

GWSW-OroX Opbouw dataset

Uitwisselprotocol

Een handreiking voor applicatiebouwers

Van: Stichting RIONED

Versiegeschiedenis:

20240423/mv	Aandacht voor multi-typing ("blinde put")
20240226/mv	Property isPartOf kan in datasets
20230823/mv	Overzicht wijzigingen per versie aangepast
20230720/mv	Tekst bij uitlaatpunt
20230310/mv	GML-versie (3.2) in namespace-verwijzing bij geometrie
20230228/mv	Extra toelichting bij Uitlaatpunt
20220317/mv	Bergbezinkbassin en Wadi toegevoegd
20210729	Toegevoegd bijlagen 1 en 2
20210729	Toegevoegd hoofdstuk "Publiceren GWSW Deelmodellen in GWSW Lijsten op apps.gswsw.nl"
20210622:	Multi-geometrieën toegevoegd aan GML-definitie
20210601:	Disclaimer uitgebreid: proces ontwerp/bouw export-import functie
20210504:	Correctie turtle-syntax bij gebruik datatype xsd:integer
20210414:	Disclaimer: gevolgen bij afwijking van deze beschrijving (zie hst Inleiding)
20200501:	Compositie Aansluitingen, alleen geen , nog uitwerken
20190812:	URI gecorrigeerd: was NiveauDoorlaat in voorbeeld, moet zijn Doorlaatliveau
20190717:	Aangepast op GWSW 1.5: Hoofdstuk "Wijzigingen per versie", nieuwe URL.
20181212:	GWSW 1.4.1: Omgaan met ontwerp-bob's, afsluiter, klep
20181129:	Conversietabel voor omzetting van GWSW-unit naar xsd:datatype toegevoegd
20180717:	GML-notatie polygonen aangepast (URL toegevoegd)
20180418:	Beperkte tekstaanpassingen
20171218:	Voorbeelden netwerk met hydraulische constructies uitgebreid
20171213:	Aangepast op GWSW 1.4. Correcties na vaststelling GWSW-Basis (incl. GWSW-Hyd) en GWSW-HydX
20171115:	Aangepast op GWSW 1.4. Nieuw hoofdstuk "Wijzigingen per versie"
20171109:	Diverse correcties en aanvullingen (voorbereiding werkgroep GWSW-OroX dd 20171110)
20171031:	Afstemming op GWSW 1.4 (te bespreken zaken in geel gemarkeerd)
20170814:	Correcties via Arie Sturkenboom
20170720:	Aangepast: Naamgeving URI toegelicht, RDF-tools voor Import
20170426:	Bijlagen (eerdere proefnemingen) verwijderd
20170425:	Toelichting structureren GWSW-OroX-export (nieuw hst 5.2)
20170422:	Naamgeving/extensie GWSW-OroX bestand
20170323:	dotNetRDF: voorbeeldcode voor inlezen grote triplestores
20170301:	Toelichting Bladerobjecten, kleine redactionele ingrepen
20161129:	Correctie tekstfouten
20161128:	Opmerkingen Eric
20161107:	Nieuwe geometrie, implementatie GWSW 1.3.1 (wijzigingen in geel gemarkeerd)
20161005:	Aanscherping syntax dataset-URI (naar aanleiding GWSW-OroX-export Moerdijk)
20160907:	Detailering ontologie
20160901:	Opmerkingen bij GWSW-OroX-export Kikker
20160530:	Diverse tekstaanpassingen. Verwijzingen naar GwsWim_Voorbeeld.ttl toegevoegd
20160511:	URL aangepast op GWSW versie 1.3
20160421:	relatie hasClassId vervalt (alleen naam als concept-identificatie in GWSW-OroX)
20160408:	GWSW-RIB, hoofdstuk 6.3 aangepast
20160205:	GWSW-RIB, tabel 6.3 aangepast
20160130:	GWSW-RIB, aanscherping bij test GWSW Browser
20151116:	GWSW-Basis, aanscherping GWSW-OroX-inrichting. Vraagpunten resteren, zie gele kaders
20151026:	GWSW-RIB: gebruik van korte URI's



