelezen worden.

Uitbreiding:
Use Case 1.1

Stappen:
[2, tijdens] Hou rekening met extra element

Uitzonderingen:
Geen

Voorbeeld:
Voorbeeld van een bushalte.

```
<BAAN>
    <naam>Middelheimlaan</naam>
    <lengte>500</lengte>
</BAAN>

<BUSHALTE>
    <baan>Middelheimlaan</baan>
    <positie>250</positie>
    <wachttijd>20</wachttijd>
</BUSHALTE>
```

1.5. Kruispunten inlezen

Prioriteit:
BELANGRIJK

Doel:
Banen kunnen elkaar kruisen op een kruispunt. Om dit in de simulatie te kunnen opnemen, moet deze data ook ingelezen worden.

Uitbreiding:
Use Case 1.1

Stappen:
[2, tijdens] Hou rekening met extra element

Uitzonderingen:
Geen

Voorbeeld:
Voorbeeld van kruispunten.

```
<BAAN>
    <naam>Middelheimlaan</naam>
    <lengte>500</lengte>
</BAAN>

<BAAN>
    <naam>Floralienlaan</naam>
    <lengte>750</lengte>
</BAAN>

<BAAN>
    <naam>Beukenlaan</naam>
    <lengte>1000</lengte>
</BAAN>

<KRUISPUNT>
<baan positie="250">Middelheimlaan</baan>
    <baan positie="500">Floralienlaan</baan>
</KRUISPUNT>

<KRUISPUNT>
<baan positie="500">Middelheimlaan</baan>
    <baan positie="500">Beukenlaan</baan>
```

```
</KRUISPUNT>
```

2.1. Simpele uitvoer

Prioriteit:
VERPLICHT

Doel:
Uitvoer van alle informatie in de simulatie.

Preconditie:
Het systeem bevat een schema van de virtuele verkeerssituatie.

Succesvol einde:
Het systeem heeft de informatie van de virtuele verkeerssituatie netjes uitgeschreven.

Stappen:
1. Schrijf huidige simulatietijd uit
2. WHILE Nog voertuigen in simulatie
2.1. Schrijf voertuig-gegevens uit

Uitzonderingen:
Geen

Voorbeeld:

```
Tijd: 0
Voertuig 1
 -> baan: Middelheimlaan
 -> positie: 20
 -> snelheid: 16.6

 Voertuig 2
 -> baan: Middelheimlaan
 -> positie: 0
 -> snelheid: 16.6
```

2.2. Grafische impressie

Prioriteit:
BELANGRIJK

Doel:
De toestand van de verkeerssituatie wordt grafisch weergegeven.

Preconditie:
Het systeem is correct geïnitialiseerd.

Succesvol einde:
Het systeem heeft een tekstbestand (ASCII) uitgevoerd, waarin de toestand van de verkeerssituatie staat beschreven.

Variant:
Het systeem heeft een tekstbestand (HTML) uitgevoerd, waarin de toestand van de verkeerssituatie staat beschreven.

Stappen:
1. Open uitvoerbestand
2. Teken gegevens uit voor de toestand van de verkeerssituatie
3. Sluit uitvoerbestand

Uitzonderingen:
Geen

Voorbeeld:
Indien een baan met twee verkeerslichten, twee bushaltes, twee auto's en een bus.

```
Middelheimlaan      | ================A=================B===========A=
 > verkeerslichten  |             |          G           |             R
 > bushaltes        |             B                      B
```

2.3 Integratie met graphics

Prioriteit:
BELANGRIJK

Doel:
Een 3D visualisatie genereren van de verkeerssituatie in de simulatie. Hiervoor kan je een standaard interface voor je graphics engine gebruiken.

Preconditie:
Het systeem is correct geïnitialiseerd.

Succesvol einde:
Elke beweging van voertuigen wordt weergegeven in een 3D omgeving.

3.1. Rijden van voertuig

Prioriteit:
VERPLICHT

Doel:
Simuleren van het rijden van een voertuig. Zie Appendix B voor meer informatie over de formules.

Preconditie:
Het systeem bevat een schema van de virtuele verkeerssituatie. Er is een voertuig op een baan.

Succesvol einde:
De positie, snelheid, en versnelling van het voertuig zijn herberekend.

Stappen:
1. Bereken nieuwe snelheid en positie van voertuig
2. Bereken nieuwe versnelling van voertuig
3. IF nieuwe positie valt buiten huidige baan
3.1. Verwijder voertuig uit simulatie

Uitzonderingen:
Geen

3.2. Simulatie van verkeerslicht

Prioriteit:
VERPLICHT

Doel:
Simuleren van verkeerslichten.

