

VERSLAG

Thematische Werkgroep 3
OSLO Verkeersmetingen

1 INHOUD

1	Inhoud.....	2
2	Praktische Info.....	3
2.1	Aanwezig.....	3
2.2	Agenda.....	3
3	Inleiding.....	4
3.1	Samenvatting vorige werkgroep.....	4
3.2	Definities.....	6
3.3	Onze aanpak.....	9
3.4	Linked Open Data.....	11
4	Overzicht model.....	12
4.1	Sneuvemodel.....	12
4.2	Storylines.....	13
4.2.1	Een meting kan verduidelijkt worden door een kwaliteitselement.....	13
4.2.2	Voor een meting kan aangegeven worden via welke procedure deze tot stand gekomen is.....	15
4.2.3	Metingen kunnen gebaseerd zijn op andere metingen.....	16
4.2.4	ADR data voorbeeld.....	17
4.2.5	Beschrijving van de meettoestellen.....	19
5	Vrijheden in het model.....	20
6	Model.....	20
7	Codelijsten.....	21
8	Volgende stappen en Q&A.....	22
9	Volgende thematische werkgroep.....	23

2 PRAKTISCHE INFO

- Datum: 02/05/2023 (9u - 12u)
- Locatie: VAC Gent (Zaal 21.04 Jacob van Artevelde)

2.1 AANWEZIGEN

Dassonneville Yaron	Digitaal Vlaanderen
Scheldeman Arne	Digitaal Vlaanderen
Soetens Steven	Provincie Antwerpen - Departement Mobiliteit, Erfgoed en Ruimte, Dienst Mobiliteit
Vlaminck David	Agentschap Wegen en Verkeer - Afdeling Planning en Coördinatie
Ghoula, Abdel Hakim	Agentschap Wegen en Verkeer
Geert Thoelen	s-Lim
Schillemans Viki	Agentschap Wegen en Verkeer
Cuypers Koen	Agentschap Wegen en Verkeer - Afdeling Elektromechanica en Telematica
Gert Brackman	Signco
Emilie Couwenberg	Mobiliteitsconsulent datacel - stad Antwerpen
Pierre Maere	Geo Mobility
Steven Logghe	Digitaal Vlaanderen
Jeroen Van Houtte	Lantis
Thijs Geert	Digitaal Vlaanderen
Stijn Vernailen	stad Antwerpen
CAPPELLE Ruben	Gewestelijke Overheidsdienst Brussel
Cécile bauvin	Icoms Detections
Windmolders Elsje	Provincie Limburg

2.2 AGENDA

09u00 - 09u10	Welkom en agenda
09u10 - 09u20	Samenvatting vorige werkgroep
09u15 - 09u25	Overzicht van aanpassingen

09u35 - 09u45	Definities klassen en attributen
09u45 - 10u45	Overzicht model adhv storylines
10u45 - 11u00	Pauze
11u00 - 11u15	LDES
11u15 - 11u45	Codelijsten
11u45 - 12u00	Q&A en volgende stappen

3 INLEIDING

3.1 SAMENVATTING VORIGE WERKGROEP

Tijdens de vorige werkgroep werd er gewerkt rond de verfijning van het model. Een aantal voorbeeld use cases werden toegelicht om de mogelijkheden van het model te illustreren. Op basis van de feedback werden een aantal kleine aanpassingen aan het model gedaan zoals een extra relatie tussen verkeersmeting en observatieprocedure en het gebruik van de originele namen voor de klassen die de toestellen beschrijven.

Graag verwijzen we hiervoor naar slide 8.

Topics vorige werkgroep?



Verfijning van het model

- Intro netwerkreferentie

Vervolg op de use cases

- Individuele voertuigmetingen, ADR codes op vrachtwagens
- Metadata van metingen en metingverzameling
- Herkomst bestemming met een afslagbeweging op een kruispunt
- Capteren noden rond meettoestellen

Aanpassingen aan het model

- Relatie 'Verkeersmeting' en 'Observatieprocedure'
- 'Meetinstallatie', 'Meetinstrument' en 'Meettechniek' aangepast naar 'Toestel', 'Sensor' en 'Observatieprocedure'

De scope van het project is het ontwikkelen van een semantisch framework waarin alle data rond verkeersmetingen in kaart wordt gebracht en gedeeld. Binnen het traject wordt een applicatieprofiel en vocabularium uitgewerkt. Dit wordt uiteindelijk ook gepubliceerd op data.vlaanderen. Binnen een OSLO traject wordt de OSLO-methodiek toegepast. Dit houdt in dat er gestart wordt vanuit bestaande use cases en deze zo veel mogelijk gealigneerd worden met bestaande standaarden, zowel op Vlaams, Europees als internationaal niveau. Indien er elementen zijn die nog niet bestaan maar wel nodig zijn, worden deze gedefinieerd.

Graag verwijzen we hiervoor naar slide 9.

Scope van het project

Ontwikkel een semantisch framework voor het in kaart brengen van verkeersmetingen en het delen van data

Ontwikkel een duurzaam **applicatieprofiel** en **vocabularium** voor Verkeersmetingen.