Omgaan met samenstelling-relaties voor ruimte/activiteit
Overlap GWSW-OroX tussen GWSW-Basis en GWSW-RIB (welke Basis-concepten gebruiken) uitgewerkt
Uitbreiding met dataset op basis van GWSW-RIB
Enkele discussiepunten toegevoegd (gele kaders)
20150820: *Door kopgroep (Frank Zwiers, Hendrik Kingma, Marinus Vonhof) vastgesteld op 20 augustus 2015*

Inhoud

1	Inleiding	5
1.1	Gebruikte begrippen	6
1.2	Organisatie	6
1.2.1	Kopgroep GWSW-OroX	6
1.2.2	Werkgroep GWSW-OroX	7
1.3	Uitgangspunten	7
1.4	Wijzigingen per versie	7
2	Uitwissel architectuur.....	9
3	Relationele Database versus Triplestore.....	11
3.1	Van tabellen naar triples	11
3.2	Beschrijving GWSW – GWSW_Lijsten	11
3.3	GWSW concepten.....	12
3.3.1	Identificatie GWSW concepten	12
3.4	Mapping tussen Native en GWSW	13
3.5	Relaties in een dataset	13
4	Uitwisselformaat GWSW-OroX (Turtle-syntax)	15
4.1	Syntax Turtle-bestand:	15
4.1.1	Prefixes voor een GWSW-dataset	15
4.1.2	Regels Turtle-syntax	16
4.2	Naamgeving GWSW-OroX-bestand.....	17
4.3	Toegepaste syntax GWSW-OroX	17
4.3.1	Gebruik URI, Labels, Literals	18
4.3.2	Kenmerken van individuals	18
5	Dataset conform GWSW-Basis: Gestructureerd opbouwen	21
5.1	Het voorbeeld-stelsel	21
5.2	Structureren van het GWSW-OroX-bestand	22
5.2.1	Locatiegegevens / ruimtes	23
5.2.2	Activiteiten	23
5.2.3	Netwerk / topologie	23
5.3	Classificeren: indelen in soorten	24
5.3.1	Detailniveau soortenboom.....	24
5.4	Samenstelling: decompositie	26
5.4.1	Samenstelling: Ruimte / Gebied / Ondergrond.....	27
5.4.2	Samenstelling: Bestaansvorm / Activiteiten	28
5.5	Specificaties: kenmerken en waarden	29
5.6	Samenhang: Topologie en geometrie	30

5.6.1	Topologie en geometrie in GWSW Ontologie	30
5.6.2	GWSW-OroX-notatie van Topologie en Geometrie	30
5.6.3	Het kenmerk Geometrie.....	31
5.6.4	Put – Topologie en Geometrie	32
5.6.5	Leiding – Topologie en Geometrie	32
5.6.6	Samenhang: grondwaterstand en maaiveldhoogte	33
5.7	Hydraulische constructies (conformiteitsklasse GWSW-Hyd).....	34
5.7.1	Netwerk: knooppunten en verbindingen	34
5.7.2	Netwerk uitgedrukt conform het GWSW	35
5.7.3	Netwerk uitgedrukt in GWSW-OroX.....	37
6	Dataset conform GWSW-Basis: Gestructureerd inlezen	42
6.1	Selecteren/doorlopen triples vanuit het GWSW-OroX-bestand	42
6.2	Putten inlezen.....	42
6.3	Leidingen inlezen.....	42
7	Gereedschappen bij ontwikkeling GWSW-OroX export/import	44
7.1	Export naar GWSW-OroX (Turtle / Triplestore)	44
7.2	Import van GWSW-OroX-bestand (Turtle)	44
7.2.1	API voor RDF en SPARQL - dotNetRDF	45
8	Publiceren GWSW Deelmodellen in GWSW Lijsten op apps.gsw.nl.	Fout! Bladwijzer niet gedefinieerd.
9	Bijlage 1: Aanmaken nieuwe datasets en toekennen rechten binnen de GraphDB database.	Fout! Bladwijzer niet gedefinieerd.
10	Bijlage 2: Genereren van een nieuw bestand in de RDF-taal.	Fout! Bladwijzer niet gedefinieerd.

1 Inleiding

Dit document heeft betrekking op versie 1.5 van het Gegevenswoordenboek Stedelijk Water (GWSW).

Sinds mei 2015 wordt het GWSW in de praktijk toegepast. Leveranciers van beheersystemen stemmen hun datamodel af en wisselen uit met GWSW-OroX, de inspectiewereld past het GWSW-RIB model en het uitwisselformaat GWSW.ribx toe, het GWSW-Hyd voor hydraulische modellering is onlangs gepubliceerd. Andere GWSW-onderdelen zijn in ontwikkeling (Maatregelen, Gemalen).