Preconditie:
Het systeem bevat een schema van de virtuele verkeerssituatie. Er is een verkeerslicht op een baan.

Succesvol einde:
Voertuigen passen zich aan afhangend van de staat van het verkeerslicht.

Stappen:
1. IF tijd sinds laatste verandering > cyclus
1.1 THEN verander de kleur van het licht (groen $\leftrightarrow$ rood)
2. IF verkeerslicht is groen
2.1 THEN voertuigen voor het verkeerslicht mag terug versnellen
3.1 IF verkeerslicht is rood
3.1.1 THEN IF het eerste voertuig voor het licht bevindt zich in de vertraagafstand
3.1.1.1 THEN pas de vertraagfactor toe op het voertuig
3.1.2 ELSE IF het eerste voertuig voor het licht bevindt zich in de eerste helft van de stopafstand
3.1.1.1 THEN laat het voertuig stoppen

3.3. Automatische simulatie

Prioriteit:
VERPLICHT

Doel:
Simulatie automatisch laten lopen.

Preconditie:
Het systeem bevat een schema van de virtuele verkeerssituatie.

Succesvol einde:
Het verkeer in het wegennetwerk wordt gesimuleerd.

Stappen:
1. FOR elk voertuig in het wegennetwerk
1.1 voer use case 3.1 uit op het voertuig
2. FOR elk verkeerslicht in het wegennetwerk
2.1 voer use case 3.2 uit op het verkeerslicht

3.4. Simulatie met voertuiggenerator

Prioriteit:
BELANGRIJK

Doel:
Genereren van voertuigen.

Preconditie:
Het systeem bevat een schema van de virtuele verkeerssituatie.
Er is een voertuiggenerator op een baan.

Succesvol einde:
Er worden automatisch voertuigen toegevoegd tijdens de simulatie.

Uitbreiding:
Use Case 3.3

Stappen:
3. FOR elke voertuiggenerator
3.1 IF tijd sinds laatste voertuig > frequentie
3.1.1 IF geen voertuig op baan tussen posities 0 en $2l$
3.1.1.1 THEN voeg voertuig toe aan baan op positie 0

3.5. Rijden van voertuigen met type

Prioriteit:
VERPLICHT

Doel:
Een specifiek voertuig zal zich anders gedragen afhangend van zijn type. Zie Appendix C voor meer informatie over de verschillende soorten voertuigen.

Uitbreiding:
Use Case 3.1

Stappen:
[1-2, tijdens] Hou rekening met de waarden van het type voertuig

Uitbreiding:
Use Case 3.2

Stappen:
[3, tijdens] IF het eerste voertuig voor het verkeerslicht is een prioriteitsvoertuig
[3, tijdens] THEN voertuig hoeft niet te vertragen of te stoppen

Uitzonderingen:
Geen

3.6. Simulatie van bushaltes

Prioriteit:
BELANGRIJK

Doel:
Een baan kan bushaltes bevatten. Voertuigen van het type bus moeten stoppen aan elke bushalte, en vervolgens blijven staan tot de wachttijd verstreken is.
Hiervoor moet je ook use case 1.4 implementeren.

Preconditie:
Het systeem bevat een schema van de virtuele verkeerssituatie. Dit schema bevat minstens 1 bushalte en 1 bus.

Succesvol einde:
De bus stopt aan de bushalte.

Stappen:
1 IF een voertuig met type bus bevindt zich in de vertraagafstand
1.1 THEN pas de vertraagfactor toe op het voertuig
2 ELSE IF een voertuig met type bus bevindt zich in de stopafstand
2.1 THEN laat het voertuig stoppen
3 IF tijd sinds voertuig met type bus gestopt is > wachttijd
2.1 THEN het voertuig mag terug vertrekken

Opmerkingen:
De stappen worden enkel toegepast op de eerste bus voor de bushalte. Indien er meerdere bussen op de baan voor de halte rijden, dan zullen de andere bussen automatisch vertragen omdat het voertuig ervoor ook vertraagt. Pas wanneer de eerste bus terug vertrekt is het nodig om de stappen op de volgende bus toe te passen.

3.7. Simulatie van kruispunten

Prioriteit:
BELANGRIJK

Doel:
Een baan kan kruispunten bevatten. Voertuigen moeten aan een kruispunt kiezen welke baan ze verder volgen.
Hiervoor moet je ook use case 1.5 implementeren.

Preconditie:
Het systeem bevat een schema van de virtuele verkeerssituatie. Dit schema bevat minstens 2 banen en 1 kruispunt.

Succesvol einde:
Voertuigen kiezen een baan aan elk kruispunt.

