



Object Type Library (OTL)

Een introductie

De beschikbare elementen uit de OSLO²-richtlijnen, krijgen een specifieke invulling voor het OTL.

- **Vocabularium**
 - Abstracten
 - Installatie
 - Onderdeel
 - Implementatie-element
 - Proef en meting
 - levenscyclus
- **Implementatieprofielen**
 - Thematische implementatieprofielen
 - Master Implementatieprofiel
- **OSLO-Class:**
 - Objecttypes
 - Collecties
 - Abstracten
- **OSLO-Datatype:**
 - Eenvoudig Datatype
 - Complex Datatype
 - Union Datatype
 - Kwantitatieve Waarde
- **OSLO-Enumeration:**

- Relaties
 - Keuzelijst
 - Benoemde types



Vooraf

Vragen over OTL?

Op het einde van deze sessie

Aan de posters

Algemene principes

Details SB250 en voorlopige versies SB270

Level of Geometry

Online documentatie

verwijzingen in (de notities van) de presentatie

Website en nieuwsbrief

Klik hier

Contactformulier

klik hier

Vlaanderen
is wegen en verkeer

Objecttypenbibliotheek



Welke objecten bevinden zich in de reële wereld?

AWV standaardiseert de informatie over de assets/infrastructuurobjecten in haar beheer in een ObjectTypenBibliotheek (OTL). Dit is **één gezamenlijke taal** om **informatie** te kunnen **uitwisselen**, overdragen en gebruiken.

Het doel van de OTL is het in kaart brengen, **definiëren** en **standaardiseren** van de **informatiebehoefte** met betrekking tot assets/infrastructuurobjecten aan de hand van een semantisch model dat als **open standaard** beschikbaar wordt gesteld.

Het vertrekpunt van de OTL zijn objecten in de reële wereld: een camera-installatie, een verlichtingspaal, een kast, een brug, de verschillende delen van de verticale wegopbouw, een verkeersbord... Maar ook minder tastbare en toch essentiële onderdelen voor de correcte werking van fysieke onderdelen bv. software. De OTL bevat ook ondersteunende elementen voor gegevens die betrekking hebben op die objecten uit de reële wereld zonder dat het eigenschappen van de objecten zelf zijn. De OTL bekijkt deze objecten en legt relevante eigenschappen en relaties vast.

Objecttypenbibliotheek

Semantische open data standaard



De AWV OTL is een objecttypenbibliotheek van alle weginfrastructuurobjecten, zoals beschreven in de verschillende **standaardbestekken**. Elk objecttype heeft daarin een eenduidige definitie, een aantal vastgelegde eigenschappen en mogelijke relaties met andere objecttypen.

Door de Objecttypen te **standaardiseren** worden volgende doelstellingen gerealiseerd:

- **Data** kan vlot **hergebruikt** worden door alle belanghebbenden. We willen immers allemaal werken met betrouwbare, uniforme, continu actuele en volledige data.
- AWV geeft hiermee de nodige **richting aan de sector** zodat de verwachtingen van AWV rond informatieoverdracht duidelijk zijn voor studiebureaus, aannemers of andere projectpartners. Door die verwachtingen te uniformiseren en gedetailleerd op te nemen in de omschrijving van de opdrachten worden de gevraagde inspanning van projectpartners en de noden van de Vlaamse Overheid op elkaar afgestemd. Niet alleen binnen een specifiek project, maar ook voor het latere beheer en onderhoud.

De Object Type Library (OTL) is een typenbibliotheek met:

- **Definities** over Objecten (voornamelijk fysieke dingen, AWV

assets zoals wegaanhorigheden) en bevat een **naam**

- **Eigenschappen** (kleur, materiaal, leeftijd, diameter)
- **Relaties** (hoort bij, is deel van)

- Elementen om **functies** van Objecten te vatten
- Elementen om de **levenscyclus** van Objecten te vatten

De OTL zal in de toekomst verder uitgebreid worden met elementen voor gegevens die relevant zijn of worden voor de Objecttypen die er deel van uitmaken.

Voor een specifiek project worden alleen die Objecttypen ontsloten die in dat project nodig zijn.



Implementatiemodellen

De OTL gebruikt implementatiemodellen

Een stap naar uitwisselingscontracten

Bouwt voort op het Vocabularyum

Specifiek gericht op de implementatie voor uitwisseling van data

Bv. DtcAWVDocument (Bestandsbijlage)

legt vast hoe bijlagen verwacht worden, niet enkel wat een bijlage is

Master implementatiemodel

Bevat alle OTL-elementen, enig volledig en dus **enig geldig model van de OTL**

Gedeeltelijke implementatiemodellen geven enkel de elementen en relaties relevant voor het thema mee:

Altijd onvolledig dus enkel ter illustratie van het Master implementatiemodel

Thematische modellen

Ondersteunend modellen



De OTL bestaat uit een Vocabularyum en een Master Implementatiemodel.

Het **Vocabularyum** legt eenduidig en definitief de betekenis vast van de elementen die in het Master Implementatiemodel opgenomen zijn. Elk element krijgt er zijn eigen URI die, van zodra een deel van het OTL officieel vastgelegd is, niet meer wijzigt. Waar het element gebruikt wordt, wordt verwezen naar die URI om het element in zijn vastgelegde betekenis te vatten.

Om het gebruiksgemak te verhogen, is het ene Vocabularyum opgesplitst in meerdere deel-vocabularia die de aard van het element weerspiegelen. Elk van die deel-vocabularia is volledig voor zijn deel en bevat alle elementen van het betrokken type volgens de huidige inhoud van de OTL.

Het **Master Implementatiemodel** legt de attributen vast van de Objecttypen van de OTL en de relaties tussen die Objecttypen. Een implementatiemodel is een **specificatie voor gegevensuitwisseling** voor applicaties die een bepaalde use case vervullen. Het laat toe om naast een gedeelde semantiek ook bijkomende restricties op te leggen, zoals het vastleggen van kardinaliteiten of het gebruik van bepaalde codelijsten. Waar het Vocabularyum enkel de betekenis vastlegt, legt het implementatiemodel ook het gebruik vast van de elementen i.f.v. data-uitwisseling. Een implementatiemodel dient als documentatie voor analisten, ontwikkelaars en dataleveranciers.

Alleen het Master Implementatiemodel is een volledig model met alle attributen en relaties van alle Klassen volgens de geldende stand van zaken van de OTL. Om het Master model inzichtelijk te houden voor mensen, zijn er ook gedeeltelijke weergaven van het model beschikbaar:

- Thematische implementatiemodellen zijn een weergave van inhoudelijk verwante Objecttypes binnen een bepaald deeldomein; het model bevat altijd alle attributen van de Klassen die er in opgenomen zijn maar niet noodzakelijk alle relaties tussen die Klassen (en dus ook niet de relaties met Klassen die geen deel uit maken van het model).
- Ondersteunende modellen bevatten elementen die niet noodzakelijk overeenkomen met objecten uit de reële wereld maar die wel nodig zijn voor de goede werking van de OTL en van de data-uitwisseling waarvoor de OTL opgezet is.

Thematische modellen zijn een startpunt om inzicht te verwerven in de OTL of wanneer je als gebruiker te maken hebt met een beperkt deeldomein. Data-uitwisseling moet evenwel gebaseerd worden op het Master implementatiemodel.



Implementatiemodellen

Het volledige master-implementatiemodel wordt opgesplitst in een aantal thematische deel-implementatiemodellen. Deze deelmodellen zijn een overzichtelijker manier om een bepaald deeldomein uit de master weer te geven. Ze behoren echter steeds tot het grotere geheel omdat er immers steeds raakvlakken zijn met andere deeldomeinen. Hieronder volgt een overzicht van de verschillende deel-implementatiemodellen.



Eenvoudige datatypes

Geeft een overzicht van de basis datatypes die gebruikt worden in de verschillende vocabularia.

Complexe datatypes

Geeft een overzicht van de complexe datatypes die gebruikt worden in de verschillende vocabularia.

Union datatypes

Geeft een overzicht van de union datatypes die gebruikt worden in de verschillende vocabularia.

Relaties

Geeft een technisch overzicht van de soorten relaties die tussen objecten kunnen voorkomen.

Level of geometry (LoG)

Geeft een totaal overzicht van de objecten en de gekoppelde geometrische objecttypen (GRB) met de verwachte geometrie (punt, lijn, polygoon, ...).

Master implementatiemodel

Geeft een overzicht van alle objecten die voorkomen in het implementatiemodel wegenenverkeer.

AWV-Object

Geeft een overzicht van de technische specificaties die nodig zijn voor datauitwisseling met de AWV AIM-omgeving.

ANPR - Trajectcontrole

Geeft een overzicht van de objecten die horen bij een ANPR-installatie of een Trajectcontrole.

Brandleiding

TODO

Cameralinstallatie

Geeft een overzicht van de objecten die horen bij de verschillende optellingen van camerainstallaties.

Draagconstructies

Geeft een overzicht van de objecten die als draagconstructie kunnen dienen voor

Dynamische borden

Geeft een overzicht van de objecten die horen bij de verschillende soorten installaties van

Ecologische maatregelen

Beschrijft onderdelen en installaties die horen bij ecologische maatregelen

Alleen het Master implementatiemodel bevat de volledige weergave van de OTL. Het bevat alle OTL-onderdelen en al hun mogelijke relaties. Enkel dat model mag beschouwd worden als normatief.

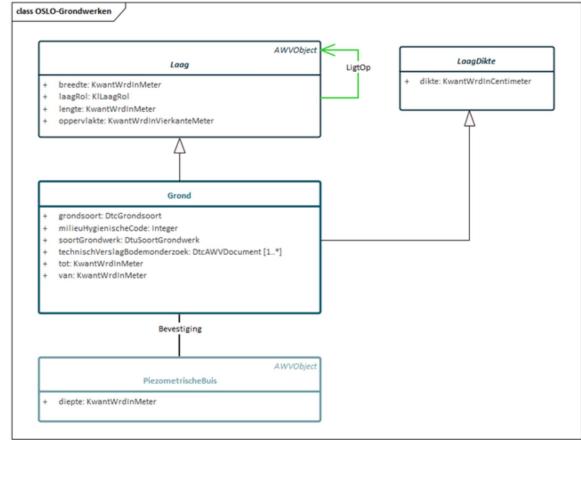
Alle andere modellen zijn deelverzamelingen van het Master implementatiemodel. Ze bevatten slechts een deel van alle OTL-onderdelen en de onderdelen die er wel in opgenomen zijn, zijn er niet noodzakelijk in opgenomen met al hun mogelijke relaties. Ze mogen daarom slechts gezien worden als illustraties van het complete Master implementatiemodel.

Afbeelding: knipsel uit <https://wegenenverkeer-test.data.vlaanderen.be/>

Model of diagram?

Een diagram maakt het model leesbaar voor een mens

```
{ "@context": { "Bevestigd": "https://wegenenverkeer-test.data.vlaanderen.be/ns/onderdeel#Bevestigd", "DtcGrondAfgravinguitgraving": "https://wegenenverkeer-test.data.vlaanderen.be/ns/onderdeel#DtcGrondAfgravinguitgraving", "DtcGrondAfgravinguitgraving.soorGrondwerk": { "@id": "https://wegenenverkeer-test.data.vlaanderen.be/ns/onderdeel#DtcGrondAfgravinguitgraving.soorGrondwerk", "@type": "http://www.w3.org/2004/02/skos/core#Concept" }, "DtcGrondafdekking": "https://wegenenverkeer-test.data.vlaanderen.be/ns/onderdeel#DtcGrondafdekking", "DtcGrondafdekking.herkomst": { "@id": "https://wegenenverkeer-test.data.vlaanderen.be/ns/onderdeel#DtcGrondafdekking.herkomst", "@type": "http://www.w3.org/2004/02/skos/core#Concept" }, "DtcGrondafdekking.soorGrondwerk": { "@id": "https://wegenenverkeer-test.data.vlaanderen.be/ns/onderdeel#DtcGrondafdekking.soorGrondwerk", "@type": "http://www.w3.org/2004/02/skos/core#Concept" }, "DtcGrondbijmenging": "https://wegenenverkeer-test.data.vlaanderen.be/ns/onderdeel#DtcGrondbijmenging", "DtcGrondophoging": "https://wegenenverkeer-test.data.vlaanderen.be/ns/onderdeel#DtcGrondophoging", "DtcGrondophoging.herkomst": { "@id": "https://wegenenverkeer-test.data.vlaanderen.be/ns/onderdeel#DtcGrondophoging.herkomst", "@type": "http://www.w3.org/2004/02/skos/core#Concept" }, "DtcGrondophoging.soorGrondwerk": { }
```



Het model is de combinatie van de OTL-Klassen zoals vastgelegd in het Vocabulary, hun attributen en hun mogelijke relaties. Er is één OTL model.