Dit document beschrijft het generieke uitwisselprotocol voor het GWSW. Daarmee kan de vakwereld in de volle breedte gebruik maken van de GWSW Ontologie.

Gebruik van dit document:

STATUS VAN DE SPECIFICATIE

Deze beschrijving is uiteraard gebaseerd op het GWSW datamodel maar niet alle uitwisselvormen zijn daarmee afgedekt. Het document heeft geen normatief karakter maar het biedt wel voldoende houvast voor het inrichten van een dataset volgens het GWSW-OroX. Ook is er een groot aantal voorbeelden opgenomen ter verduidelijking en aanvulling.

Wij raden u dringend aan om bij afwijking van deze beschrijving - met inbegrip van de opgenomen voorbeelden - contact op te nemen met Stichting RIONED. De correcte toepassing van GWSW datasets kan bij afwijking niet worden gegarandeerd.

BOUWEN VAN DATASETS

Als u - bijvoorbeeld als softwareleverancier - een export-import functie voor GWSW datasets wilt bouwen, hou dan rekening met het volgende:

- Bouw de functie op basis van deze specificatie met in acht neming van de hiervoor genoemde status
- Neem in de testprocedure de bijdrage van een GWSW adviseur op, deze toetst de dataset op inhoudelijke kwaliteit (syntax, samenhang en volledigheid). De ervaring leert dat hiermee veel problemen in het vervolgtraject worden weggenomen.
- De vorige stap is essentieel om vervolgens effectief de website <https://apps.gsw.nl> te gebruiken voor proefnemingen met uploads en bijvoorbeeld validatie. Deze GWSW-apps zijn helaas niet ingericht om in ontwikkeling zijnde datasets uitgebreid te toetsen maar kunnen in het vervolgtraject van grote waarde zijn.

Het GWSW-team van Stichting RIONED via gsw@rioned.org wordt heel graag in de ontwikkelfase van uw werk betrokken. Wij trekken graag samen op met applicatiebouwers om goede uitwisseling conform de specificatie mogelijk te maken. Ook kunnen wij u adviseren bij het betrekken van GWSW adviseurs in uw ontwikkeltraject.

Het back end van het GWSW staat in de Gellish taal, dit is een semantische modelleringstaal in het zogenaamde ORO (Object-Relatie-Object) formaat. Hoewel Gellish een semantisch rijke taal is, wordt het als voorloper slechts op beperkte schaal gebruikt.

Voor aansluiting op de linked-data-wereld is het GWSW aan het front end omgezet naar de gebruikelijke formaten binnen het "Semantisch Web". Het W3C definieert standaarden voor het Semantisch Web met als basis een triple-vorm volgens het RDF (Resource Description Framework): de Subject-Predicate-Object constructie, waarbij de "relatie" een richting krijgt ("directed graph");

De term RDF wordt in deze notitie gebruikt voor de combinatie van RDF met de afgeleide protocollen (schema's) RDFS en OWL-2.

Het GWSW in RDF vorm verbindt zich met andere ontologiën in het Semantisch Web, uitwisseling en doorstroming van gegevens is gebaseerd op de gangbare en breed gedragen ontwikkelingen.

Het GWSW-model in RDF beschrijft een ontologie waarin zoveel mogelijk kwaliteitsborging is opgenomen. De ontologie bevat kwaliteitseisen die getoetst kunnen worden conform het RDF-protocol (“reasoning”).

Om de GWSW ontologie in RDF uit te kunnen drukken is het uitwisselingsprotocol GWSW-OroX ontwikkeld. De term “orox” staat voor **Object-Relatie-Object eXchange**, de Gellish-afkomst is nog in de naam terug te vinden. Het uitwisselingsprotocol GWSW-OroX kan worden gezien als een schema op RDF.

Deze notitie beschrijft het formaat en de toepassing van het uitwisselingsprotocol GWSW-OroX.

1.1 Gebruikte begrippen

Concept

Een soort of klasse vanuit de GWSW-Ontologie. Bijvoorbeeld “rioolput” of “leiding”.

Individu / individual

Een instantie van een concept. Zoals individual “0980” de bestaande betonnen constructie van het soort/klasse/concept “rioolput” is.