Stappen:
1 IF een voertuig komt aan een kruispunt
1.1 THEN kies een willekeurige baan van het kruispunt om verder op te rijden

Opmerkingen:
Als de positie van een kruispunt op een baan gelijk is aan de lengte van de baan, dan wil dat zeggen dat de baan eindigt op dat kruispunt. Voertuigen zullen deze baan dus niet kunnen kiezen bij dit kruispunt.
Als de positie van een kruispunt op een baan gelijk is aan nul, dan wil dat zeggen dat de baan begint op dat kruispunt.

3.8. Simulatie van kruispunten met verkeerslicht

Prioriteit:
NUTTIG

Doel:
Verkeerslichten aan een kruispunt houden rekening met elkaar om botsingen te voorkomen.

Preconditie:
Het systeem bevat een schema van de virtuele verkeerssituatie. Dit schema bevat minstens 2 banen en 1 kruispunt met een verkeerslicht op het kruispunt.

Succesvol einde:
Verkeerslichten op een kruispunt houden rekening met elkaar.

Uitbreiding:
Use case 3.7

Opmerkingen:
Indien er op elke baan van een kruispunt een verkeerslicht staat waarvan de positie gelijk is aan deze van het kruispunt, dan zullen deze rekening moeten houden met elkaar. Kies in dit geval 1 van de verkeerslichten om zijn vaste cyclus te volgen. Het andere verkeerslicht zal altijd de tegenovergestelde kleur zijn, los van zijn cyclus.

3.9. Slimme verkeerslichten

Prioriteit:
NUTTIG

Doel:
Verkeerslichten houden rekening met voertuigen op de baan om de doorstroming te verbeteren.

Preconditie:
Het systeem bevat een schema van de virtuele verkeerssituatie. Dit schema bevat minstens 1 kruispunt met verkeerslichten.

Succesvol einde:
Verkeerslichten houden rekening met voertuigen.

Uitbreiding:
Use case 3.2 en Use case 3.8

Opmerkingen:
In plaats van een vast patroon aan te houden, zullen verkeerslichten op een kruispunt proberen om voertuigen zo veel mogelijk groen te geven. Je mag zelf kiezen hoe je dit algoritme implementeert. Wel moet het zich aan de volgende condities houden:

- Het verkeerslicht moet minstens 10 seconden zijn kleur (groen of rood) behouden voordat het terug mag veranderen.
- Het verkeerslicht mag maximaal 60 seconden zijn kleur (groen of rood) behouden voordat het terug moet veranderen.
- Verkeerslichten op een kruispunt hebben steeds een tegenovergestelde kleur.

3.10. Simulatie van oranje verkeerslicht

Prioriteit:
NUTTIG

Doel:
Verkeerslichten met drie kleuren kunnen simuleren.

Preconditie:
Het systeem bevat een schema van de virtuele verkeerssituatie. Dit schema bevat minstens 1 verkeerslicht.

Succesvol einde:
Verkeerslichten worden oranje voordat ze rood worden.

Uitbreiding:
Use case 3.2

Opmerkingen:
Om een verkeerslicht met drie kleuren te simuleren, hanteer je de volgende regels:

- Een verkeerslicht dat van groen naar rood verandert, zal eerst oranje worden.
- Een verkeerslicht dat van rood naar groen verandert, zal niet oranje worden.
- De tijd dat een verkeerslicht oranje blijft is gelijk aan 10% van zijn cyclus.
- Wanneer een verkeerslicht oranje is, dan mogen voertuigen in de stopafstand aan maximale snelheid verder rijden. Het eerste voertuig in de vertraagafstand zal beginnen vertragen.

Voorbeeld:
Een verkeerslicht met een cyclus van 20 seconden volgt het volgende patroon:

- 20 seconden groen
- 2 seconden oranje
- 20 seconden rood
- 20 seconden groen
- ...

4.1. GUI voor simulatie

Prioriteit:
NUTTIG

Doel:
Een gebruiksvriendelijke userinterface voor het uitvoeren van de simulatie.

Preconditie:
Het systeem bevat een schema van de virtuele verkeerssituatie.

Succesvol einde:
De simulatie kan beheerd worden aan de hand van een grafische userinterface.

4.2. GUI voor verkeerslichten

Prioriteit:
NUTTIG

Doel:
Een gebruiksvriendelijke userinterface hebben voor het beheren van verkeerslichten: tussen simulatiestappen kan je de staat van verkeerslichten aanpassen.

Preconditie:
Het systeem bevat een schema van de virtuele verkeerssituatie.