We volgen de OSLO methodiek, wat betekent dat:



We starten van use cases



We definiëren zelf zaken waar nodig



We aligneren zoveel mogelijk met bestaande standaarden

Tijdens eerdere werkgroepen kwam de vraag naar boven om een Engelstalige standaard te voorzien voor OSLO Verkeersmetingen. Er werd gekozen om eerst de standaard verder te ontwikkelen in het Nederlands en daarna een Engelstalige vertaling te voorzien die zal uitgestuurd worden naar de leden van de werkgroep. Om dit te faciliteren en goed te keuren, wordt ook een extra werkgroep voorzien in september. De ambitie blijft om deze standaard later ook mee te nemen in projecten op Europees niveau.

Graag verwijzen we hiervoor naar slide 10.

Vervolg van traject- Engels

- 1 Finaliseren van OSLO Verkeersmetingen in Nederlands (27/06)
- 2 Uitvoeren publiek review & Engelstalige vertaling voorzien (eind augustus)
- 3 Engelstalige vertaling delen, kans om feedback te geven (begin september)
- 4 Extra werkgroep eind september om Engelse vertaling te bespreken
- 5 Ambitie om standaard mee te nemen in projecten op Europees niveau



3.2 DEFINITIES

De definities gebruikt in deze werkgroep zullen terug te vinden zijn op zowel data.vlaanderen.be als op [GitHub](#). Op [GitHub](#) is er de mogelijkheid om feedback en/of input te geven.

Na discussie over deze definities werd besloten om de term wegverkeer aan te passen naar verkeer.

Met opmerkingen [DY1]: Er was als ik mij niet vergis ook nog de discussie rond Verkeersmeetpunt en traject (waarbij iets toegevoegd moet worden aan usageNotes) en daarnaast ook rond voertuigType (die moeten ook hier gecapteerd worden in de discussie)

Met opmerkingen [DP2R1]: Zie vraag en antwoord, is daar toegevoegd

Definities

Klasse	Definitie	Gebruik
Verkeersmeting	Het vaststellen van de waarde van een bepaald kenmerk van het wegverkeer op een bepaald tijdstip of tussen twee tijdstippen.	<i>Het gaat om schattingen van de waarde, maar op de waarde zit een zekere foutmarge. Voorbeelden zijn aantal, snelheid,... Dit is verschillend van waarden die zijn toegekend en dus geen schattingen zijn zoals naam of prijs.</i>
Wegverkeer	De verplaatsing van mensen, dieren en/of objecten, die over een weg gebeurt.	
Verkeerskenmerk	Klasse die kenmerken van instanties van wegverkeer vertegenwoordigt.	<i>In de context van verkeersmetingen is dit een kenmerk van het geobserveerdObject (of van het bemonsterdObject of een onderdeel daarvan) en in principe van alle instanties van dat type geobserveerdObject. Bv Aantal, snelheid,...</i>

Definities

Klasse	Definitie	Gebruik
Verkeersmetingverzameling	Aantal bij elkaar horende verkeersmetingen.	<i>Bv een reeks metingen die allemaal op dezelfde dag hebben plaatsgevonden of die allemaal over hetzelfde verkeerskenmerk gaan.</i>
Verkeersmeetpunt	Een meetlocatie die de vorm aanneemt van een punt.	<i>Van het Object dat men wil observeren wordt in dat geval een representatief puntvormig deel beschouwd. Voorbeeld is een (Meet)Station.</i>
Verkeersmeettraject	Een meetlocatie die de vorm aanneemt van een lijn of traject	<i>Van het Object dat men wil observeren wordt in dat geval een representatief trajectvormig deel beschouwd. Voorbeeld is een doorsnede van een rijvak.</i>

* Van alle overige klassen erven we de definitie en gebruik over van andere OSLO trajecten.

Definities

Attribuut	Definitie	Gebruik
VerkeerskenmerkType	Type van verkeerskenmerk dat men wil meten	<i>Bv aantal, snelheid, ...</i>
VoertuigType (of ResourceType?)	Type van voertuig dat men wil meten	<i>Bv auto, fiets, voetganger, ...</i>

Definities

De definities (van klassen, attributen, enumeraties en datatypes) zullen terug te vinden zijn op GitHub.

Feedback is welkom!



Feedback/input kan gegeven worden via GitHub:

[OSLO thema verkeersmetingen](#)

Via het aanmaken van **issues**



Hiervoor verwijzen we graag naar slides 11 tem 15.

Opmerkingen:

Vraag: Zijn meerdere waarden mogelijk op één tijdstip?

Antwoord: Dit is uiteraard perfect mogelijk.

Digitaal Vlaanderen ///

Vraag: Opletten dat het concept verkeersmeting niet voor ander verkeer (bv luchtverkeer) gebruikt wordt?

Antwoord: Dit is inderdaad niet afgedwongen in de definitie. Er ontstond een korte discussie in de werkgroep waarbij de conclusie was dat dit niet beperkt mag worden. De metingen zullen we toelaten voor alle types verkeer.

Vraag: Hoe wordt omgegaan met een doorsnedemeting, dit is de meest voorkomende meting (dwars over een weg)?

Antwoord: Het is mogelijk via een verkeersmeetpunt en netwerkreferentie aan te geven dat je transversaal meet. Daarbij kan je al dan niet de rijstrook definiëren die je mee. We zullen dit duidelijker maken bij de gebruiksnotities van verkeersmeetpunt.

Vraag: De netwerkreferenties maken het moeilijker om zaken op een kaart te visualiseren?

Antwoord: Dezelfde voorstellingen zouden ook mogelijk moeten zijn met wegsegmenten, voor doorsnedemetingen is dit zeker ook doenbaar maar dit is een implementatieprobleem. Iedere applicatie kan daar anders mee omgaan.