Dataset

Een dataset (voorheen BIM - “bouwwerk informatie model” - genoemd) bevat de gegevens van individuele stedelijk water systemen, de individuals op basis van het GWSW.

RDF

Het RDF staat voor Resource Description Format, de W3C basisdefinitie van modellen op basis van het triplet subject-predicate-object. In de tekst verstaan we onder RDF de combinatie van RDF met het RDF Schema (RDFS) en OWL 2 (Web Ontology Language, zie W3C).

Voor de relatie (tussen subject en object) zijn meerdere namen gebruikelijk, we hanteren in de tekst de termen “relatie”, “property” en “predicate”.

Triplestore

Een Triplestore is een triple-gebaseerde database op RDF-basis voorzien van interfaces om de data te kunnen benaderen. Een **graph-database** (refererend aan de “graaf theorie”) is een andere veel gebruikte aanduiding voor zo’n database.

SPARQL

Een protocol om **triplestores** te kunnen ondervragen en onderhouden. Een belangrijke component is de querytaal waarmee deze bewerkingen worden gedefinieerd.

1.2 Organisatie

1.2.1 Kopgroep GWSW-OroX

Enkele bedrijven (Riodesk, Sweco) hadden al eerder (in 2013/2014) ervaring opgedaan met het uitwisseling volgens het Gellish formaat. Stichting RIONED heeft deze bedrijven, versterkt met Antea, gevraagd om het GWSW-OroX te ontwikkelen, in de praktijk te toetsen en vervolgens vast te stellen.

Deze werkgroep heeft daarop de in- en exportfuncties in hun applicaties uitgewerkt en de ervaringen uitgewisseld. Op basis daarvan is de eerste versie van GWSW-OroX vastgesteld op 20 augustus 2015.

1.2.2 Werkgroep GWSW-OroX

Het GWSW-OroX is ondertussen operationeel binnen de lokale en de webbased werkomgeving (GWSW Server). Drie veel gebruikte beheersystemen wisselen conform dit formaat uit, zowel rechtstreeks onderling als via de GWSW Server.

In het najaar van 2017 is de kopgroep opgevolgd door de werkgroep GWSW-OroX. Alle leveranciers van beheersystemen voor stedelijk waterbeheer kunnen deelnemen, vooralsnog zijn de volgende partijen betrokken:

- Riodesk: beheersysteem Kikker
- Antea group: beheersysteem GBI
- DataQuint: beheersysteem GeoVisia
- DG Groep: beheersysteem Gisib
- Innovyze: beheersysteem Infonet
- Sweco: beheersysteem Obsurv



De eerste taak van de nieuwe werkgroep was de ontwikkeling van GWSW-OroX versie 1.4.

Belangrijkste aanpassing is de specificatie van bijzondere constructies zoals overlaten en rioolgemalen in het GWSW-OroX. Deze specificatie is een logisch vervolg op de uitwerking van GWSW-Hyd, de conformiteitsklasse voor hydraulische modellering.

In juli 2019 is versie 1.5 van de GWSW Ontologie uitgebracht, daarin is de nieuwe module GWSW-Hyd formeel opgenomen. In versie 1.4 is al grotendeels voorgesorteerd op GWSW-Hyd zodat er in versie 1.5 op dat vlak nauwelijks verschillen zijn.

1.3 Uitgangspunten

In de voorbeelden en in de praktijk (bij de uitwisseling van GWSW-gegevens) wordt het Turtle-formaat gebruikt. Voor GWSW-concepten geldt de prefix “gsw:”, voor individuals in een dataset wordt de prefix “bim:” (de oude term voor dataset) gebruikt. Zie ook hoofdstuk 4.1.

Voor de ontwikkeling van in- en exportfuncties van een GWSW-OroX binnen beheersystemen zijn de volgende applicaties en documenten nodig:

- Een browser voor de GWSW ontologie (op <https://data.gsw.nl> of clientbased via de (freeware) applicatie Protégé)
- De spreadsheet GWSW_Lijsten_*.xlsx met de belangrijkste overzichten van de GWSW ontologie. Deze spreadsheet staat op <https://apps.gsw.nl>.
- Een Turtle-bestand met daarin een voorbeeld-stelsel. Dit bestand is beschikbaar op <https://apps.gsw.nl>.