Succesvol einde:
Verkeerslichten kunnen manueel aangepast worden aan de hand van een grafische userinterface.

4.3. GUI voor voertuiggenerator

Prioriteit:
NUTTIG

Doel:
Een gebruiksvriendelijke userinterface hebben voor het beheren van voertuiggeneratoren: tussen simulatiestappen kan je een voertuiggenerator een voertuigen laten toevoegen.

Preconditie:
Het systeem bevat een schema van de virtuele verkeerssituatie.

Succesvol einde:
Voertuiggeneratoren kunnen manueel voertuigen toevoegen aan de hand van een grafische userinterface.

A Invoer formaat

Het invoerformaat voor de virtuele verkeerssituatie is zodanig gekozen dat nieuwe attributen en elementen makkelijk kunnen worden toegevoegd.

```
Verkeerssimulatie = { Element }
Element = "<" ElementType ">" AttributeList "</" ElementType ">"
ElementType = "VOERTUIG" | "BAAN" | "VERKEERSLICHT" | "VOERTUIGGENERATOR"
| "BUSHALTE" | "KRUISPUNT"
AttributeList = Attribute { Attribute }
Attribute = "<" AttributeType ">" AttributeValue "</" AttributeType ">"
AttributeType = "naam" | "lengte" | "baan" | "positie"
                | "cyclus" | "frequentie" | "type" | "wachttijd"
AttributeValue = Integer | String
Integer = Digit { Digit }
Digit = "0" ... "9"
String = Character { Character }
Character = "a" ... "z" | "A" ... "Z" | Digit
```

Merk op dat de attribuutlijst een relatief vrij formaat heeft wat sterk zal afhangen van het soort element dat gedefinieerd wordt. De volgende tabel toont de verschillende attributen voor elk element:

Element	Attribuut
BAAN	naam, lengte
VERKEERSLICHT	baan, positie, cyclus
VOERTUIG	baan, positie, type
VOERTUIGGENERATOR	baan, frequentie, type
BUSHALTE	baan, positie, wachttijd
BUSHALTE	baan, positie

Bovendien zal afhankelijk van het attribuuttype slechts een bepaalde attribuutwaarde toegelaten zijn:

Attribuut	Waarde
naam, baan, type	String
lengte, positie, cyclus, frequentie, wachttijd	Integer

Bovendien moet de openings tag steeds overeenkomen met de sluitingstag. Vandaar dat tijdens de invoer moet gecontroleerd worden of de invoer al dan niet geldig is.

Het bestand met de in te lezen verkeerssituatie wordt met de hand geschreven. Om de ingelezen verkeerssituatie te kunnen simuleren moet de informatie consistent zijn.

Een verkeerssituatie is consistent als:

- Elk voertuig staat op een bestaande baan.
- Elk verkeerslicht staat op een bestaande baan.
- Elke voertuiggenerator staat op een bestaande baan.
- Elke bushalte staat op een bestaande baan.
- Elk voertuig en elke voertuiggenerator heeft een geldig type.
- De positie van elk voertuig is kleiner dan de lengte van de baan.
- De positie van elk verkeerslicht is kleiner dan de lengte van de baan.
- De positie van elke bushalte is kleiner dan de lengte van de baan.
- Een verkeerslicht mag zich niet in de vertraagafstand van een ander verkeerslicht bevinden (zie Appendix B).

Opmerkingen:

- Lengte, positie, cyclus, frequentie, en wachttijd moeten altijd positief zijn.
- De naam wordt gebruikt als unieke identificatie van een baan.

B Simulatiemodel

B.1 Variabelen

Onderstaande tabel bevat een overzicht van de variabelen die gebruikt worden in het simulatiemodel.

Afkorting	Naam	Betekenis
l	lengte	Lengte van een voertuig.
x	positie	Huidige positie van een voertuig, gemeten vanaf de voorste bumper.
v	snelheid	Huidige snelheid van een voertuig.
v_{max}	maximale snelheid	De gewenste maximale snelheid van een voertuig. Kan veranderen door externe factoren (zoals een verkeerslicht).
V_{max}	maximale snelheid	De absolute maximale snelheid dat een voertuig kan rijden.
a	versnelling	Huidige versnelling van een voertuig.
a_{max}	maximale versnelling	De maximale versnelling van een voertuig.
b_{max}	maximale remfactor	De maximale remfactor van een voertuig.
f_{min}	minimale volgafstand	De minimale gewenste volgafstand van een voertuig.
Δt	simulatietijd	De tijd tussen twee stappen van de simulatie.
Δx_s	vertraagafstand	De afstand tot een verkeerslicht waarin een voertuig moet vertragen, inclusief de stopafstand.
Δx_{s0}	stopafstand	De afstand tot een verkeerslicht waarin een voertuig moet stoppen.
s	vertraagfactor	Factor waarmee een voertuig moet vertragen voor een verkeerslicht.