Met opmerkingen [DY3]: Aangeven dat hier een discussie rond ontstaan is en dat er een keuze werd gemaakt met de werkgroep

3.3 ONZE AANPAK

De vanuit de business werkgroep geïdentificeerde concepten werden verder onderverdeeld in drie categorieën, namelijk binnen scope, buiten scope en feature/implementation. Samen met het kernteam werd beslist om deze onderverdeling te maken, om zo een goede scoping uit te kunnen voeren en de feature / implementatie zaken mee te nemen richting de werkgroepen van de ecosysteemwerking. Hieronder is de onderverdeling van de concepten binnen deze categorieën zichtbaar.

Hiervoor verwijzen we graag naar slide 18.

Vertrekken van use cases

➤ Opdeling van use cases/concepten in verschillende categorieën:

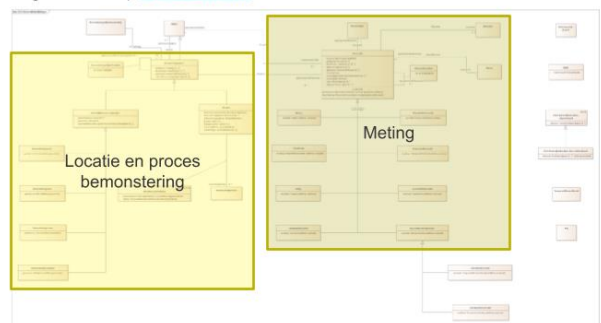
Binnen scope	Buiten scope	Feature / Implementatie
Meting (Tellingen (Fiets, Auto, ...), Snelheid, ...)	Looplijnen passanten P+R	Privacy
Rijstrook telling	Bezetting (OV)	Data aggregatie
Kruispunt telling / Herkomst -> bestemming	Verkeersoverredingen	LDES
Type vervoersmiddel	(bijna) ongevallen	Modal Shift
Metadata Meetinstrument (bv Kwaliteit meting sensor bij regenweer,...)	Floating car data	Verkeersdrukke / beleidsindicatoren
Meettechnieken (inductie, camera, glasvezel, radar, ...)	Gedetailleerde codelijsten van bv fietscategorieën?	Gebruikersovereenkomst voor data
Data publisher / Data owner	Inname openbaar domein	Live monitoring
Locatie verkeersmeting		Gebruik in alle bestuurslagen
Meetsysteem(Telraam, ANPR camera,...)		
Moment / Periode / Tijd		

Zoals aangehaald in 3.1 wordt er vertrokken van bestaande modellen. Voor OSLO Verkeersmetingen werd inspiratie gehaald uit een ander bestaand model, namelijk OSLO Observaties en metingen. Deze datastandaard bevat al heel wat aspecten rond observaties en metingen. Het applicatieprofiel van dit OSLO-model is bereikbaar via [ISO 19156:2011](#).

Voor verdere informatie verwijzen we graag naar slide 19.

OSLO Observaties en metingen

- Bevat al heel wat rond observaties / metingen
- Dit applicatieprofiel is gebaseerd op [ISO 19156:2011](#).



3.4 LINKED OPEN DATA

Linked data gaat over het publiceren en verbinden van gegevens op het web. De data verklaart zelf waar ze vandaan komt en wat ze betekent.

Alle elementen krijgen een unieke URI die mensen en machines kunnen opzoeken. De uri bevat gestructureerde info over het element.

Er worden verbindingen gelegd tussen verschillende identificatielinken om zoveel mogelijk verbanden te leggen.

Deze manier van data opslaan heeft als voordelen dat data gemakkelijker kan uitgewisseld worden, gemakkelijker kan uitgebreid worden.

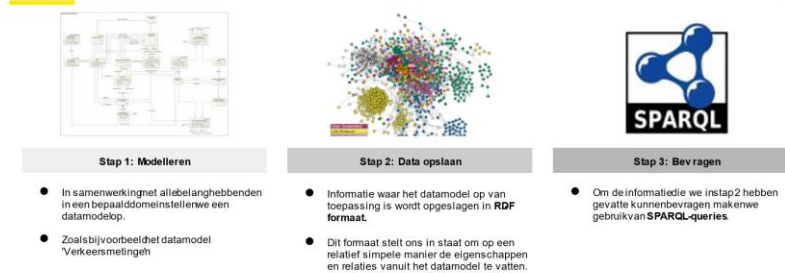
De bouwstenen van Linked Open Data

Aan de hand van de principes van Linked Open Data onderscheiden we drie basale bouwstenen: Een 'iets' met een URI, een eigenschap en een relatie.



Met opmerkingen [DY4]: Misschien moeten we hier wat visuele slides meenemen

Hoe passen we Linked Open Data toe?



Voor verdere informatie en een voorbeeld verwijzen we graag naar slides 20 - 25.