Het ene OTL-model wordt op verschillende manieren aangeboden:

- In meerdere **html-pagina's** voor het Vocabulary en de implementatiemodellen met de volledige documentatie.
- Als een visualisatie in de vorm van een **UML Klassen-diagram** als deel van de html-documentatie. Deze diagrammen bestaan enkel voor de thematische en (enkele) ondersteunende delen van het OTL, niet voor het Master Implementatiemodel
- Als **xmi** voor import in software voor het modelleren van domeinen en data.
- Als **SQLite** met de volledige OTL in de vorm van een relationele databank. Dit artefact bevat een gecoördineerde versie van de Onderdelen met linken naar afhankelijke elementen zoals datatypes en keuzelijsten en is daarom het beste vertrekpunt voor een volledig zicht om de vereiste data.
- Als **JSON-LD** i.f.v. data-uitwisseling.
- Als **SHACL** met een onvolledige weergave van de OTL.

Een diagram is een visuele weergave van het model die dat model inzichtelijk moet maken voor mensen. Het Master Implementatiemodel kan geen visuele weergave krijgen omdat de complexiteit van die weergave zijn doel zou voorbijschieten. Geen weergave kan het volledige model inzichtelijk maken. Daarom zijn de diagrammen van de thematische implementatiemodellen een goede start om het OTL te leren. Het

SQLite artefact is de beste bron om met zekerheid alle relaties en constraints van een Objecttype te kennen volgens het Master Implementatiemodel.

Afbeelding rechts: knipsel uit artefact van thematisch implementatie model *Grondwerken*

<https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/grondwerken/ontwerpdocument/v28/context/grondwerk-en.jsonld>

Afbeelding rechts: knipsel uit thematisch implementatie model *Grondwerken* <https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/grondwerken/>

Vindplaats vocabularia en implementatiemodellen



Test [Link](#)

Voorlopige versies

Geen garantie dat de versies in Test ongewijzigd overgenomen worden

Productie [Link](#)

Enig geldende versie

Updates in officiële releases conform OSLO-richtlijnen

Momenteel SB250 in publieke review



De vocabularia en implementatiemodellen zijn terug te vinden in **twee publicatieomgevingen**:

1. Test (<https://wegenenverkeer-test.data.vlaanderen.be>) voor voorlopige versies; de elementen in deze omgevingen geven een mogelijke richting maar kunnen nog altijd wijzigen;
2. Productie (<https://wegenenverkeer.data.vlaanderen.be>) voor definitieve versies; de elementen in deze omgevingen kunnen niet wijzigen zonder de publicatie van een nieuwe release volgens de governance-richtlijnen van OSLO

Meer details over het release management proces volgen later. Voor elk project dat gebruik maakt van de OTL zal wel meegegeven worden met welke termijnen voor updates van bestaande versies van het OTL rekening moet gehouden worden.

Implementatiemodellen van thema's buiten het standaardbestek 250 zijn op vandaag, 13 november 2019, met de nodige omzichtigheid te bekijken. Sommige ervan zijn al verder uitgewerkt dan andere. Ze identificeren met relatief grote zekerheid de objecten uit de reële wereld die we willen vatten maar de attributen en relaties van de overeenkomstige Klassen staan nog niet op punt.



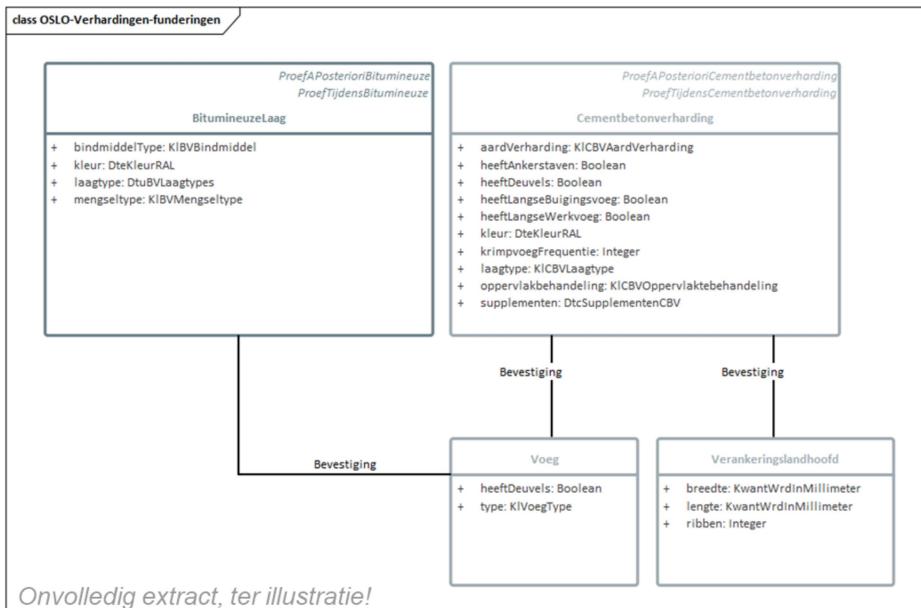
OTL elementen

De beschikbare elementen uit de OSLO-richtlijnen, krijgen een specifieke invulling voor het OTL.

- Vocabulary
 - Abstracten
 - Installatie
 - Onderdeel
 - Implementatie-element
 - Proef en meting
 - Levenscyclus
 - ...
- OSLO-Class:
 - Onderdeel
 - Installatie
 - Abstracten
- OSLO-Datatype:
 - Eenvoudig Datatype
 - Complex Datatype
 - Union Datatype
 - Kwantitatieve Waarde
- OSLO-Enumeration:
 - Keuzelijst
- Relaties

- Benoemde types, directioneel of met onbestemd richting

Onderdeel



Een Onderdeel is de **representatie van een (hoofdzakelijk) fysiek object**. Het is als atomair bestanddeel de bouwsteen van het OTL. Een object uit de reële wereld vult als een instantie van een Onderdeel de gegevens voor data-uitwisseling in. Van zodra er voor dat object een instantie aangemaakt is, zal alle verdere data-uitwisseling gebeuren d.m.v. dezelfde instantie.

De OTL maakt gebruik van **UML** (Unified Modeling Language) en principes van object-georiënteerde analyse en design (OOAD). Onderdelen worden voorgesteld als een **UML-Class** en gedragen zich ook als Classes in OOAD. Elke instantie van een Klasse bevat de data van één concreet object in de reële wereld (of van een ondersteunend concept).

De **naam van de Klasse is altijd uniek binnen het OTL**. Hij benoemt het Onderdeel op een herkenbare wijze. De mensleesbare versie van die naam is opgenomen in de tag *label-nl*. Het is deze mensleesbare vorm die terug te vinden is het overeenkomstig vocabularium en in de implementatiemodellen waarin het Onderdeel opgenomen is. Om aan te geven dat een verzameling gegevens betrekking geeft op een instantie van een Onderdeel, moet die verzameling verwijzen naar de URI van het Onderdeel uit het vocabularium *Onderdeel* (<https://wegenenverkeer-test.data.vlaanderen.be/ns/onderdeel/>).

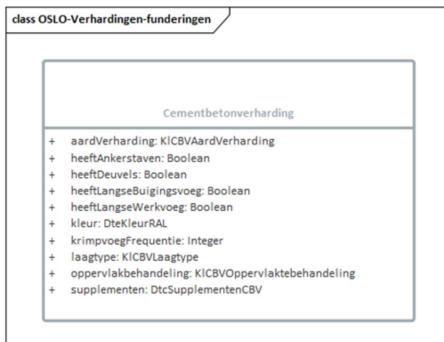
Het vocabularium bevat de **definitie** die de betekenis van het Onderdeel vastlegt. De definitie kan aangevuld zijn met een bijkomende toelichting (usagenote-nl) krijgen over het gebruik maar die de definitie niet kan wijzigen.

Onderdelen hebben **Attributen** (eigenschappen) en hebben **getypeerde Relaties** met andere Onderdelen en Installaties.

Het OTL bevat **geen hiërarchie en is geen taxonomie**: het is een platte verzameling van bouwstenen die allemaal op hetzelfde niveau staan.

Afbeelding: knipsel uit thematisch implementatie model *Verhardingen en funderingen*
<https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/verhardingen-funderingen/>

Onderdeel



Onvolledig extract, ter illustratie!

Vocabularium Onderdelen

Klasse Cementbetonverharding

Type	Klasse
URI	https://wegenenverkeer-test.data.vlaanderen.be/ns/onderdeel#Cementbetonverharding
Specialisatie van	https://wegenenverkeer-test.data.vlaanderen.be/ns/abstract#LaagBouwklasse
Definitie	Stijve verharding met of zonder wapening verkregen door cementbeton te spreiden en mechanisch te verdichten.

Implementatiemodel Verhardingen en funderingen

Cementbetonverharding

Beschrijving

Stijve verharding met of zonder wapening verkregen door cementbeton te spreiden en mechanisch te verdichten.

Subklasse van

Laag_bouwklasse

Eigenschappen

Voor deze entiteit zijn de volgende eigenschappen gedefinieerd: [aard verharding](#), [bevestigd](#), [bevestigd](#), [heeft ankerstaven](#), [heeft deuvels](#), [heeft langse buigingsvoeg](#), [heeft langse werkoog](#), [kleur](#), [krimvoeg frequentie](#), [laagtype](#), [oppervlakbehandeling](#), [Supplementen van de verharding](#).

Eigenschap	Verwacht Type	Kardinaliteit	Beschrijving	Gebruik	Codelijst
aard_verharding	CBV_aard_verharding	1	De aard van de betonverharding.	Link	

Zoals elk OTL-element krijgt het onderdeel een **eigen en permanente URI**. De vastgelegde betekenis is terug te vinden in het Vocabularium *Onderdeel* en in elk implementatiemodel waarin het Onderdeel gebruikt wordt.

De URI van een element in een Vocabularium is opgebouwd als domein + 'ns' + naam vocabularium + #naam element, telkens gescheiden door '/' met alles in kleine letters en

- Domein = de omgeving waarbinnen het element gepubliceerd is (<https://wegenenverkeer-test.data.vlaanderen.be> of <https://wegenenverkeer.data.vlaanderen.be>)
- Naam vocabularium voor Onderdelen = onderdeel
- Naam element = de naam van de betrokken Klasse zoals die ook getoond wordt op de diagrammen van de implementatiemodellen (niet label-nl want dit bevat spaties)

De URI is opgebouwd volgens de richtlijnen van de Vlaamse URI-standaard (https://data.vlaanderen.be/cms/VlaamseURI-StandaardVoorData_V1.0.pdf) en kan altijd gebruikt worden als URL om de definitie van het Onderdeel te achterhalen.

Elk implementatiemodel krijgt ook een eigen publicatie ter documentatie. Elke Klasse in het model wordt daarin beschreven met zijn al attributen en met de relaties die binnen het model opgenomen zijn. Alleen het Master Implementatiemodel bevat alle relaties van elke Klasse.

Afbeelding: knipsel uit thematisch implementatie model *Verhardingen en funderingen*
<https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/verhardingen-funderingen/>

Afbeelding rechtsboven: knipsel uit *Vocabularium Onderdelen* <https://wegenenverkeer-test.data.vlaanderen.be/ns/onderdeel/#Cementbetonverharding>

Afbeelding rechtsonder: knipsel uit thematisch implementatiemodel *Verhardingen en funderingen* <https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/verhardingen-funderingen/#Cementbetonverharding>

Attributen



Onvolledig extract, ter illustratie!

Implementatiemodellen Verkeersregelinstallatie en Draagconstructies

Seinbrug

Beschrijving

Metalen constructie bestaande uit 2 of meer verticale steunen met voetplaat en uit een enkele of een dubbel uitgevoerde horizontale dwarsverbinding, allen kokerformig met rechthoekige doorsnede. Ook wel portiek of portaal genoemd

...

Eigenschap	Verwacht Type	Kardinaliteit	Beschrijving	Gebruik	Codelijst
aantal_ladders	https://wegenenverkeer-test.data.vlaanderen.be/ns/onderdeel#Seinbrug.aantalLadders		seinbrug toegankelijk is.		
aantal_steenen	Decimal	1	Het aantal steenen waarmee de seinbrug oefrazen wordt		

Vocabularium Onderdelen

Eigenschap aantal ladders

Type	Eigenschap
URI	https://wegenenverkeer-test.data.vlaanderen.be/ns/onderdeel#Seinbrug.aantalLadders
Domein	https://wegenenverkeer-test.data.vlaanderen.be/ns/onderdeel#Seinbrug
Bereik	http://www.w3.org/2001/XMLSchema#decimal
Definitie	Het aantal ladders waarmee de seinbrug toegankelijk is.

Aan elke Klasse zijn een aantal attributen gekoppeld die het **objecttype verder specificeren**. Dit kan bv. gaan om de kleur van een paal, het merk en type van een camera, het asfaltmengsel dat gebruikt is bij een verharding, de hoogte van een seinbrug, het schakelplan van een laagspanningsbord,... Attributen zijn eigenschappen van een Klasse die **relevant zijn om data over te verzamelen**. Het criterium om een attribuut toe te voegen is niet volledigheid maar relevantie in functie van inventarisatie voor de verschillende fasen die een project doorloopt en voor gebruik nadien.