1.4 Wijzigingen per versie

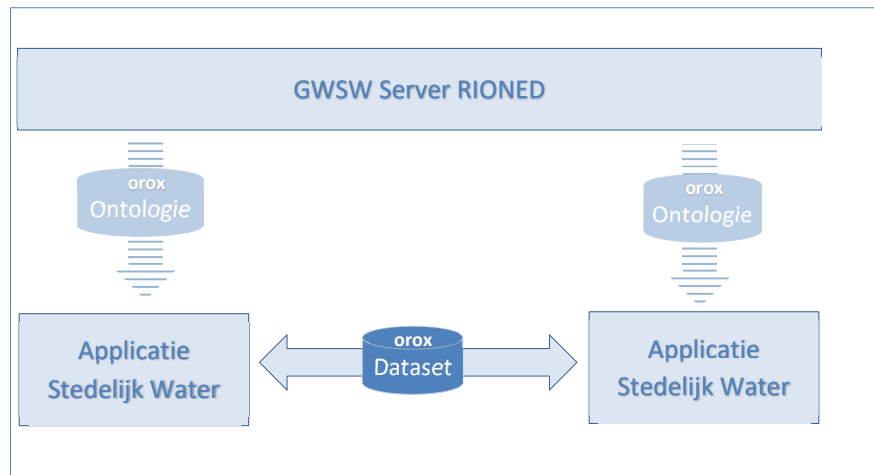
Het GWSW-OroX formaat wordt altijd gebaseerd op de GWSW_Lijsten. Een overzicht van de wijzigingen per GWSW-versie (en het daarvan afgeleide GWSW_Lijsten) staat op de homepage <https://data.gsw.nl>. In dat overzicht is ook aangegeven welke gevolgen de ontologie-wijzigingen hebben voor de toepassingen, met name voor GWSW-OroX datasets. Zie de rubriek "Omschrijving impact".

Daarnaast zijn er aanpassingen in het GWSW-OroX die betrekking hebben op het protocol, in de volgende tabel staat een overzicht van die wijzigingen.

Versie	Datum vastgesteld	Omschrijving	Zie hst.
1.6	20230116	De prefix "gws:" kan nu ook verwijzen naar de URL http://data.gws.nl/1.6/totaal/	4.1
		De eenheid "l/s" is gewijzigd in "m3/h", dit geldt dus voor datasets vanaf GWSW 1.6. Het betrekking op alle kenmerken met die eenheid (de kenmerknaam bevat de term "afvoer", "invoer", "debiet", "capaciteit", "lozing")	
1.5	20190717	De prefix "gws:" kan nu ook verwijzen naar de URL http://data.gws.nl/1.5/totaal/	4.1
		Aansluitingen: Inlaat en Ontstoppingsstuk hebben nu ook Hulpstukoriëntatie, zijn subklasse van Hulpstuk. Nieuwe types aansluitput Zandvangput, Vetvangput, enz. geïntroduceerd. Samenstelling Inlaat (relatie isPartOf) aangescherpt, gespecificeerd voor Put en Leiding	
		De temporele status van kenmerken altijd via "wijze van inwinning" aanduiden. Kenmerken die een status aangeven (zoals "ontwerp-bob") vervallen. (geïntroduceerd in GWSW vs 1.4.1)	5.5
1.4	20171113	De prefix "gws:" gebruiken voor de nieuwe URL http://data.gws.nl/1.4/totaal/	4.1
		Altijd de geometrie van leidingen meegeven, niet meer afleiden van gekoppelde putten.	5.6
		Hydraulische constructies (in GWSW-Hydx "kunstwerken", in GWSW-Hyd Wand, Doorlaat, Pomp) hoeven geen onderdeel van een omvattende constructie te zijn.	5.7
		De verbindingen (in GWSW-Hyd termen Leiding, Wand, Doorlaat en Pomp) alleen koppelen aan topologisch element van het type Knooppunt. Koppelingen (bijvoorbeeld bij externe lozingen) moeten altijd aanwezig zijn.	5.7
1.3.1	20161107	URL aangepast op GWSW versie 1.3.1	4.1
		Nieuwe geometrie, gebaseerd op geoSPARQL met gml-notatie	
1.3	20160511	URL aangepast op GWSW versie 1.3	4.1
		Relatie gws:hasClassId vervalt (alleen naam als concept-identificatie in GWSW-OroX)	