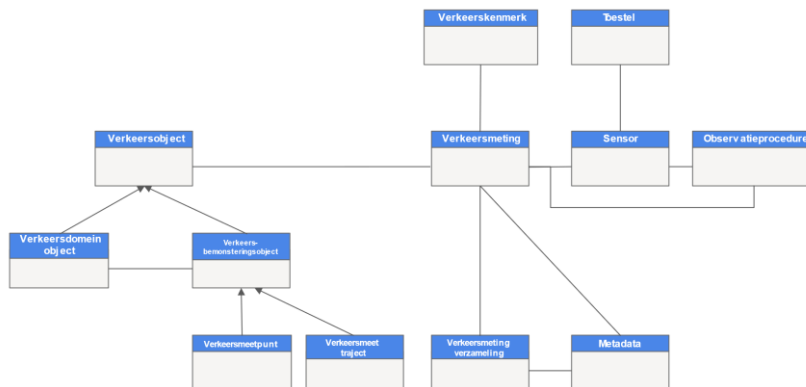
4 OVERZICHT MODEL

4.1 SNEUVELMODEL

Hieronder is het volledige model terug te vinden. Met de feedback verkregen tijdens deze werkgroep zullen verdere aanpassingen doorgevoerd worden aan het model.

Deze kan ook teruggevonden worden op slide 27.

Sneuvemodel



4.2 STORYLINES

4.2.1 Een meting kan verduidelijkt worden door een kwaliteitselement

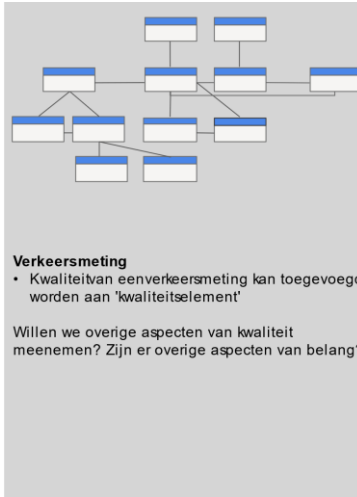
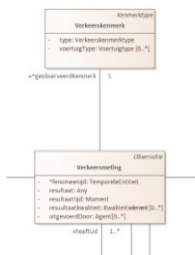
Het kwaliteitselement komt uit de [OSLO standaard Datakwaliteit](#). Dit is op ISO gebaseerd net zoals Observaties en metingen waar het model van verkeersmeting op gebaseerd is.

Er zijn verschillende types kwaliteitselementen, ontbrekende gegevens, thematische nauwkeurigheid, positionele nauwkeurigheid,...

De standaard laat toe het type kwaliteitsmeting te selecteren.

Storyline

Kwaliteitselement



Verkeersmeting

- Kwaliteit van een verkeersmeting kan toegevoegd worden aan 'kwaliteitslement'

Willen we overige aspecten van kwaliteit meenemen? Zijn er overige aspecten van belang?

Er kwamen er vanuit de leden van de werkgroep enkele opmerkingen naar boven:

- Wat hier besproken wordt is niet de essentie van het kwaliteitssysteem van Wegen en Verkeer. Er wordt een maat gegeven voor hoeveel procent van de data ontbreekt na reconstructie maar meestal kan alles gereconstrueerd worden. Het is heel belangrijk te weten op basis waarvan de reconstructie gebeurd is.

De manier waarop verkeersmetingen beschrijven gaat allemaal over kwaliteit. Wie heeft de meting gedaan en welke observatiemethode (bv interpolatie) is gebruikt. Je kan op heel veel manieren de kwaliteit van gegevens meegeven. Dit zijn allemaal aspecten van kwaliteit, dit voorbeeld is de formele manier van kwaliteit meegeven.

Vanuit de leden van de werkgroep zijn er ook een aantal vragen naar boven gekomen:

Vraag: Lijkt complex te worden door één aspect op drie plaatsen te zetten. Is verwarrend.

Antwoord: Het idee is om geen tekst te gebruiken voor kwaliteit maar effectief een duidelijk interpreteerbare waarde. Dit is inderdaad beschikbaar op meerdere plaatsen in het semantische model. In een effectieve implementatie kan dit uiteraard allemaal samen beschikbaar gemaakt worden.

Vraag: Moeilijk om kwaliteit te bepalen als de informatie zo verspreid is in het model.

Met opmerkingen [DYS]: Hier is meer over gezegd geweest waarom dit op 3 plaatsen staat. Semantisch gezien in model. + in effectieve toepassing houdt niets tegen om effectief alles bij elkaar te zetten in de functionaliteiten

Digitaal Vlaanderen ///

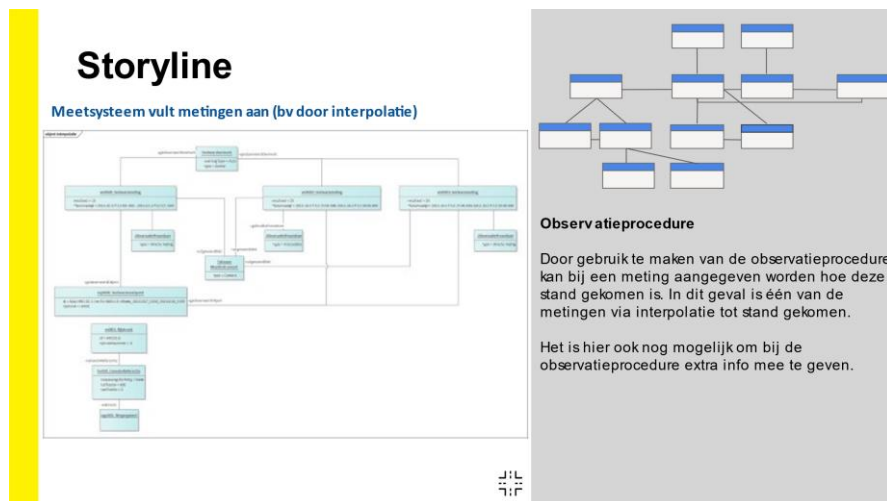
Antwoord: Hier gaat het om het datamodel waar we machine leesbaarheid nastreven en de relaties van objecten tov elkaar definiëren. Daarna komt een functioneel aspect als er een applicatie gebouwd wordt. Daar kan je dan inderdaad de gewenste data samenbrengen afhankelijk van de insteek en wensen van de gebruiker.