Attributen maken deel uit van het **Vocabularium van hun Klasse**. **Elk attribuut is uniek voor de Klasse** (in dit geval het Onderdeel) waartoe het behoort en het wordt volledig bepaald door die Klasse: Bovenbouw.hoogte <> Gebouw.hoogte. Dit geldt ook wanneer de attributen hetzelfde Datatype gebruiken en eenzelfde of gelijkaardige definitie hebben! Dit komt ook tot uiting in de opbouw van de permanente URI van het attribuut: de Klasse waartoe het attribuut behoort, maakt deel uit van de URI. Elke wijziging van een attribuut is ook altijd een wijziging van de Klasse waartoe het behoort maar enkel van die Klasse, nooit van andere Klassen.

De mensleesbare versie van de naam van het attribuut zit in de tag label-nl.

Elk attribuut heeft een **eigen definitie** die opgenomen is in het betreffende vocabularium. De definitie maakt duidelijk waar dit attribuut over gaat. Bv. is de 'hoogte van een seinbrug' de vrije hoogte onder de seinbrug, het hoogste punt van de seinbrug of de maaiveldhoogte van de locatie waar de seinbrug geplaatst is?

Een attribuut heeft een **datatype** dat op zijn beurt ook vastgelegd is in het vocabularium of het kan gebruik maken van een Keuzelijst van het OTL.

Een attribuut heeft een **kardinaliteit**. Wanneer deze niet expliciet vermeld is, is de cardinaliteit 1. Alle attributen zijn in principe verplicht. De tijdigheid van de aanlevering van data voor elk attribuut gebeurt volgens de vereisten van het bestek en het BIM-uitvoeringsplan.

Een attribuut kan ook bijkomende beperkingen krijgen (**constraints**) volgens een implementatiemodel.

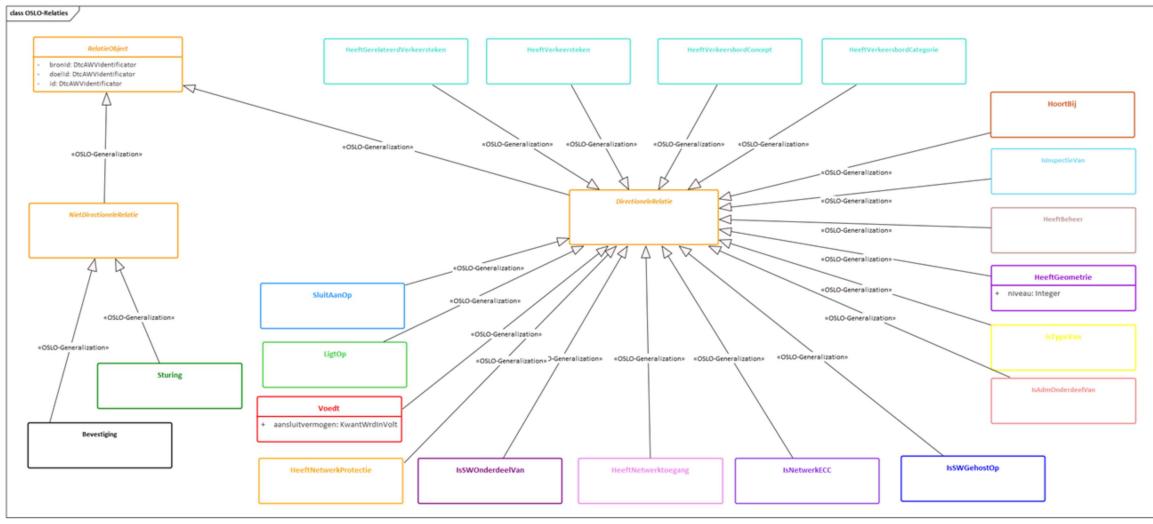
Afbeelding: knipsel uit thematisch implementatie model *Verkeersregelinstallatie*
<https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/verkeersregelinstallatie/>

Afbeelding rechtsboven: knipsel uit thematisch implementatiemodel *Verkeersregelinstallaties*:
<https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/verkeersregelinstallatie/>

Seinbrug heeft als bron het thematisch implementatiemodel van *Draagconstructies*:
<https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/draagconstructies/#Seinbrug>

Afbeelding rechtsonder: knipsel uit *Vocabularium Onderdelen*, attribuut van *Seinbrug*:
<https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Seinbrug.aantalLadders>

Relaties



Een object staat zelden volledig alleen in de fysieke realiteit. Het bestaat net in relatie met andere objecten. Bv een camera is bevestigd aan een paal, een aftakking voedt een verkeersregelaar. De relatie tussen onderdelen uit de echte wereld, krijgt in het OTL een vertaling in **getypeerde relaties**.

Elke relatie heeft ook zijn eigen **definitie** in het **Vocabularium Onderdeel**. De betekenis ligt vast en een relatie moet altijd in diezelfde zin gebruikt worden.

In de diagrammen bij de implementatiemodellen worden relaties altijd getekend als UML-associaties. In de software waarmee de diagrammen beheerd worden zoals Enterprise Architect, gaat de koppeling tussen de associatie en de Relatie-Klassen verloren. Enkel de naamgeving van de associatie zorgt nog voor de koppeling met de Relatie uit het vocabularium.

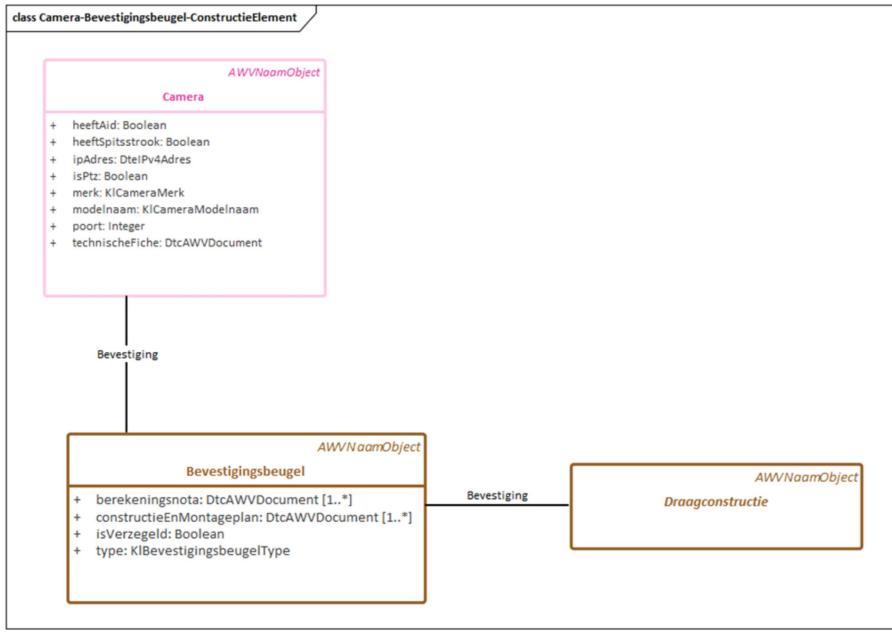
De relaties tussen Objecttypes drukken altijd **mogelijke relaties** uit, nooit verplichte. Wanneer relaties uit de OTL geen equivalent hebben in een concreet geval in de reële wereld, dan wordt ook geen data aangeleverd voor een relatie die er niet is. De relaties uit de OTL zijn echter ook de **enige toegelaten relaties**: er kan geen data uitgewisseld worden over een relatie tussen twee objecten uit de reële wereld indien die relatie niet bestaat in de OTL.

Er zijn twee types van relaties: relaties met een **uitdrukkelijke richting** en relaties

met een **onbestemde richting**. Dit onderscheid is in diagrammen gevisualiseerd door relaties met of zonder pijl aan het uiteinde.

Afbeelding: ondersteunend Implementatiemodel *Relaties* <https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/relaties/>

Relaties met een onbestemde richting



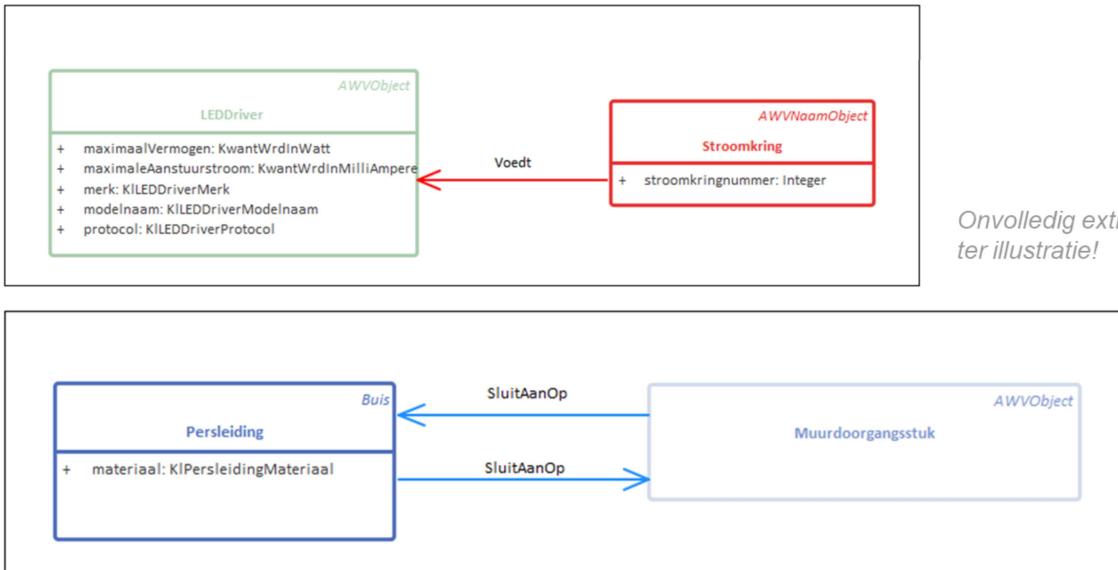
Onvolledig extract,
ter illustratie!

Relaties met een onbestemde richting drukken uit dat beide uiteinden van de relatie zich op dezelfde manier tot elkaar verhouden of dat hoe zich zich tot elkaar verhouden niet van belang is.

Bv. tussen een *Camera* en een *Bevestigingsbeugel* zit een *Bevestiging*-relatie zonder richting omdat moeilijk uit te maken valt of de *Camera* nu bevestigd is aan de *Beugel* of omgekeerd. Anderzijds kan de *Bevestigingsbeugel* ook bevestigd zijn aan een *Draagconstructie*. En hoewel dat laatste element hier duidelijk de beugel “draagt”, wijst de *Bevestiging*-relatie er op dat het onderscheid tussen beide uiteinden voor het OTL niet relevant geacht wordt.

Afbeelding: knipsel uit thematisch implementatie model Camerainstallatie <https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/camerainstallatie/>

Relaties met een uitdrukkelijke richting



In **relaties met een uitdrukkelijke richting** daarentegen, heeft de richting van de relatie wél een **betekenis die moet gerespecteerd worden**. De definitie van de relaties geeft ook aan wat de betekenis van de richting is.

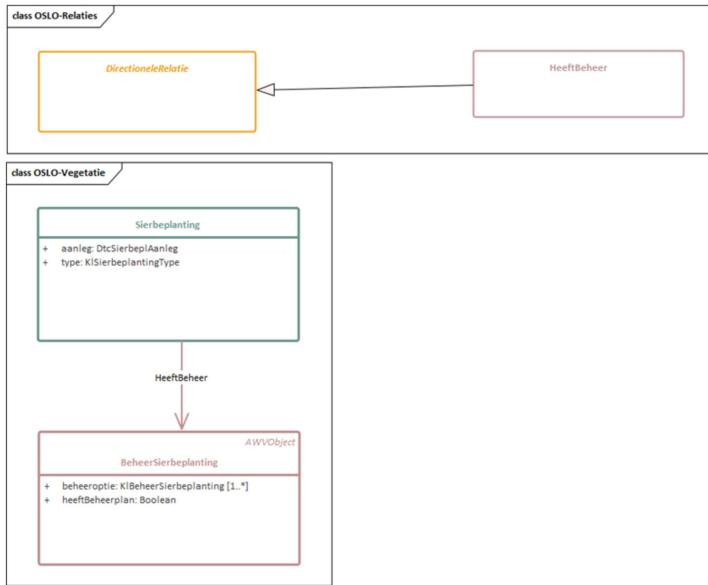
De *Voedt*-relatie bv. geeft aan welk Objecttype een ander Objecttype voedt. In het voorbeeld voedt de *Stroomkring* de *LEDDriver*. Een omgekeerde relatie is hier uitgesloten.

De *SluitAanOp*-relatie drukt oa. de horizontale richting uit waarin water stroomt, in het voorbeeld tussen een *Persleiding* en een *Muurdoorgangsstuk*. Hier moeten er twee *SluitAanOp*-relaties zijn met elk hun richting omdat het *Muurdoorgangsstuk* kan aansluiten op de *Persleiding* en omgekeerd. Dat het ene element het vertrekpunt is en het andere het eindpunt is hier wel van belang. De opeenvolging van aansluitingen zal bv. toelaten om simulaties uit te voeren.