B.2 Formules positie en snelheid

Voor het aanpassen van de positie en snelheid van een voertuig maken we onderscheid tussen twee situaties:

- Indien $v + a\Delta t$ kleiner is dan nul, dan zou de snelheid negatief worden. Dit is niet toegestaan in ons model. In dit geval passen we de positie aan als volgt:

  $$x = x - \frac{v^2}{2a}$$

  Vervolgens zetten we de snelheid gelijk aan nul.

- Indien dit niet het geval is, dan passen we eerst de snelheid aan:

  $$v = v + a\Delta t$$

Vervolgens passen we de positie aan:

$$x = x + v\Delta t + a\frac{\Delta t^2}{2}$$

B.3 Formules versnelling

Voor het aanpassen van de versnelling van een voertuig i berekenen we eerst δ, de interactieterm met voertuig $i-1$ (het voertuig dat voor voertuig i rijdt).

We berekenen eerst de volgafstand Δx:

$$\Delta x = x_{i-1} - x_i - l_{i-1}$$

We berekenen dan het snelheidsverschil Δv:

$$\Delta v = v_i - v_{i-1}$$

Hiermee berekenen we δ:

$$\delta = \frac{f_{min} + max(0, v + \frac{v\Delta v}{2\sqrt{a_{max}b_{max}}})}{\Delta x}$$

Indien er geen voertuig voor rijdt, en dus voertuig i het eerste voertuig op de baan is, dan zetten we δ gelijk aan nul.

We kunnen dan de versnelling berekenen als volgt:

$$a = a_{max}(1 - \left(\frac{v}{v_{max}}\right)^4 - \delta^2)$$

B.4 Vertragen en versnellen

Om het eerste voertuig op de baan te laten vertragen, dan zetten we v_{max} gelijk aan sV_{max}, met s de vertragingsfactor. Voertuigen achter dit voertuig zullen automatisch ook vertragen.

Indien het eerste voertuig op de baan terug mag versnellen, dan zetten we v_{max} gelijk op V_{max}. Voertuigen achter dit voertuig zullen automatisch ook versnellen.

B.5 Stoppen

Indien het eerste voertuig op de baan tot stilstand moet komen, dan passen we elke simulatiestap de versnellig aan als volgt.

$$a = -\frac{b_{max} v}{v_{max}}$$

Voertuigen achter dit voertuig zullen automatisch ook tot stilstand komen.

C Standaard waarden

Onderstaande tabel bevat een overzicht van de algemene standaard waarden die gebruikt worden in het simulatiemodel

Algemeen		
Afkorting	Naam	Waarde
Δt	simulatietijd	0.0166
Δx_s	vertraagafstand	50
Δx_{s0}	stopafstand	15
s	vertraagfactor	0.4

Onderstaande tabel bevat een overzicht van de standaard waarden die gebruikt worden voor de verschillende soorten voertuigen: auto's en bussen.

Auto		
Afkorting	Naam	Waarde
l	lengte	4
V_{max}	maximale snelheid	16.6
a_{max}	maximale versnelling	1.44
b_{max}	maximale remfactor	4.61
f_{min}	minimale volgafstand	4
Bus		
Afkorting	Naam	Waarde
l	lengte	12
V_{max}	maximale snelheid	11.4
a_{max}	maximale versnelling	1.22
b_{max}	maximale remfactor	4.29
f_{min}	minimale volgafstand	12

Onderstaande tabel bevat een overzicht van de standaard waarden die gebruikt worden voor de verschillende soorten prioriteitsvoertuigen: brandweerwagen, ziekenwagen, en politiecombi.

Brandweerwagen		
Afkorting	Naam	Waarde
l	lengte	10
V_{max}	maximale snelheid	14.6
a_{max}	maximale versnelling	1.33
b_{max}	maximale remfactor	4.56
f_{min}	minimale volgafstand	10
Ziekenwagen		
Afkorting	Naam	Waarde
l	lengte	8
V_{max}	maximale snelheid	15.5
a_{max}	maximale versnelling	1.44
b_{max}	maximale remfactor	4.47
f_{min}	minimale volgafstand	8
Politiecombi		
Afkorting	Naam	Waarde
l	lengte	6
V_{max}	maximale snelheid	17.2
a_{max}	maximale versnelling	1.55
b_{max}	maximale remfactor	4.92
f_{min}	minimale volgafstand	6