Hiervoor verwijzen we graag naar slides 29 tem 30.

4.2.2 Voor een meting kan aangegeven worden via welke procedure deze tot stand gekomen is

In deze storyline zijn er drie verkeersmetingen terug te vinden. Twee van deze metingen zijn directe metingen en hebben effectief plaatsgevonden. De derde meting is een meting die door interpolatie toegevoegd is.

Hoe een meting tot stand gekomen is kan aangegeven worden via de observatieprocedure. Het is daarenboven ook mogelijk om te verwijzen naar een document dat de procedure in detail beschrijft.



Vanuit de leden van de werkgroep zijn er ook een aantal vragen naar boven gekomen:

Vraag: Aangezien er meerdere links in een weglinkequentie mogelijk zijn, kan het eerste Wegsegment gelinkt worden aan herkomst en het laatste Wegsegment aan bestemming?

Digitaal Vlaanderen ///

Antwoord: Herkomst en bestemming hoeven inderdaad geen aansluitende links te zijn. Deze kunnen beide gerepresenteerd worden door respectievelijk een beginknoop en een eindknoop.

Vraag: Is er de mogelijkheid om kruispunt en herkomst/bestemming onder dezelfde logica te vatten?

Antwoord: Er kunnen meerdere herkomsten en bestemmingen zijn op eenzelfde traject. Als voorbeeld kan er een route bestaan van A naar C die ook een bestemming heeft in B, dat gelegen is tussen A en C. Zo heb je verschillende mogelijkheden om het traject te bekijken. Er kan bijvoorbeeld gekeken worden hoeveel voertuigen er in zowel A als C gepasseerd zijn. In deze data kan er dan nog een onderscheid gemaakt worden tussen de voertuigen die wel of niet door punt B geweest zijn.

Vraag: Een rotonde kan verschillen in layout en dus ook van het aantal takken. Wordt een rotonde dan beschouwd als één enkele zone of als een zone met verschillende knooppunten?

Antwoord: Een rotonde wordt beschouwd als een zone met verschillende knooppunten.

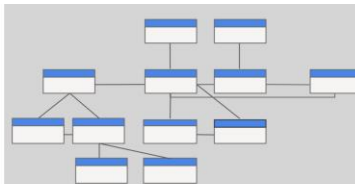
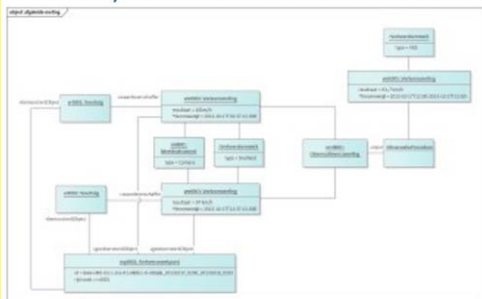
Hiervoor verwijzen we graag naar slide 31.

4.2.3 Metingen kunnen gebaseerd zijn op andere metingen

Een meting kan afgeleid worden van een observatieverzameling van andere metingen. Dus de observatieverzameling wordt gebruikt als input voor een andere meting.

Storyline

Een meetsysteem meet snelheden en berekent de V85 waarde.



Metingen

Naast de basismetingen kunnen metingen ook afgeleid zijn van andere metingen. In dit geval is de V85 meting afgeleid van de snelheidsmetingen.

Vanuit de leden van de werkgroep zijn er ook een aantal vragen naar boven gekomen:

Vraag: Kan je ook enkel de individuele metingen of de afgeleide meting weergeven?

Antwoord: Dit is geen probleem, je kan uiteraard enkel de individuele metingen doorgeven en de gebruiker zelf de V85 laten berekenen. Alle opties zijn mogelijk. Individuele meting, enkel de afgeleide meting of beide.

Vraag: Kan ik ook bij een afgeleide meting meegeven met welk toestel etc de onderliggende metingen gedaan zijn?

Antwoord: Je kan inderdaad ook alle mogelijkheden van een "gewone" meting gebruiken om de meting verder te specificeren.

Hiervoor verwijzen we graag naar slides 32 tem 33.

4.2.4 ADR data voorbeeld

Dit is een nog eens een herhaling van het voorbeeld van de vorige keer, hier tonen we hoe dit er uitziet in JSON-LD.