Afbeelding onder: knipsel uit thematisch implementatie model *Verlichtingsinstallatie*
<https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/verlichtingsinstallatie/>

Afbeelding boven: knipsel uit thematisch implementatie model *Camerainstallatie* <https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/camerainstallatie/>

Ondersteunende relaties



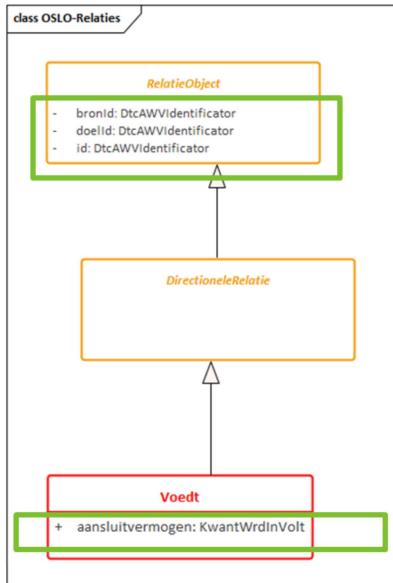
Onvolledig extract, ter illustratie!

Relaties drukken hoofdzakelijk verbanden uit tussen de fysieke onderdelen in de echte wereld en zijn van belang voor inventarisatie. Maar de relaties worden ook gebruikt om OTL-elementen die geen inventaris-objecten representeren te koppelen aan een inventaris-object uit de echte wereld, bv. om een beheer-acties te koppelen aan het object.

Afbeelding boven: knipsel uit ondersteunend Implementatiemodel *Relaties* <https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/relaties/>

Afbeelding boven: knipsel uit thematisch implementatie model *Vegetatie* <https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/vegetatie/>

Attributen van een relatie



Onvolledig extract, ter illustratie!

Een relatie kan zelf ook attributen hebben. Een voorbeeld daarvan is de **Voedt**-relatie die een attribuut heeft waarmee het aansluitvermogen van de voeding meegegeven wordt.

Om dit mogelijk te maken, worden **relaties als aparte entiteiten meegeleverd bij data-uitwisseling**. Die entiteit verwijst naar de instanties van de Objecttypen waartussen de relatie ligt. Omdat de relatie ook een eigen id krijgt, kan ze ook wijzigen zonder de Klassen die er bron en doel van zijn te wijzigen.

Om de diagrammen inzichtelijk te houden, worden relaties met attributen niet als een UML-AssociationClass getoond maar als eenvoudige associatie (al dan niet met een richting). Meer nog dan voor andere relatie-klassen is hier de link tussen de Relatie-Klasse en de associatie die op het diagram getekend is, doorgeknipt. Alleen door er de Relatie-Klassen bij te nemen, kan je in een diagram weten of de getoonde associaties, al dan niet eigen attributen heeft waarover gegevens kunnen/moeten uitgewisseld worden.

Afbeelding: knipsel uit ondersteunend Implementatiemodel *Relaties* <https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/relaties/>

Abstracten en overerving

Een modelleringstechnische ingreep

Concept

overgenomen uit OOAD/UML

Definitie

Een element dat op zich niet bestaat en waarover nooit data uitgewisseld wordt

Enkel gebruikt als superklasse om

Gedeelde attributen eenmalig te modelleren

Relaties met andere elementen eenmalig te modelleren

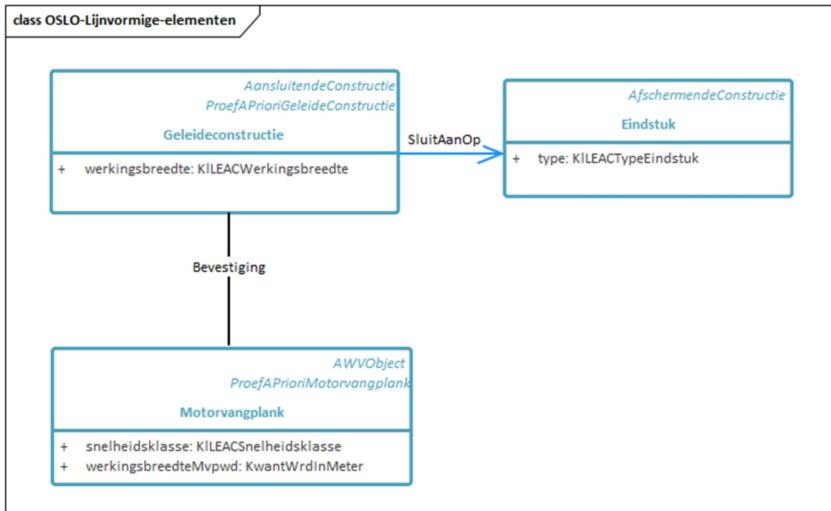
De **Abstracte** is een concept dat overgenomen is uit OOAD/UML: het gaat om een Klasse die nooit instanties kan krijgen en waarvoor dus **als zodanig geen data uitgewisseld** wordt. De OTL gebruikt Abstracten om via **overerving**, nog een OOAD/UML concept, **attributen en relaties** te kunnen toewijzen aan meerdere Objecttypes zonder deze voor elk Objecttype expliciet te moeten herhalen. Het is een modelleringstechnische ingreep om complexiteit te reduceren.

Een overerving wordt in de OTL aangeduid door een relatie van het type **Generalisatie**. Op basis van de generalisaties kan bepaald worden wat de *superklassen* zijn van een Klasse en kan dus ook de afdaling gedaan worden. Alle attributen en relaties van een superklasse zijn integraal van toepassing op een *subklasse*. Omgekeerd is dit niet het geval: superklassen erven geen attributen maar vooral geen relaties over van hun subklassen en ze kunnen deze dus ook niet doorgeven aan hun andere subklassen.

Instantieerbare klassen (klassen waarover data uitgewisseld wordt) die bij Objecttypes horen zijn alle klassen in de OTL die niet van het type “abstracte” zijn. Een abstracte klasse kan herkend worden doordat ze zich steeds bevindt in het **vocabularium abstracten** (<https://wegenenverkeer.data.vlaanderen.be/ns/abstracten/>).

In de implementatiemodellen van de OTL wordt gebruik gemaakt van **overerving** tussen verschillende klassen. Deze overerving blijft zichtbaar in de implementatiemodellen maar moet opgelost worden om tot de juiste, volledige objecttypes te komen. Voor een **volledig en correct beeld** op een implementatiemodel moet gekeken worden naar de gecoördineerde, samengevoegde klassen. Het zijn enkel deze die instantieerbaar zijn en waarvoor data aangeleverd kan worden.

Abstracten: illustratie



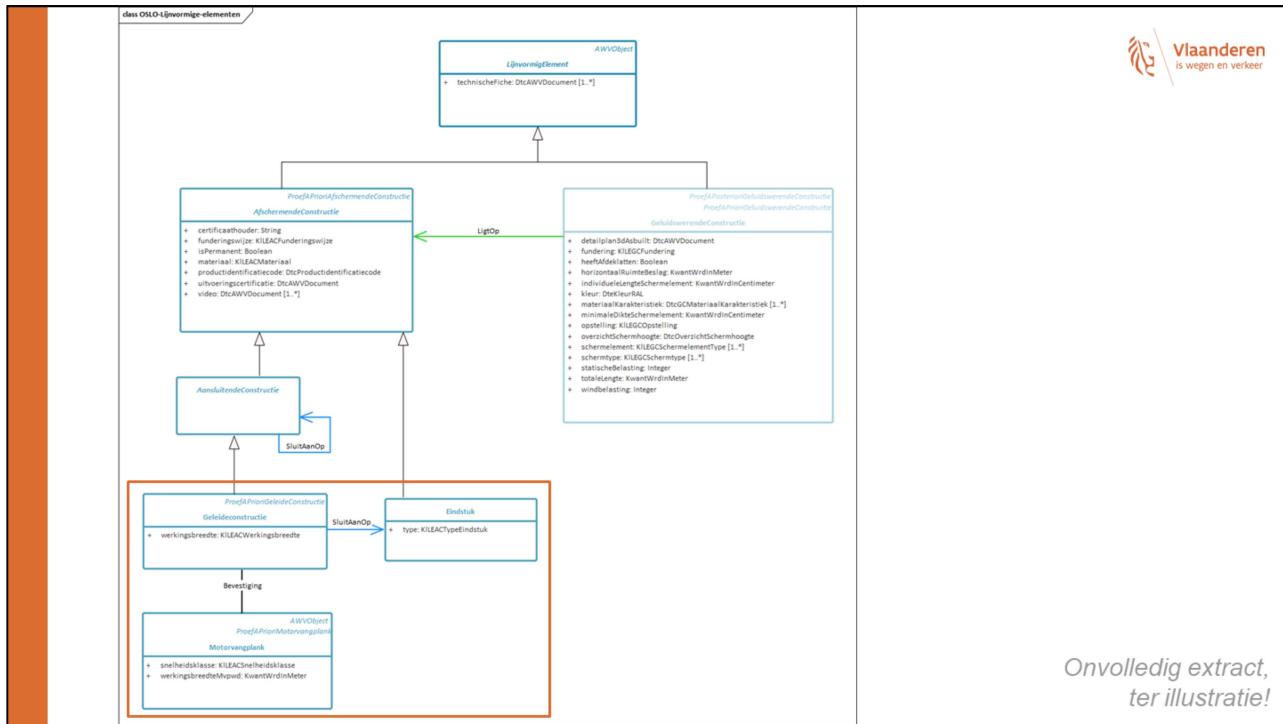
Onvolledig extract,
ter illustratie!

Een voorbeeld ter illustratie van het principe van overerving en het gebruik van abstracten binnen het OTL: de *Geleideconstructie* en de *Motorvangplank*.

Beide zijn Objecttypen binnen de OTL die overeenkomen met fysieke objecten waarover AWV data wil verzamelen en uitwisselen. Beide Objecttypes zijn als UML-Klassen opgenomen in het thematisch implementatiemodel *Lijnvormige elementen* (<https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/lijnformige-elementen/>). Een geleideconstructie en motorvangplank zijn aan elkaar bevestigd, in de OTL gerepresenteerd d.m.v. de *Beverstiging*-relatie (relatie met onbestemde richting) tussen beide Objecttypen. Het Objecttype *Geleideconstructie* kan volgens de OTL ook aansluiten op het Objecttype *Eindstuk*. De omgekeerde relatie bestaat niet.

Al bij al toch een mager beestje? Zeker als je gaat kijken naar welke attributen gevraagd worden?

Afbeelding: knipsel uit thematisch implementatie model *Lijnvormige elementen* <https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/lijnformige-elementen/>



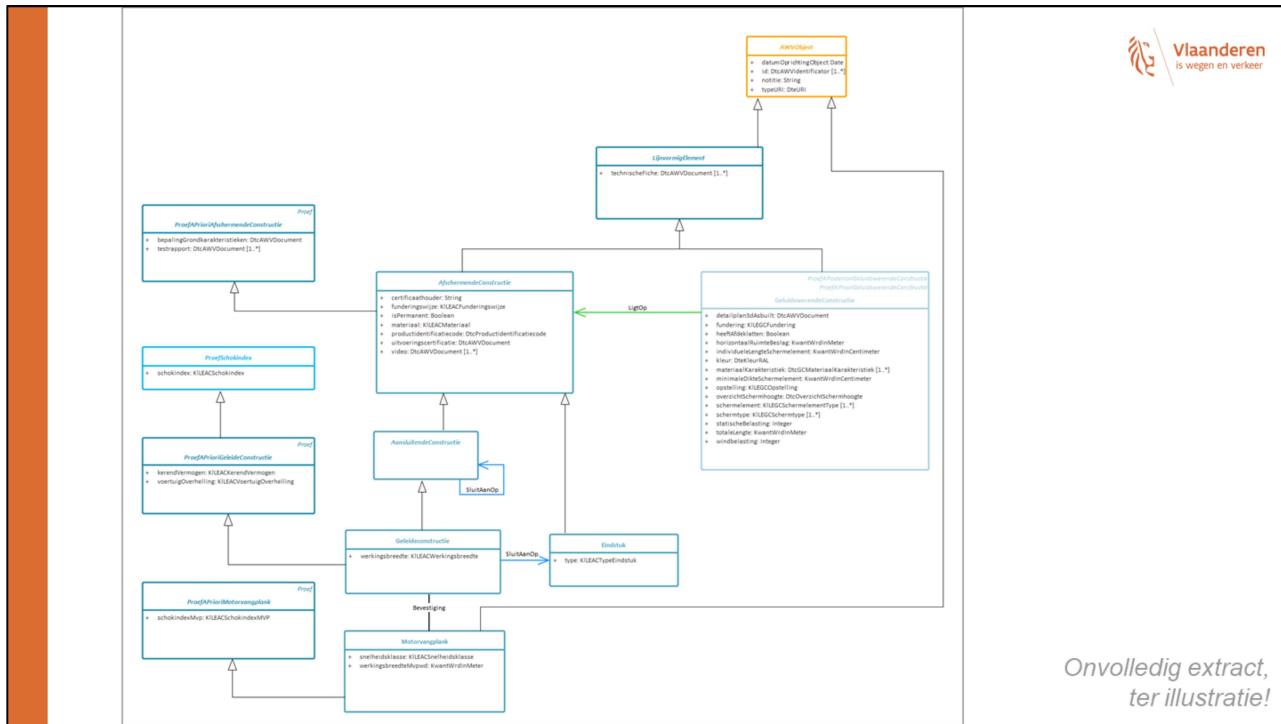
Onvolledig extract,
ter illustratie!