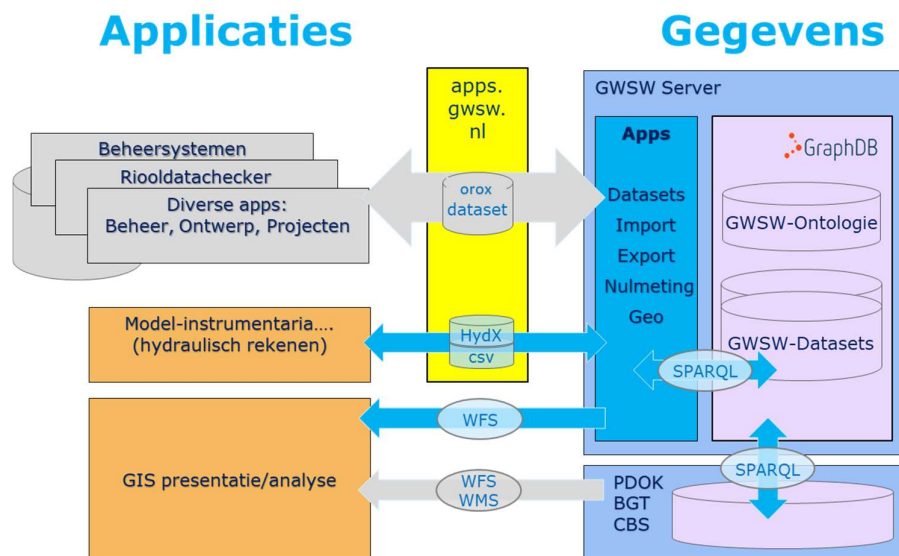
2 Uitwissel architectuur

Figuur 1: Lokale uitwisseling



Bij lokale uitwisseling is een web-verbinding niet nodig. Applicaties (veelal beheersystemen) wisselen rechtstreeks uit door middel van een GWSW-OroX dataset.

Figuur 2: Webbased uitwisseling



Op de GWSW-Server staat de GWSW-Ontologie, er is publieke toegang tot deze ontologie. Externe partijen kunnen hun eigen datasets in GWSW-OroX vorm uploaden. De GWSW Server verzorgt de validatie van deze datasets. De “individu-informatie” worden daartoe getoetst aan de kwaliteitseisen in de GWSW-Ontologie.

De server verzorgt (via GWSW Apps) ook de conversie van afwijkende formaten zoals GWSW.hydX. Die afwijkende formaten zijn soms noodzakelijk voor bepaalde disciplines maar er wordt naar gestreefd om de uitwisseling tussen de gangbare applicaties voor stedelijk waterbeheer altijd via GWSW-OroX te verzorgen. Zo beperken we de uitwisselformaten en kunnen generieke voorzieningen zoals de nulmeting van de gegevenskwaliteit worden benut.



De GWSW Server is parallel met eerste implementaties van het GWSW-OroX ontwikkeld en is als volgt benaderbaar:

data.gwsw.nl	Ontsluiting (HTML) voor raadplegen van de ontologie
data.gwsw.nl/{filter}	Ontsluiting (HTML) voor raadplegen van filters op de GWSW-ontologie. Bijvoorbeeld de module GWSW-Hyd (op data.gwsw.nl/Hyd).
review.gwsw.nl/webprotege	Ontsluiting (Webprotégé) voor review
apps.gwsw.nl	Toepassingen op de GWSW Server voor import datasets, nulmeting en omzetting naar specifieke formaten
sparql.gwsw.nl	De GraphDB workbench op het SPARQL-endpoint voor rechtstreekse ontsluiting van GWSW ontologie en datasets

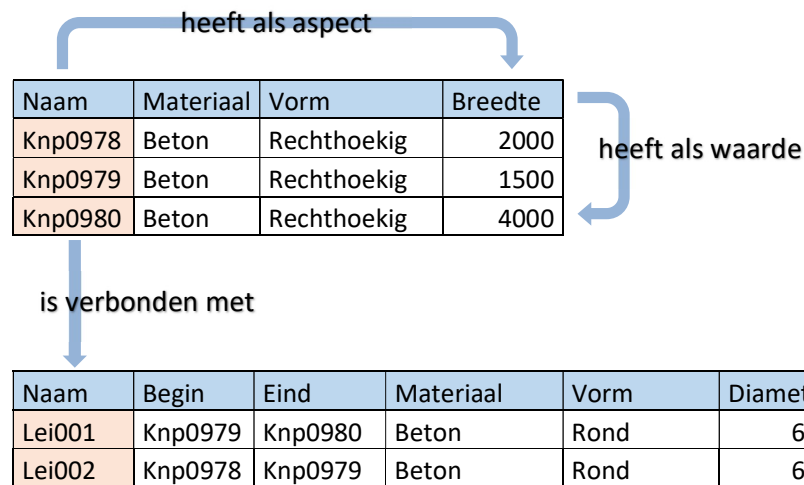
3 Relationale Database versus Triplestore