Storyline

ADR voorbeeld in ISO-11D

```

// 1. create an array of 10 random numbers
int arr[10];
for (int i = 0; i < 10; i++)
    arr[i] = rand() % 100;

// 2. print the array
for (int i = 0; i < 10; i++)
    cout << arr[i] << " ";

// 3. find the maximum value
int max = arr[0];
for (int i = 1; i < 10; i++)
    if (arr[i] > max)
        max = arr[i];

// 4. find the minimum value
int min = arr[0];
for (int i = 1; i < 10; i++)
    if (arr[i] < min)
        min = arr[i];

// 5. find the average value
double sum = 0;
for (int i = 0; i < 10; i++)
    sum += arr[i];
double avg = sum / 10;

// 6. find the median value
int sortedArr[10];
for (int i = 0; i < 10; i++)
    sortedArr[i] = arr[i];
for (int i = 0; i < 10; i++)
    for (int j = i + 1; j < 10; j++)
        if (sortedArr[i] > sortedArr[j])
            swap(sortedArr[i], sortedArr[j]);
int median = sortedArr[5];

// 7. find the mode value
int mode = 0;
int count = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        if (arr[i] == arr[j])
            count++;
    if (count > mode)
        mode = count;
    count = 0;

// 8. find the range value
int range = arr[9] - arr[0];

// 9. find the standard deviation value
double variance = 0;
for (int i = 0; i < 10; i++)
    variance += (arr[i] - avg) * (arr[i] - avg);
double stdDev = sqrt(variance / 10);

// 10. find the correlation coefficient value
double corrCoef = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        corrCoef += (arr[i] - avg) * (arr[j] - avg);
corrCoef /= 10;

// 11. find the skewness value
double skewness = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        for (int k = 0; k < 10; k++)
            skewness += (arr[i] - avg) * (arr[j] - avg) * (arr[k] - avg);
skewness /= 1000;

// 12. find the kurtosis value
double kurtosis = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        for (int k = 0; k < 10; k++)
            for (int l = 0; l < 10; l++)
                kurtosis += (arr[i] - avg) * (arr[j] - avg) * (arr[k] - avg) * (arr[l] - avg);
kurtosis /= 10000;

// 13. find the entropy value
double entropy = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        entropy += -log2(arr[i] * arr[j]);
entropy /= 100;

// 14. find the information gain value
double infoGain = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        infoGain += -log2(arr[i] * arr[j]);
infoGain /= 100;

// 15. find the feature importance value
double featureImportance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        featureImportance += -log2(arr[i] * arr[j]);
featureImportance /= 100;

// 16. find the model accuracy value
double modelAccuracy = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelAccuracy += -log2(arr[i] * arr[j]);
modelAccuracy /= 100;

// 17. find the model loss value
double modelLoss = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelLoss += -log2(arr[i] * arr[j]);
modelLoss /= 100;

// 18. find the model error value
double modelError = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelError += -log2(arr[i] * arr[j]);
modelError /= 100;

// 19. find the model precision value
double modelPrecision = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelPrecision += -log2(arr[i] * arr[j]);
modelPrecision /= 100;

// 20. find the model recall value
double modelRecall = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelRecall += -log2(arr[i] * arr[j]);
modelRecall /= 100;

// 21. find the model f1 score value
double modelF1Score = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelF1Score += -log2(arr[i] * arr[j]);
modelF1Score /= 100;

// 22. find the model area under the curve value
double modelAUC = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelAUC += -log2(arr[i] * arr[j]);
modelAUC /= 100;

// 23. find the model log loss value
double modelLogLoss = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelLogLoss += -log2(arr[i] * arr[j]);
modelLogLoss /= 100;

// 24. find the model brier score value
double modelBrierScore = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelBrierScore += -log2(arr[i] * arr[j]);
modelBrierScore /= 100;

// 25. find the model hamming loss value
double modelHammingLoss = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelHammingLoss += -log2(arr[i] * arr[j]);
modelHammingLoss /= 100;

// 26. find the model jaccard index value
double modelJaccardIndex = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelJaccardIndex += -log2(arr[i] * arr[j]);
modelJaccardIndex /= 100;

// 27. find the model cosine similarity value
double modelCosineSimilarity = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelCosineSimilarity += -log2(arr[i] * arr[j]);
modelCosineSimilarity /= 100;

// 28. find the model euclidean distance value
double modelEuclideanDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelEuclideanDistance += -log2(arr[i] * arr[j]);
modelEuclideanDistance /= 100;

// 29. find the model manhattan distance value
double modelManhattanDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelManhattanDistance += -log2(arr[i] * arr[j]);
modelManhattanDistance /= 100;

// 30. find the model chebyshev distance value
double modelChebyshevDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelChebyshevDistance += -log2(arr[i] * arr[j]);
modelChebyshevDistance /= 100;

// 31. find the model mahalanobis distance value
double modelMahalanobisDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelMahalanobisDistance += -log2(arr[i] * arr[j]);
modelMahalanobisDistance /= 100;

// 32. find the model hamming distance value
double modelHammingDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelHammingDistance += -log2(arr[i] * arr[j]);
modelHammingDistance /= 100;

// 33. find the model jaccard distance value
double modelJaccardDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelJaccardDistance += -log2(arr[i] * arr[j]);
modelJaccardDistance /= 100;

// 34. find the model cosine distance value
double modelCosineDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelCosineDistance += -log2(arr[i] * arr[j]);
modelCosineDistance /= 100;

// 35. find the model euclidean distance value
double modelEuclideanDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelEuclideanDistance += -log2(arr[i] * arr[j]);
modelEuclideanDistance /= 100;

// 36. find the model manhattan distance value
double modelManhattanDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelManhattanDistance += -log2(arr[i] * arr[j]);
modelManhattanDistance /= 100;

// 37. find the model chebyshev distance value
double modelChebyshevDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelChebyshevDistance += -log2(arr[i] * arr[j]);
modelChebyshevDistance /= 100;

// 38. find the model mahalanobis distance value
double modelMahalanobisDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelMahalanobisDistance += -log2(arr[i] * arr[j]);
modelMahalanobisDistance /= 100;

// 39. find the model hamming distance value
double modelHammingDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelHammingDistance += -log2(arr[i] * arr[j]);
modelHammingDistance /= 100;

// 40. find the model jaccard distance value
double modelJaccardDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelJaccardDistance += -log2(arr[i] * arr[j]);
modelJaccardDistance /= 100;

// 41. find the model cosine distance value
double modelCosineDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelCosineDistance += -log2(arr[i] * arr[j]);
modelCosineDistance /= 100;

// 42. find the model euclidean distance value
double modelEuclideanDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelEuclideanDistance += -log2(arr[i] * arr[j]);
modelEuclideanDistance /= 100;

// 43. find the model manhattan distance value
double modelManhattanDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelManhattanDistance += -log2(arr[i] * arr[j]);
modelManhattanDistance /= 100;

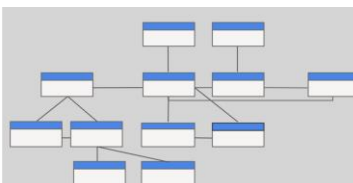
// 44. find the model chebyshev distance value
double modelChebyshevDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelChebyshevDistance += -log2(arr[i] * arr[j]);
modelChebyshevDistance /= 100;

// 45. find the model mahalanobis distance value
double modelMahalanobisDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelMahalanobisDistance += -log2(arr[i] * arr[j]);
modelMahalanobisDistance /= 100;

// 46. find the model hamming distance value
double modelHammingDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelHammingDistance += -log2(arr[i] * arr[j]);
modelHammingDistance /= 100;

// 47. find the model jaccard distance value
double modelJaccardDistance = 0;
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 10; j++)
        modelJaccardDistance += -log2(arr[i] * arr[j]);
modelJaccardDistance /= 100;

// 48. find the model cosine distance value
double modelCosineDistance = 
```