Het implementatiemodel *Lijnvormige element* plaatst *Geleideconstructie* en *Motorvangplank* echter in een breder plaatje waarin Abstracten en overerving een belangrijke rol spelen.

De generalisatie-relaties die expliciet getoond worden als UML-Generalisation maken al duidelijk dat de *Geleideconstructie* een subklasse is van de Abstracte *AansluitendeConstructie*. De *SluitAanOp*-relatie van *AansluitendeConstructie* wijst er op dat een instantie van het Objecttype (een concreet stuk *AansluitendeConstructie*) met een bepaalde richting kan aansluiten op een andere instantie van hetzelfde Objecttype. Als subklasse erft *GeleideConstructie* deze relatie over van zijn superklasse. Zo kan de ene instantie van *GeleideConstructie* aansluiten op de andere.

AansluitendeConstructie is op haar beurt een subklasse van *AfschermendeConstructie*. *AfschermendeConstructie* heeft verschillende attributen en is het doel van een *LigtOp*-relatie die vertrekt *GeluidsverendeConstructie*. Overerving zorgt er voor dat de *Geleideconstructie* ook deze attributen en relatie krijgt via *AansluitendeConstructie*. Nog een trapje hoger geldt hetzelfde voor de attributen en de relatie van *LijnvormigElement*.

Afbeelding: knipsel uit thematisch implementatie model *Lijnvormige elementen* <https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/lijnvormige-elementen/>



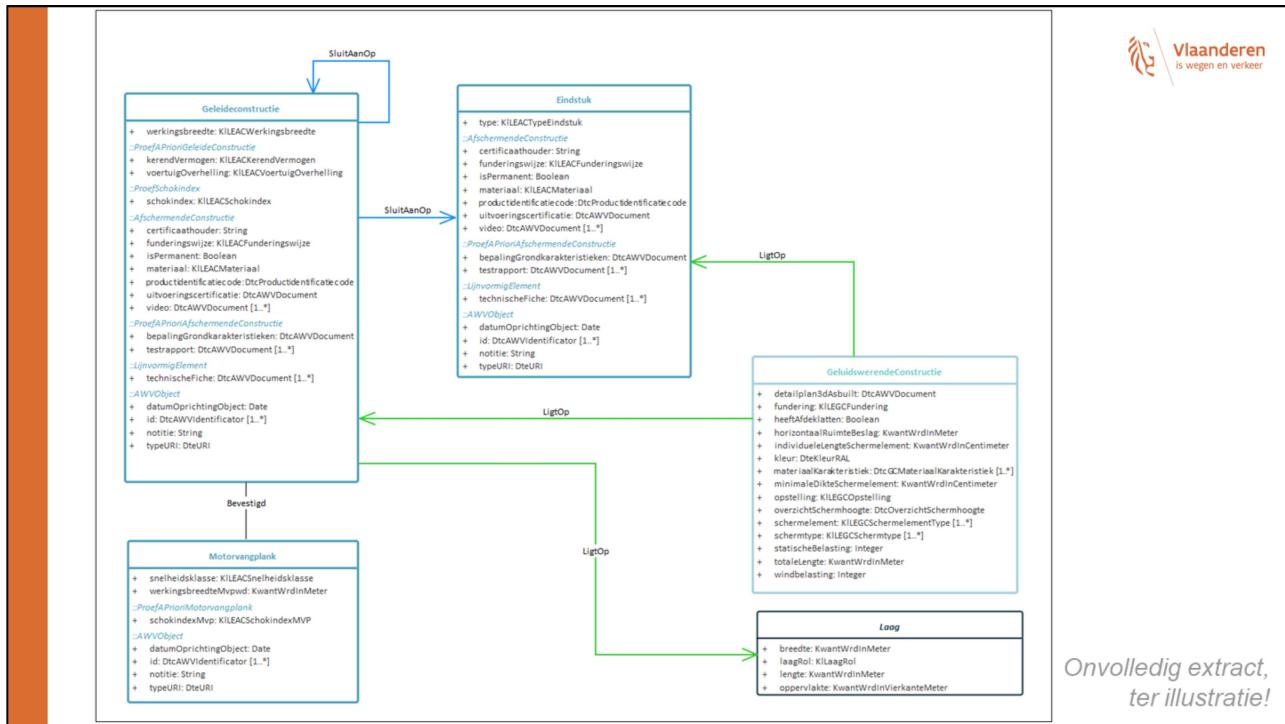
Naast de zichtbare generalisaties, erven de klassen op het diagram ook over van klassen die niet op het diagram staan. Omdat de klassen wél deel uitmaken van het Master Implementatie en de OTL als geheel, moet ook met deze klassen rekening gehouden worden.

Superklassen van een subklasse die niet op het diagram staan, worden op het diagram schuingedrukt getoond in de rechterbovenhoek van de subklasse. Deze visualisatie is eerder specifiek voor Enterprise Architect en maakt geen deel uit van de UML-specificaties. Andere software waarin de diagrammen kunnen geraadpleegd worden d.m.v. het xmi-artefact van de OTL, kunnen dit op een andere manier weergeven of geven dit misschien niet weer. In de documentatie van het betrokken implementatiemodel zijn altijd alle superklassen van een Objecttype opgenomen, ook die superklassen die niet getoond zijn in het diagram van het implementatiemodel. Voor een subklasse wordt wel enkel de eigen superklasse getoond en niet de recursieve superklassen. Aangezien achter elke superklasse een hyperlink zit, kan je evenwel binnen de pagina eenvoudig navigeren naar de opeenvolgende superklassen.

In de afbeelding zijn die “verborgen” superklassen expliciet gemaakt ter illustratie van het principe van overerving.

Afbeelding: samengesteld diagram met alle klassen van de Onderdelen Geleideconstructie en Motorvangplank. Gebaseerd op het thematisch implementatie model *Lijnvormige elementen*

<https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/lijnformige-elementen/>



Rekening houdend met de recursieve superklassen van Geleideconstructie en Motorvangplank, krijgen deze Objecttypes opeens wel wat meer om het lijf: er zijn heel wat meer attributen en relaties bijgekomen. Het is ook in deze platgeslagen versie dat data uitgewisseld wordt voor concrete objecten die gebaseerd zijn op deze Objecttypen.

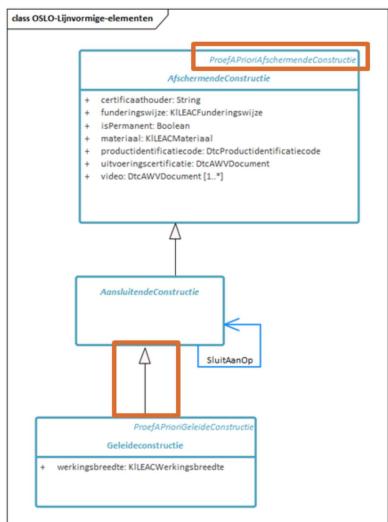
Merk op dat Enterprise Architect het mogelijk maakt om de overgeërfde attributen op een diagram te tonen in de subklasse klasse maar niet de overgeërfde relaties. De afbeelding van deze slide is een manuele herwerking van een bestaand diagram die er voor zorgt dat in het model dat gebaseerd wordt op dit diagram, de relaties dubbel zullen voorkomen: een keer via de overerving en een keer o.w.v. de rechtstreekse relatie. In de OTL zit deze verdubbeling niet!

De enige volledig geconsolideerde versie van een Objecttype is terug te vinden in het SQLite artefact van de OTL. Daarin vind je zowel alle attributen en alle relaties van een Objecttype terug.

Afbeelding: gecoördineerde versie van de Onderdelen Geleideconstructie en Motervangplank obv. hun (iteratieve) superklassen volgens de attributen van die superklassen en hun relaties. Gebaseerd op het thematisch implementatie model *Lijnvormige elementen* <https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/lijnvormige-elementen/>

Abstracten en superklassen herkennen

<https://wegenenverkeer-test.data.vlaanderen.be/ns/abstracten/>



Vocabularium Onderdelen

Klasse Geleideconstructie

Type	Klasse
URI	https://wegenenverkeer-test.data.vlaanderen.be/ns/onderdeel#Geleideconstructie
Specialisatie van	https://wegenenverkeer-test.data.vlaanderen.be/ns/abstracten#AansluitendeConstructie https://wegenenverkeer-test.data.vlaanderen.be/ns/abstracten#AWVGeleideConstructie https://wegenenverkeer-test.data.vlaanderen.be/ns/abstracten#GeleideConstructie https://wegenenverkeer-test.data.vlaanderen.be/ns/abstracten#ProefAPrioriGeleideConstructie https://wegenenverkeer-test.data.vlaanderen.be/ns/abstracten#ProefAPrioriAansluitendeConstructie
Definitie	Een doorlopende afschermende constructie voor voertuigen geïnstalleerd langs de weg of in de middenberm.

Implementatie model Lijnvormige elementen

Geleideconstructie

Beschrijving

Een doorlopende afschermende constructie voor voertuigen geïnstalleerd langs de weg of in de middenberm.

Subklasse van
[Proef a priori geleide constructie](#), [Aansluitende constructie](#)

Eigenschappen

Abstracten zijn verzameld in een **eigen Vocabularium** en hebben zo een eigen namespace binnen het OTL: <https://wegenenverkeer-test.data.vlaanderen.be/ns/abstracten/>

In de **diagrammen** die een visualisatie zijn van een implementatiemodel, zijn Abstracten herkenbaar omdat hun naam **schuingedrukt** weergegeven is. Eigen aan Enterprise Architect waarin de diagrammen gemaakt zijn en de OTL beheerd wordt, is dat de software de superklassen van een element die niet op het diagram voorkomen, weergeeft in de rechterbovenhoek van het element. Zo kan je altijd zien van welke andere Abstracten een element ook overerft.

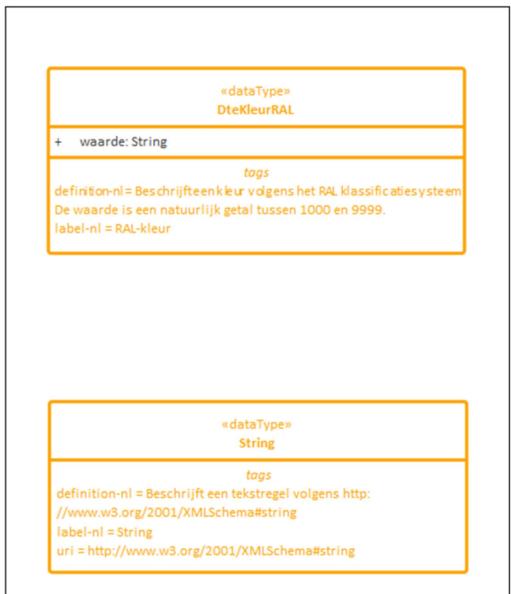
Alleen een Abstracte kan een superklasse zijn, doel van een Generalisatie-relatie. Een Abstracte kan wel een subklasse zijn van een andere Abstracte.
In het vocabularium van andere types OTL-elementen zijn de rechtstreekse superklassen van een Objecttype ook terug te vinden onder de kop **Specialisatie van**. Als superklassen zijn ze ook terug te vinden de documentatie van de implementatiemodellen bij de Objecttypen die er van overerven als subklasse onder de kop **Subklasse van** van de subklasse..

Afbeelding links: knipsel uit het diagram van het thematisch implementatiemodel *Lijnvormige elementen* <https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/lijnformige-elementen>

Afbeelding rechtsboven: knipsel uit Vocabularium Onderdelen <https://wegenenverkeer-test.data.vlaanderen.be/ns/onderdeel/#Geleideconstructie>

Afbeelding rechtsonder: knipsel uit de documentatie bij het thematisch implementatie model *Lijnvormige elementen* <https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/lijnvormige-elementen/#Geleideconstructie>

Datatypes: eenvoudige en primitieve



Eenvoudig Datatype

Prefix 'Dte'

Eén attribuut met eigen Datatype

Eigen vocabularium [Link](#)

Met vaste definitie

Primitief Datatype

Zonder prefix

Geen eigen vocabularium

Wel eigen definities die het datatype eenduidig vastleggen

Boolean, Datum, DatumTijd, ...

Attributen hebben een datatype. De **OTL legt ook alle datatypen vast** die gebruikt worden door attributen van Klassen van de OTL. Net als elk Objecttype en elk attribuut, heeft elk datatype zijn **eigen definitie** en vastgelegde betekenis. **Binnen de OTL kunnen ook uitsluitend datatypen gebruikt worden die de OTL zelf definiëert.**

In tegenstelling tot attributen die uniek zijn voor elke Klasse waarin ze gebruikt worden, kunnen datatypes van de OTL wel **hergebruikt worden**. Twee verschillende attributen kunnen maw. gebruik maken van hetzelfde datatype.