Datavoorbeeld

Dit is hoe een datavoorbeeld rond het ADR voorbeeld van vorige keer er in JSON-LD uit ziet.

Storyline

ADR voorbeeld in JSON-LD

[illegible]

Datavoorbeeld

Dit is hoe een datavoorbeeld rond het ADR voorbeeld van vorige keer er in JSON LD uit ziet.

Vanuit de leden van de werkgroep zijn er ook een aantal vragen naar boven gekomen:

Vraag: Wat doe je als het juiste segment nog niet bestaat in het wegenregister? Kan je de datum aangeven met welke datum van het wegenregister je wil werken?

Antwoord: Je kan inderdaad een versie van het wegenregister ingeven. Maar soms zal het segment ontbreken. Dan kan je zelf een segment maken in afwachting van aanmaak door bv het wegenregister. Aangezien dit op dezelfde manier gedefinieerd is kan je

er wel naar refereren. Beide zijn mogelijk. Je kan ook de geometrie van het segment zelf mee uitwisselen.

Vraag: Als je niet specifiek weet waar het gemeten is maar enkel het wegsegment weet kan dit ook?

Antwoord: Dat kan zeker, je kan zelfs zonder wegsegment. In implementatie kunnen deze mogelijkheden uiteraard beperkt worden.

Hiervoor verwijzen we graag naar slides 34 tem 37.

4.2.5 Beschrijving van de meettoestellen

Wat we gezien hebben is dat er heel weinig gegevens beschikbaar zijn. Het is goed dat dit in het model zit maar de focus in de huidige data ligt op de tellingen, niet op de toestellen of de installaties.

Alles wat er ontvangen is van voorbeelddata kan zeker in wat voorzien is in het model.

Storyline

Beschrijving van het toestel en de gebruikte procedure

```
classDiagram
    class SAMRFtoestel {
        <<Abstract>>
        +Fabricant: Langstring [0..1]
        +Knocht: Functie [0..1]
        +model: Langstring [0..1]
        +task: Task [0..1]
    }
    class SSMNOVASensor {
        +Ligging: Locatie [0..1]
        +System
        +Systeem: Systeemverrekening [0..1]
        +verrekening: Overlegingscapaciteit [0..1]
        +type: Systeemtype [0..1]
        +werkingsbereik: Werkingsbereik [0..1]
    }
    class SSMNOVAObservatiegegevens {
        +input: Input [0..1]
        +output: Output [0..1]
        +parameter: Sensoraantalwaarde [0..1]
        +specificatie: URI [0..1]
        +type: Observatiegegevens [0..1]
    }
    SAMRFtoestel "0..*" -- "1" SSMNOVASensor
    SSMNOVASensor "0..*" -- "1" SSMNOVAObservatiegegevens
```

Gebaseerd op de info die we tot nu toe hebben ontvangen lijkt dit zeker voldoende om alles te captureren.

Zijn er zaken die mogelijks toch ontbreken?

Hiervoor verwijzen we graag naar slide 38.

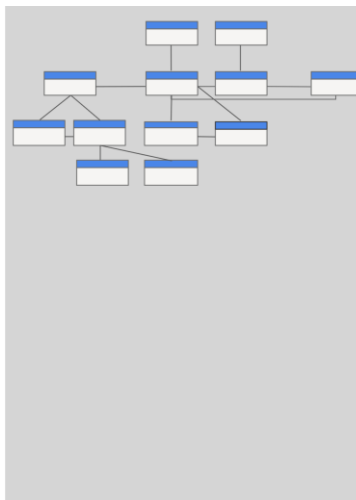
5 VRIJHEDEN IN HET MODEL

Er werd een lijst voorzien van de vrijheden die het model momenteel nog bevat. Dit zijn zaken die bij de implementatie verder moeten bekeken worden en waar er afspraken rond moeten gemaakt worden.

Vrijheden model

Een korte oplistijng van vrijheden in het model die nog afspraken behoeven voor een uniforme implementatie.

- Types verkeersmetingen
 - Intensiteit, Snelheden,...
- Verkeerskenmerk
 - Wat wil je allemaal meten?
 - Snelheid, Aantal, ...
- Voertuigtype
 - Welke voertuigclassificaties wil je meten?
 - Auto, fiets,...
- Sensortype
 - Welke types van sensor wil je gebruiken?
 - Telslang, camera,...
- Netwerkreferentie
 - Welke wegsegmenten gebruik je om naar te verwijzen?
 - OSM, Wegenregister,...
 - Afspraken rond rijrichting
 - Afspraken rond rijstroken

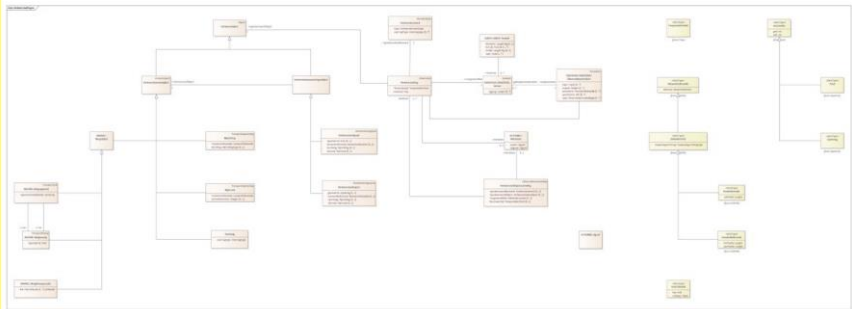


Hiervoor verwijzen we graag naar slide 39.

6 MODEL

Dit model bouwt verder op bestaande specificaties en daarin zal op de website naar verwezen worden. Alle bovenliggende links naar andere modellen zullen beschikbaar zijn. Dit is het principe van OSLO, verder bouwen op reeds ontwikkelde standaarden.

Versie 2 - Model OSLO Verkeersmetingen



Hiervoor verwijzen we graag naar slide 40.

7 CODELIJSTEN

Op basis van de ontvangen data is er gekeken op welke manieren er naar verschillende voertuigtypes gekeken wordt en hoe deze op elkaar kunnen gemapt worden.

Voertuigtype

Telraam	AWV	Mapping AWV	AWV2	Straatvinken
Voetgangers	Motoren	Licht	Personenwagens	Alle types vrachtwagens, tractoren, graafmachines, ...
Tweewielers	Personenauto's	Licht	Eénledige vrachtwagens	Bus en tram
Auto's	Bestelauto's	Licht	Personenwagens met aanhangwagen	Kleine bestelwagen en minibus
Vrachtwagens	Ongeleed vrachtwagen	Medium	Meerdere vrachtwagens	Auto, zware motor, lichtbromfiets, (elektrische) scooter
	Bussen	Medium	Andere voertuigen	Fiets, elektrische fiets/pedelec, tandem, skateboard, hoverboard, rolschaatsen, schootmobiel, ruiter op paard
	Geleed vrachtwagen	Zwaar		Voetganger, rolstoel, kindwagen

Digitaal Vlaanderen ///

Voorstel is om te starten met basis klassen (licht, medium, zwaar) maar wel ruimte laten om per meettechniek nog andere specifieke klassen toe te voegen.

Je kan putten uit meerdere codelijsten om het voertuigtype aan te duiden. Op die manier kan je inderdaad de mapping tussen de verschillende classificaties doen.

De classificaties van voertuigtypes kunnen als een taxonomie opgezet worden. Hier zijn zelfs dwarsverbanden mogelijk als iets onder twee classificaties valt. Bv een elektrische bakfiets is een bakfiets en een elektrische fiets. Er dient wel opgelet te worden dat er geen codelijsten gemengd geraken.

De categorisering van de voertuigtypes kan afhangen van het doel van de telling.

Voorlopige conclusie lijkt om te starten met: Actief (fietsers voornamelijk); Licht; Medium; Zwaar

Hiervoor verwijzen we graag naar slides 50 tem 55.

Met opmerkingen [DY6]: Deze zaken staan hier precies wat los en verloren

8 VOLGENDE STAPPEN EN Q&A

Richting de volgende thematische werkgroep zullen een aantal stappen genomen worden. Op de slide hieronder is zichtbaar wat deze stappen zijn.

Voor de volgende stappen verwijzen we ook graag naar slides 46 - 48.

Volgende stappen



Verwerken van alle input uit de thematische werkgroep.



Rondsturen van een verslag van deze werkgroep. Feedback is zeker welkom.



Feedback capteren via GitHub.



Aangepaste versie van semantisch model publiceren op GitHub en data.vlaanderen.be. Hier is feedback ook zeker welkom.

9 VOLGENDE THEMATISCHE WERKGROEP

De volgende thematische werkgroep zal plaatsvinden op **dinsdag 27 juni** van 9u00 tot 12u00. Deze zal fysiek doorgaan in VAC gent. Aansluitend vindt er een broodjeslunch plaats en wordt nadien de ecosysteem werkgroep georganiseerd van 13u tot 16u.

Voor je in te schrijven voor de volgende thematische werkgroep kan dat via de volgende [link](#).

OSLO tijdslijn