Het OTL kent enkele soorten datatype:

- Het **primitief** datatype dat eenduidig de klassieke datatypen vastlegt ongeacht de technologie waarmee data uitgewisseld wordt.
- Een **eenvoudig** datatype voor het vatten van een enkele waarde.
- **Kwantitieve waarden** die een enkele waarde vatten maar een strikte beperking bevatten die de waarde interpreerbaar maken voor een machine.
- Een **complex** datatype waarmee meerdere waarden van verschillende aard gevatt worden voor hetzelfde attribuut.
- Een **union** datatype dat aangeeft dat een attribuut een waarde heeft voor slechts één van de mogelijke kenmerken van het datatype.

Het **Eenvoudig Datatype** maakt het mogelijk één waarde van een bepaald type op te slaan voor een attribuut:

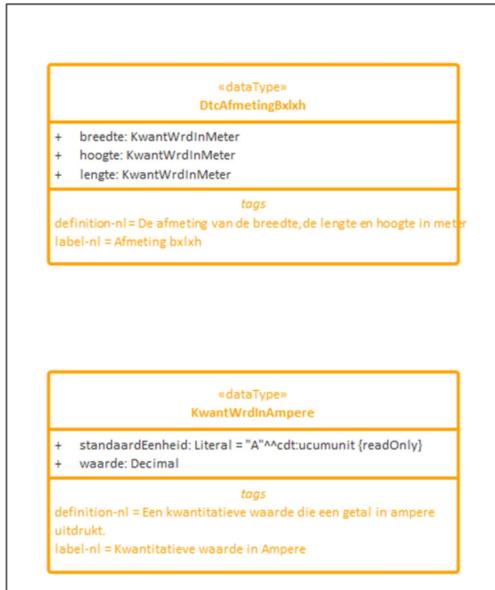
- Elk Eenvoudig Datatype heeft een eigen definitie en een vastgelegde betekenis.
- Het type van de mogelijke waarde is op zijn beurt vastgelegd als datatype van het attribuut.
- Het attribuut dat gebruik maakt van het Datatype kan een kardinaliteit hebben die moet gerespecteerd worden.
- Het is herkenbaar aan de prefix ‘**Dte**’ in de naam.
- Eenvoudige Datatypes hebben een **eigen implementatiemodel** (<https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/eenvoudige-datatypes/>).

Het **Primitief Datatype** legt eenduidig vaak gebruikte datatypes vast onafhankelijk van de technologie waarmee de data van de attributen gevat en uitgewisseld worden. Voorbeelden zijn Boolean, Datum, DatumTijd. Primitieve Datatypen:

- Hebben geen **eigen prefix**.
- Hebben geen eigen implementatiemodel..
- Ze hebben wel een eigen definitie en hun betekenis ligt eenduidig vast.

Afbeelding: visuele voorstelling van verschillende datatypes

Datatypes: complexe en kwantitatieve waarde



Complex Datatype

Prefix 'Dtc'

Meerdere attributen met eigen Datatype

Elk attribuut moet meegegeven worden

Eigen vocabularium [Link](#)

Met vaste definitie

Kwantitatieve waarde

Prefix KwantWrdIn

Geeft betekenis aan de aangeleverde primitieve

Geen eigen vocabularium

Wel eigen definities die het datatype eenduidig vastleggen

Een **Complex Datatype** waarmee **meerdere waarden van verschillende aard** vatten voor hetzelfde attribuut.

- Elk Complex Datatype heeft een eigen definitie en een vastgelegde betekenis.
- De vereiste waarden zijn vastgelegd als attributen van het element dat het Datatype representeren.
- Voor elk attribuut moet een waarde aangeleverd worden volgens de kardinaliteit van het attribuut in kwestie.
- Het type van elke waarde is op zijn beurt vastgelegd als datatype van het attribuut van het Complex Datatype.
- Het is herkenbaar aan de prefix '**Dtc**' in de naam.
- Complexe datatypes hebben een eigen **implementatiemodel** (<https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/complexe-datatypes/>).

De **Kwantitatieve Waarde** vat een enkele waarde maar heeft ook een strikte beperking die de **waarde interpreerbaar maken voor een machine**.

- Elke Kwantitatieve Waarde heeft een eigen definitie en een vastgelegde betekenis.
- Ze is herkenbaar aan de prefix 'KwantWrdIn' (Kwantitatieve Waarde in ...) in de naam van het Datatype.
- De beperking die de waarde interpreerbaar maakt, wordt door de OTL meegegeven als onveranderbare tekst die enkel kan gelezen worden.
- Elk attribuut heeft een kardinaliteit van 1.

Afbeelding complex datatype: visuele weergave van datatype uit ondersteunend implementatie model
Complexe datatypes <https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/complexe-datatypes/>

Afbeelding kwantitatieve waarde: visuele weergave van datatype vastgelegd in *Vocabularium Implementatieelement* <https://wegenenverkeer-test.data.vlaanderen.be/ns/implementatieelement/>

Datatypes: union



Union Datatype

Prefix 'Dtu'

Keuze uit een van de attributen afhankelijk van het object waarvoor het datatype gebruikt wordt

Eigen vocabularium [Link](#)

Met vaste definitie

Het **Union Datatype** geeft een keuze voor het aanleveren van waarden voor een attribuut van een Objecttype. De attributen van een Union Datatype moeten gezien worden als een **keuzelijst** waaruit **exact één attribuut** -met overeenkomstige waarde(n)- moet gekozen worden bij het aanleveren en uitwisselen van gegevens.

- Het Union Datatype is herkenbaar aan de prefix '**Dtu**' in de naam
- Elk attribuut heeft op zijn beurt een datatype dat een Eenvoudig of Complex Datatype kan zijn of een Kwantitatieve Waarde.
- Union Datatypes hebben een **eigen implementatiemodel** (<https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/union-datatypes/>).

Afbeelding complex datatype: visuele weergave van datatype uit ondersteunend implementatie model *Union datatypes* <https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/union-datatypes/>

Keuzelijsten

Eigenschap	Verwacht Type	Kardinaliteit	Beschrijving	Gebruik	Codelijst
bevestiging	Motorvangplank		Deze relatie geeft aan dat 2 onderdelen direct fysiek op elkaar bevestigd zijn. Dit kan zowel aan de buitenkant als aan de binnenkant zijn zoals bv. een camera aan een paal of een laagspanningsbord in een kast. Deze relatie heeft geen richting.		
sluit_aan_op	Eindstuk		TODO		
werkingsbreedte	Werkingsbreedte	1	Op het voorvlak van een geleideconstructie en loodrecht op de as van de weg gemeten afstand tussen de voorkant van de geleideconstructie in normale positie en de plaats van het verst uitwijkend onderdeel aan de achterzijde van de geleideconstructie bij aanrijding.	Link	

Het **datatype** van een attribuut (van een Objecttype of een Datatype van de OTL) kan ook een **keuzelijst** zijn.

Verschillende attributen kunnen dezelfde keuzelijst gebruiken.

De keuzelijsten zelf maken geen deel uit van de OTL. De OTL legt weliswaar hun definitie en betekenis eenduidig vast maar de keuzelijst zelf is geen onderdeel van de OTL **om de inhoud ervan te kunnen wijzigingen** zonder nieuwe releases van de OTL noodzakelijk te maken. Via de link onder Codelijst bij een attribuut in de documentatie van een implementatiemodel, kan de keuzelijst geraadpleegd worden.

Afbeelding boven: knipsel uit thematisch implementatie model *Lijnvormige elementen* <https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/lienvormige-elementen/#Geleideconstructie>

Thematisch implementatie model Lijnvormige elementen: <https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/lienvormige-elementen/#Geleideconstructie>

Keuzelijsten: URI voor de lijst



Vlaanderen \ DATA VLAANDEREN

Conceptschema: KILEACWerksbreedte

< <https://wegenenverkeer-test.data.vlaanderen.be/id/conceptscheme/KILEACWerksbreedte>

Gegevens

label

definitie

heeft topconcepten

dataset

Werksbreedte

Keuzelijst van de verschillende mogelijke werksbreedtes

- > [W3](#)
- > [W1](#)
- > [W5](#)
- > [W7](#)
- > [W6](#)
- > [W2](#)
- > [W4](#)
- > [W8](#)

Meer

<https://data.vlaanderen.be/id/dataset/codelist>

Keuzelijsten worden aangeboden als SKOS-lijsten

(<https://www.w3.org/2004/02/skos/>) via een eigen, **vaste URI** die als URL bruikbaar is om de definitie te raadplegen en de waarden van de keuzelijst op een bepaald moment in de tijd (a.d.h.v. de versie van de lijst).

Afbeelding: knipsel uit de webpagina voor de permanente URI voor de Keuzelijst LEAC Werksbreedte: <https://wegenenverkeer-test.data.vlaanderen.be/doc/conceptscheme/KILEACWerksbreedte>

Keuzelijsten: URI voor elke waarde in de lijst



Vlaanderen \ DATA VLAANDEREN

Concept KI^{LEAC}Werkingsbreedte/W3

< <https://wegenenverkeer.data.vlaanderen.be/id/concept/KI^{LEAC}Werkingsbreedte/W3>

Gegevens

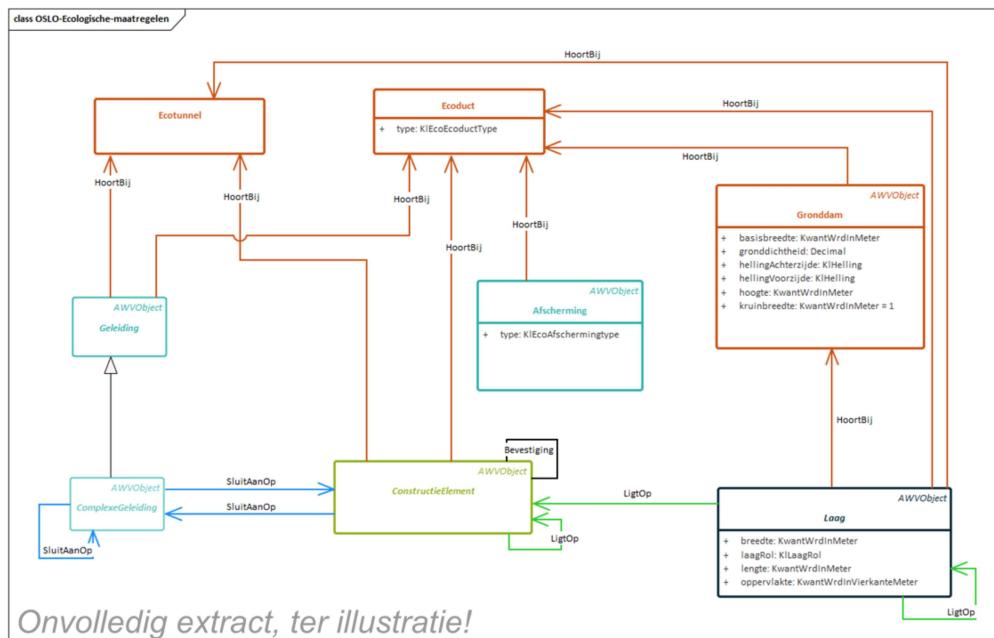
label	W3
definitie	Wn<=1,0
bijkomende info	
notatie	
heeft specialisatie	
is specialisatie van	
is topconcept van	> Werkingsbreedte
in schema	> Werkingsbreedte
dataset	https://data.vlaanderen.be/id/dataset/codelist

Elke waarde in een keuzelijst heeft ook een **eigen URI** die als URL kan gebruikt worden. Net zoals de keuzelijst zelf, heeft elke waarde ook een eenduidige definitie die de keuze voor de ene of de andere waarde mogelijk maakt.

Afbeelding: knipsel uit de webpagina voor de permanente URI voor de waarde *W3* van de Keuzelijst *LEAC Werkingsbreedte*:

<https://wegenenverkeer.data.vlaanderen.be/doc/concept/KI^{LEAC}Werkingsbreedte/W3>

Installatie



Onderdelen zijn atomaire bouwstenen van de OTL die overeenkomen met objecten uit de reële wereld. De OTL heeft evenwel ook Objecttypes nodig die verschillende **atomaire bestanddelen bundelen**. De **Installatie** is de weerspiegeling van die bundeling. De bundeling kan **fysiek** zijn, bv. de onderdelen van een installatie uit de reële wereld, **of functioneel**, bv. om alle *Onderdelen* en *Installaties* bijeen te brengen die samen een *Zone 30* (een Installatie) een specifieke vorm geven. De term *Installatie* moet dus breder gezien worden dan de representatie van een louter fysiek object.

Een Installatie **groepeert Onderdelen en andere Installaties**. De groepering gebeurt d.m.v. de **directionele relatie HoortBij** met het Objecttype dat deel uitmaakt van de bundeling als vertrekpunt en de Installatie als doel. Een Onderdeel kan nooit het doel zijn van een *HoortBij*-relatie en kan dus ook nooit een Installatie zijn.

De Installatie wordt voorgesteld als **UML-Class** en gedraagt zich ook zo. Ze kan **eigen attributen** krijgen en **relaties** van andere types dan enkel *HoortBij*. Voor de attributen en relaties gelden dezelfde regels als voor het Objecttype *Onderdeel*. Momenteel zijn er geen abstracte Installaties maar niets in de OTL sluit uit dat er in de toekomst toch abstracten komen voor Installaties. In dat geval gelden voor deze Abstracten dezelfde regels als voor de Abstracten voor Onderdelen.

Het voorbeeld toont drie Installaties: *Ecotunnel*, *Ecoduct* en *Gronddam*.

Hetzelfde Onderdeel of dezelfde Installatie en zelfs dezelfde instanties kunnen **deel zijn van instanties van verschillende installaties**. Voor de instanties gaat het dan vooral over Installaties als functionele bundeling.

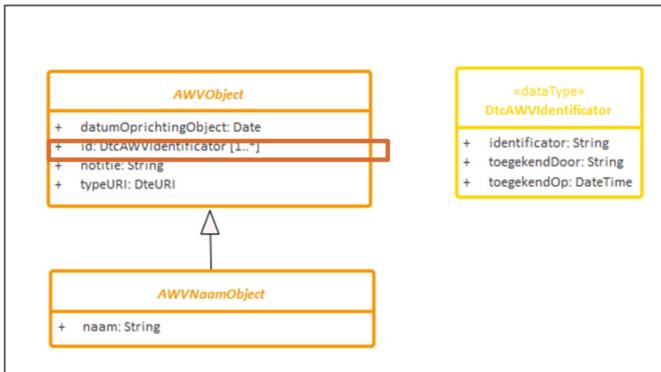
Afbeelding: knipsel uit thematisch implementatie model *Ecologische maatregelen*

<https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/ecologische-maatregelen/>



Aandachtspunten

Identificatie van objecten



Eerste aanlevering

Met **eigen ID** (ToegekendDoor = eigen organisatie)

AWV ID voor nieuw object aangemaakt (nieuwe Identifier met ToegekendDoor = AWV)

Ophalen en volgende aanlevering

Uitsluitend AWV ID (enige Identifier is die met ToegekendDoor = AWV)

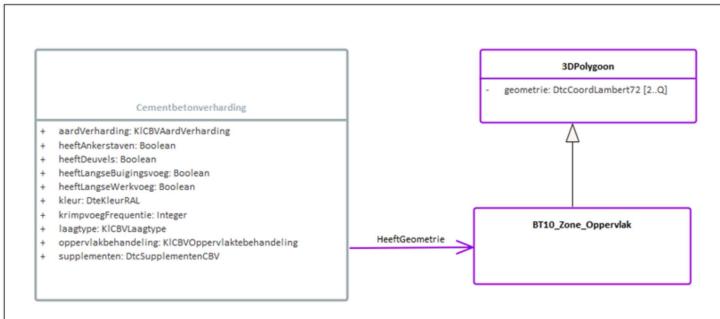
Alle Onderdelen en Installaties erven via een van hun recursieve superklassen over van **AWVObject**. Dit laat toe om een instantie van een Onderdeel of Installatie te voorzien van een **unieke id**. Hiervoor gebruikt de OTL het Complex Datatype **DtcAWVIdentifier** dat de eigenlijke waarde van de **id** bevat maar ook de **organisatie die de id toegekend** heeft (voorlopig als String maar in de nabije toekomst komen hier constraints op die er voor moeten zorgen dat er enkel gekende Organisaties kunnen gebruikt worden). Omdat het attribuut `id` van **AWVObject** een kardinaliteit van een of meer heeft, kunnen er **id's van verschillenden bronnen/organisaties** gekoppeld worden aan dezelfde instantie.

Bij **gegevensuitwisseling** **MOET** evenwel altijd de **id** gebruikt worden die **toegekend is door AWV**. In de eigen systemen kunnen verschillende id's naast elkaar gebruikt worden als toepassing van de mogelijkheden van de Abstracte **AWVObject**. Bij een eerste aanlevering, wanneer de AWV-id nog niet gekend is, wordt de id meegeleverd die toegekend is door de eigen organisatie op de manier voorzien in **AWVObject**. N.a.v. die eerste aanlevering zal AWV een id geven aan de nieuwe instantie en vanaf dan wordt enkel nog dat id gebruikt bij gegevensuitwisseling.

Afbeelding: knipsel uit thematisch implementatie model **AWV Object** <https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/awv-object/> aangevuld met de visuele representatie van het toepasselijk complex datatype uit het ondersteunend implementatie model

<https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/union-datatypes/>

Onderdeel en Installatie: het object zonder meer!



Bv. Level of geometry

Aparte klassen voor de geometrie van een onderdeel

Getypeerde relatie om te koppelen aan het onderdeel

Data aangeleverd in instanties van de betrokken Klassen

Idem voor Levenscyclus objecten, Toegang, ...

Objecttypes zijn een representatie van objecten uit de reële wereld. Ze vatten uitsluitend de eigenschappen van het object zelf. **Eigenschappen** die relevant zijn voor inventarisatie, beheer, toegang, ... maar **die niet het object zelf beschrijven**, maken geen deel uit van het Objecttype zelf maar van **Klassen die afgestemd zijn op de specifieke gevallen** waarvoor de gegevens gebruikt zullen worden. Dit maakt het mogelijk dat die ondersteunende gegevens wijzigen zonder dat de instantie van het object zelf moet wijzigen. Het laat ook toe om die ondersteunende gegevens veel meer op maat van specifieke noden aan te bieden.

Een voorbeeld zijn de locatiegegevens van een object. Hoewel dit inventarisgegevens zijn die bij het object zelf horen, zijn deze locatiegegevens in de OTL als aparte Klassen gemodelleerd. De Klasse is geassocieerd met een Objecttype door de directionele relatie *HeeftGeometrie*. Als inventarisgegeven is de locatie uiteraard onlosmakelijk verbonden met het object maar door de geometrie als aparte Klasse te modelleren kan het vereiste Level of Geometrie aangepast worden zonder dat de gegevens van het object zelf moeten aangepast worden.

Zo bevat de OTL nog ondersteunende Klassen die geen Objecttypes zijn maar die dienen om gegevens met betrekking tot een instantie te verzamelen zonder de instantie zelf te moeten wijzigen of het Objecttype waarop het gebaseerd is. Een ander voorbeeld -dat nog verder uitgewerkt zal worden- zijn Klassen m.b.t. de levenscyclus van een object en inspecties.

Deze ondersteunende Klassen blijven UML-Classes en gedragen zich dus als Onderdelen en Installaties. Ze hebben eigen attributen en relaties en kunnen ook Abstracten hebben. Ze maken ook deel uit van de OTL maar zijn daarom niet voor elke afnemer even relevant.

Afbeelding: knipsel uit thematisch implementatie model *Verhardingen en funderingen*

<https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/verhardingen-funderingen/>

De OTL en Level of Information



Volgens de OTL, indien niet explicet vermeld:

is de kardinaliteit van Attributen = 1

is de kardinaliteit van Relaties = 0..*

Elke relatie is een mogelijke relatie, geen verplichte

Relaties zijn wel beperkend

Bestekken en BIM-uitvoeringsplan

Bepalen welke Onderdelen, Installaties en ondersteunende elementen moeten aangeleverd

worden en met welke Attributen en Relaties

Binnen de OTL krijgen elementen eerder uitzonderlijk een expliciete kardinaliteit. Wanneer ze wel explicet een eigen kardinaliteit hebben, moet deze gevuld worden. Attributen en relaties zonder explicet vermelde kardinaliteit, hebben wél een standaard kardinaliteit.

De standaard kardinaliteit voor een Attribuut is 1. Dit betekent voor elk attribuut verplicht een waarde moet hebben. Die verplichting geldt evenwel slechts voor de **finite state van de instantie** dus pas wanneer de gegevensverzameling voor de instantie (het object op zich) beëindigd is. Het betekent niet dat alle vereiste gegevens als bij een eerste aanlevering deel moeten uitmaken van de aanlevering.

De standaard kardinaliteit van een Relatie is 0 of meer. Dit betekent dat de OTL geen verplichting oplegt dat een relatie tussen de Klassen ook moet voorkomen tussen instanties van de betrokken Klassen. Dezelfde instantie van de Klasse kan ook een relatie hebben met meerdere instanties van de gerelateerde Klasse. De OTL bevat wel de enig toegelaten relaties: een relatie die niet voorkomt tussen Klassen van de OTL, mogen ook niet aangeleverd worden als relaties tussen instanties van de Klassen.

Als de kardinaliteit zelf niet vastlegt wanneer **welke gegevens moeten aangeleverd worden of kunnen verwacht worden**, hoe kan je dan weten waaraan je je kan verwachten? Hiervoor is het **Bestek en het BIM-uitvoeringsplan de norm**. Zij

bepalen voor welke Objecttypen gegevens verwacht worden (incl. de relaties als data, zie boven onder Relaties) en wanneer ze verwacht zijn. De OTL moet dus altijd samen met een bestek en het BIM-uitvoeringsplan gelezen worden.



Aan de slag

OTL-conforme data verzamelen



Bestek en BIM-uitvoeringsplan

Bepalen vereiste inhoud als deelverzameling van het OTL

[Data.vlaanderen.be > Thema Wegen en Verkeer](#)

Test ([Link](#))

Productie ([Link](#))

Artefacten

Master implementatiemodel

[master implementatiemodel op Test \(link\)](#)

[master implementatiemodel op Productie \(link\)](#)

Thematische en ondersteunende modellen

xmi, SQLite, JSON-LD, SHACL, ...

Het **bestek en het BIM-uitvoeringsplan zijn normatief**. Zij geven aan welke gegevens van welke OTL-elementen uitgewisseld worden in een concreet project. Dit is dus telkens weer het startpunt om te bepalen wat je nodig hebt uit de OTL.

De **online documentatie op data.vlaanderen kan daarna in principe volstaan** om te weten wat precies moet aangeleverd worden maar vooral waarvoor de gevraagde gegevens staan. Alle definities en aanvullende documentatie van Objecttypen, ondersteunde Klassen, Attributen met hun Datatypes en Keuzelijsten en Relaties kan in de online documentatie gevonden worden.

De Test-omgeving bevat een laatste stand van zaken van de OTL in volle ontwikkeling. Nieuwe releases zullen eerst hier gepubliceerd worden om de inhoud van de OTL zelf en de integratie met externe software te kunnen testen.

<https://wegenenverkeer.data.vlaanderen.be/>

De Productie-omgeving bevat de actuele release waarop de data-uitwisseling gebaseerd wordt.

<https://wegenenverkeer.data.vlaanderen.be/>

Beide omgevingen werken op dezelfde manier, enkel hun inhoud verschilt.

Die online documentatie bevat de verschillende Vocabulary's, het Master Implementatiemodel en de verschillende thematische en ondersteunende

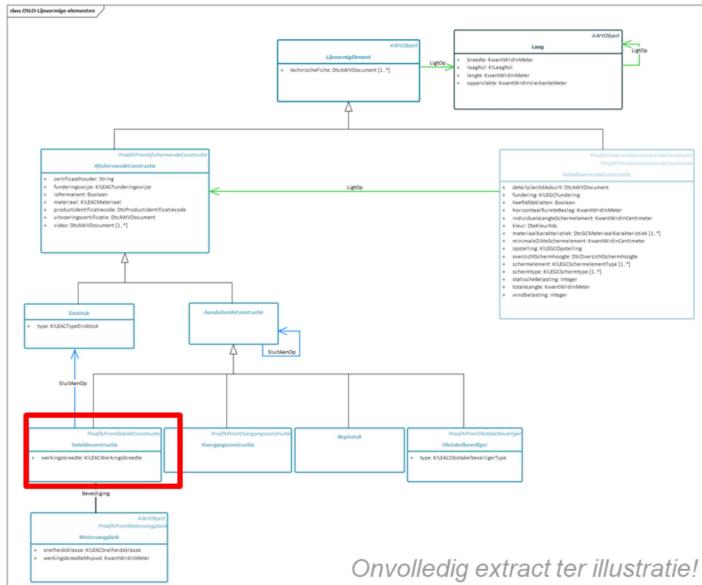
implementatiemodellen. **Alleen het Master Implementatiemodel bevat altijd alle Klassen met hun attributen en al hun mogelijke relaties.** De diagrammen bij de verschillende gedeeltelijke thematische modellen zijn een goed vertrekpunt.

Bij de online documentatie vind je ook andere artefacten die kunnen helpen bij het opzetten van een systeem voor de data-uitwisseling:

- Het **xmi-bestand** kan je gebruiken om de diagrammen en de OTL-elementen in te laden in software waarmee je UML modellen maakt. Het bevat alle OTL elementen.
- De **SQLite** is een database waarin de OTL-elementen opgenomen zijn in databank tabellen. Het bevat alle OTL elementen en van de Objecttypen altijd de gecoördineerde versie na recursieve overerving van alle superklassen.
- Andere artefacten kunnen helpen om inzicht te krijgen in de OTL maar zijn onvolledig en minder bruikbaar.

OTL integreren: van implementatie model ...

OTL integratie voor datamapping



Onvolledig extract ter illustratie!

Implementatie model Lijnvormige elementen

Geleideconstructie

Beschrijving
Een doorlopende afscherende constructie voor voertuigen geïnstalleerd langs de weg of in de middenberm.

Subklasse van
Aansluitende constructie

Eigenschappen
Voor deze entiteit zijn de volgende eigenschappen gedefinieerd: [bevestigd](#), [sluitAanOp](#), [werkingsbreedte](#).

Eigenschap	Verwacht Type	Kardinaliteit	Beschrijving	Gebruik	Codelijst
bevestigd	Neutraalgegeven		Dit relatie geeft aan dat 2 onderdelen direct fysiek op elkaar bevestigd zijn. Dit kan zowel aan de buitenkant van de weg als aan de binnenkant van de weg. Zoals bij een camera aan een paal of een laagspanningsboord in een kast. Deze relatie heeft geen richting.		
sluitAanOp	Endpunt		TODD		
werkingsbreedte	Werkingsbreedte	1	Op het voorblad van een geleideconstructie en loopt recht op de as van de weg gemeten afstand tussen de buitenkant van de geleideconstructie in normale positie en de plaats van het verst uitstaand onderdeel. De breedte van de geleideconstructie bij aanrijding.	Link	

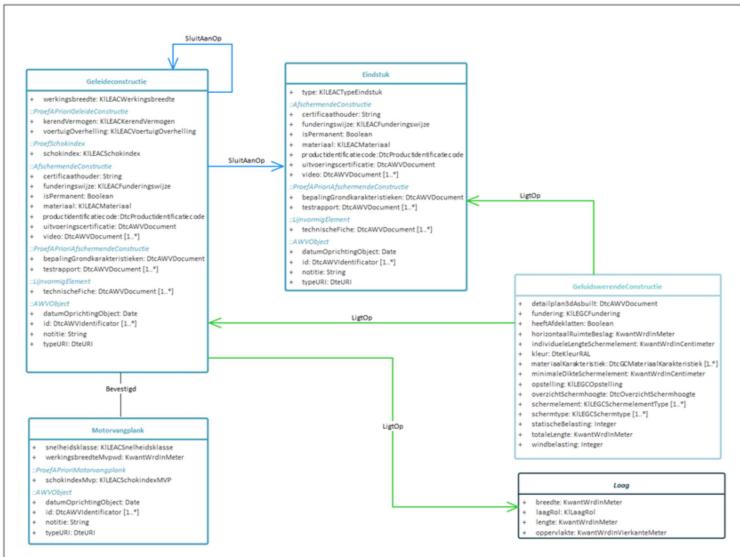
De implementatiemodellen bevatten alle elementen die relevant zijn voor het betrokken (deel)model. Je vindt er de definities terug van de Klassen en hun attributen en de relaties die relevant zijn voor het betrokken model. **Enkel het Master Implementatiemodel bevat alle Klassen en alle relaties voor die Klassen.**

De diagrammen bij de thematische implementatiemodellen zijn voor ons, mensen, vaak toegankelijker dan de tekstuele versie van het model. In de tekstuele versie kan je terugvinden wat in voorkomend geval de superklassen is van een klasse. Door te klikken op de superklasse, navigeer je naar haar details waar je dan weer haar superklassen vindt. In de diagrammen is de overerving in een oogopslag te ontdekken.

Afbeelding: knipsel uit thematisch implementatiemodel *Verhardingen en funderingen*
<https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/lijnvormige-elementen/#Geleideconstructie>

OTL implementeren: over coördinatie ...

OTL integratie voor datamapping

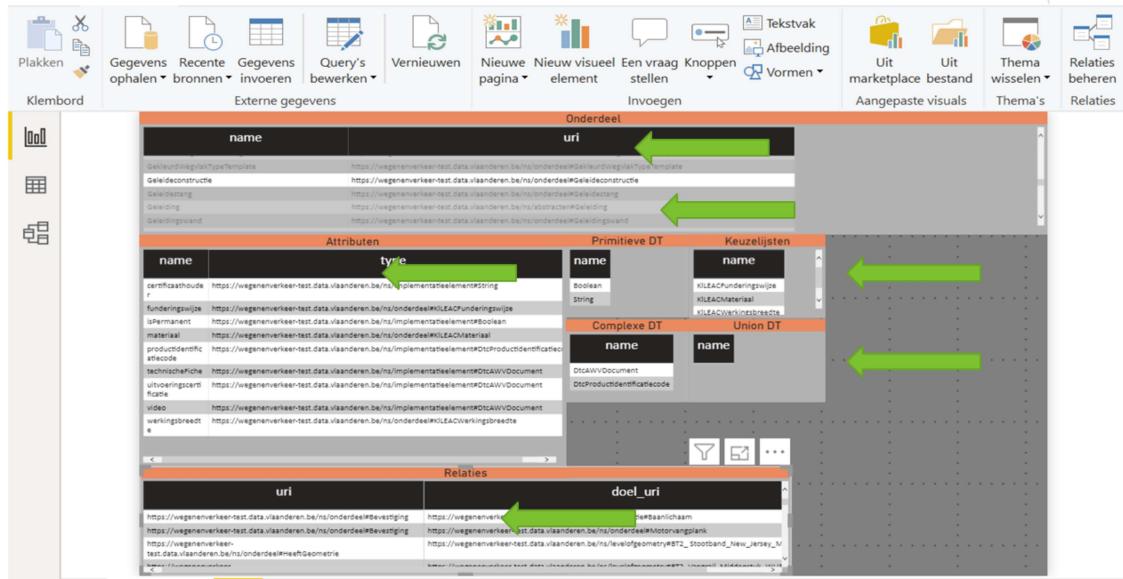


Bij de data-uitwisseling worden nooit instanties van Abstracten aangeleverd! Je moet de **overerving dus eerst oplossen** vooraleer je weet wat alle attributen en mogelijke relaties van een Klasse zijn. UML-tools kunnen je hierin helpen zoals in de afbeelding waar Enterprise Architect de overgeërfde attributen kan tonen maar niet de overgeërfde relaties die je zelf moet leggen.

Gebruik het **xmi-artefact** om de OTL te importeren in de software die jij gebruikt om klassenmodellen te maken. Dit bestand maakt het ook mogelijk om verder te bouwen op de OTL-Klassen voor je eigen systemen.

Afbeelding: visualisatie van de gecoördineerde versie van het Onderdeel *Geleideconstructie*

in SQLite ...



Zie ook [Parallelle sessie 9 Hoe implementeer je als ontwikkelaar of technisch specialist ...](#)

Het **SQLite-artefact** bevat de **volledige OTL als relationele databank**. Elke Klasse komt er ook in zijn **gecoördineerde versie** in voor dus met alle recursief overgeërfd attributen en mogelijke relaties.

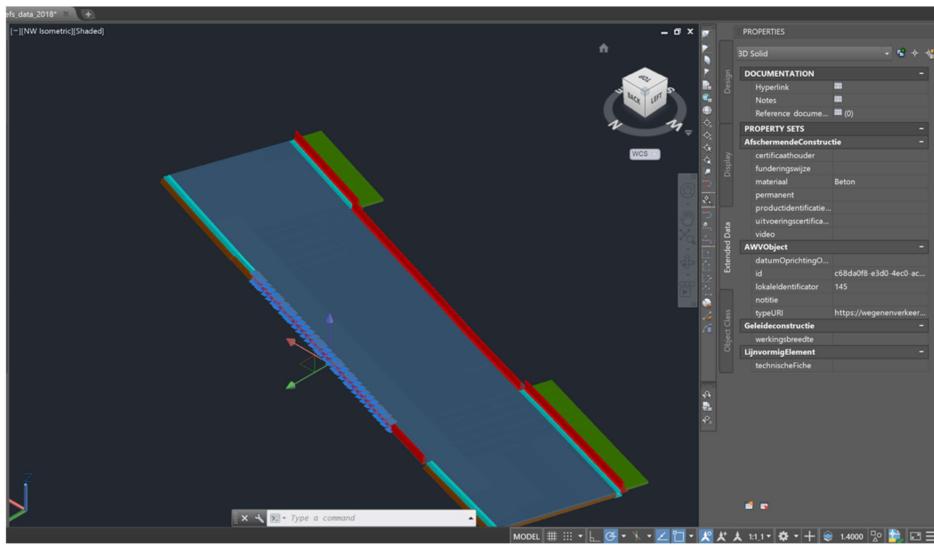
Je kan deze databank gebruiken om de OTL te verkennen maar ook de **OTL te integreren in andere software**.

Afbeelding: schermafbeelding voor (de gecoördineerde versie van) het Onderdeel *Geleideconstructie* in het SQLite artefact

tot Civil3D



OTL integratie in bestaande tooling



De OTL kan dus ook geïntegreerd worden in software die je gebruikt bij het opzetten en uitvoeren van projecten. Hier is een rol weggelegd voor de leveranciers van dergelijke instrumenten.

Afbeelding: schermafbeelding voor de integratie van (de gecoördineerde versie van) het OTL-Onderdeel *Geleideconstructie* in een bestaande toepassing.

Integratie van OTL en andere OSLO standaarden



Voor wie nog een stap verder wil gaan



Als organisatie kan je er ook voor kiezen om de OTL **als standaard te implementeren binnen je eigen organisatie**. Je kan nog een stap verder gaan en inschakelen in de OSLO²-standaarden als geheel.

Deze verregaande afstemming is niet nodig voor eenvoudige data-uitwisseling maar hoe meer aparte domeinen op elkaar afgestemd zijn, des te vanzelfsprekender wordt data-uitwisseling. Vooral voor entiteiten van de Vlaamse overheid of voor lokale overheden is dit van belang.

Integratie van OTL en andere OSLO standaarden



Voor wie nog een stap verder wil gaan

Handleiding

Profielgebaseerde handleidingen

Business verantwoordelijke Integrenen van de informatiestrategie in een projectplan.	Analist Uitwerken van een informatiemodel op basis van de behoeften.
Bestaande resources	
Proces en methode voor standaarden Document dat meer details verschafft rond het proces en de ontwikkeling van datastandaarden.	Vocabularia, applicatieprofielen en codelijsten Lijst van de beschikbare Vocabularia, applicatieprofielen en codelijsten.
OSLO tooling en publicatie Toolchain om een dataspecificatie te genereren.	Architecturale overwegingen Tips rond architectuurkeuzes voor gegevensuitwisseling op het web.

[Handreiking_\(link\)](#)

SAVE THE DATE

6/12/2019
Opleidingssessie
Aan de slag met OSLO

Voor business verantwoordelijken, analisten en ontwikkelaars



Meer over de afstemming tussen domeinen op basis van OSLO²-standaarden kan je vinden bij OSLO² zelf.

Handreiking OSLO: <https://informatievlaanderen.github.io/handreikingOslo/>



Vragen?



Meer info

Online documentatie

Meer informatie over OSLO

[FAQ](#)
[Handreiking](#)

Vlaamse URI-standaard

[PDF](#)

OTL model, diagrammen en andere artefacten:

[Test \(link\)](#) en [Productie \(link\)](#)

Achtergrond en bijkomende informatie

[Technische documentatie pdf \(link\)](#)

BIM@AWV [link](#)

[Algemene info en Nieuwsbrief](#)

Contactformulier voor alle vragen en opmerkingen

[Contactformulier \(link\)](#)

OSLO²

Veelgestelde vragen <https://wegenenverkeer-test.data.vlaanderen.be/#veelgestelde-vragen>

Handreiking <https://informatievlaanderen.github.io/handreikingOslo/>

Vlaamse URI Standaard

https://data.vlaanderen.be/cms/VlaamseURI-StandaardVoorData_V1.0.pdf

OTL implementatiemodellen, vocabularia en andere artefacten

Test-omgeving <https://wegenenverkeer-test.data.vlaanderen.be/>

Productie-omgeving <https://wegenenverkeer-test.data.vlaanderen.be/>

Achtergrond en bijkomende informatie

Link bij de documentatie op Test en Productie <https://wegenenverkeer-test.data.vlaanderen.be/doc/implementatiemodel/master/html/OTL-TechnischeDocumentatie.pdf>

Nieuwsbrief en algemene info over BIM en de rol van de OTL in BIM

<https://wegenenverkeer.be/bim>

Voor vragen en opmerkingen of BIM en OTL, zie ons contactformulier

https://docs.google.com/forms/d/e/1FAIpQLSe1Q0oH5devyappJl8bdN1vpz1S8lFeTL1OUI9LPRRV_q8IQQ/viewform

Dank u en
succes met uw BIM-project



Michiel De Keyzer
Joost De Weghe

AGENTSCHAP
WEGEN & VERKEER