3.1 Van tabellen naar triples

De grootste uitdaging ligt bij het exporteren en importeren van GWSW-OroX bestanden. Er dient een transformatie (“mapping”) tussen native database en een triple-gebaseerd bestand ontwikkeld te worden.

De volgende figuren illustreren het verschil tussen traditionele database-tabellen en RDF-triples:

Database-tabellen Putten en Leidingen



De database-tabellen worden in RDF als triples beschreven:

```
<Knp0980> <is van het type> <Put>
<Knp0980> <heeft breedte> 4000
<Knp0980> <is verbonden met> <Lei001>
```

<Put> = de tabel-naam
 <heeft breedte> = de tabelkolom-naam
 <is verbonden met> = tabel-relatie

Of, met specificatie van het aspect:

```
<Knp0980> <is van het type> <Put>
<Knp0980> <heeft als aspect> <Breedte_1>
<Breedte_1> <is van het type> <BreedtePut>
<Breedte_1> <heeft de waarde> 4000
<Knp0980> <is verbonden met> <Lei001>
```

<Put> = de tabel-naam
 <Breedte_1> = de tabelcel-naam
 <BreedtePut> = de tabelkolom-naam
 4000 = de tabelcel-inhoud
 <is verbonden met> = tabel-relatie

Deze laatste, uitgebreidere methode wordt toegepast in het GWSW. Daarmee wordt het mogelijk extra (meta)gegevens aan een aspect (“tabel-cel”) toe te kennen en aspecten binnen een aspect te definiëren.

3.2 Beschrijving GWSW – GWSW_Lijsten

Als hulpmiddel bij de omzetting vanuit relationele databases (RDB) naar GWSW-OroX wordt een spreadsheet GWSW_Lijsten geleverd met daarin de volgende overzichten (in werkbladen):

- **Klassen:** alle in het GWSW opgenomen concepten
 - Name: identificatie-string van het concept (in CamelCase). De is het UID in RDF.
 - Number: identificatie-nummer van het concept

- Unit: gebruikte eenheid (mm, yyyyymmdd, enz)
- Label: naam van het concept
- Code: de alternatieve notatie van het concept. Eventueel variërend per context.
- Definition: GWSW-definitie
- DefinedBy: Verwijzing naar de normen met definitie
- SuperClass: De superklasse bij het concept
- **Relaties:** de voor een dataset relevante relaties tussen de concepten
 - Subject: het linker-concept, de relatie geldt ook voor alle subklassen van het Subject
 - Property: de relatiesoort
 - Object: het rechter-concept, de relatie geldt ook voor alle subklassen van het Object
- **Soortenboom:** de “taxonomie”, per concept de bijbehorende superklassen
 - Class: het betreffende concept (alle GWSW-concepten staan in de tabel)
 - SuperClass1-5, de soortenboom waartoe het concept behoort
- **Collecties:** de “domeintabellen” of “codetabellen”
 - Aspect: het kenmerk dat refereert aan een element uit de collectie
 - Collection: de naam van de collectie
 - NameElement: de element-namen en bij de collectie, de UID in RDF
 - NumberElement: identificatie-nummer van het element
 - CodeElement: de alternatieve notatie van het element. Eventueel variërend per context.

Deze spreadsheet met het betreffende blad wordt in dit document verder aangegeven met “GWSW_Lijsten/Bladnaam”.

In principe bevat de spreadsheet de vrijgegeven modules van het GWSW. De spreadsheet is verkrijgbaar via <http://www.riool.net/-/informatie-over-gwsw-voor-ontwikkelaars>.

3.3 GWSW concepten

Zoals genoemd zijn in GWSW_Lijsten/Klassen alle GWSW-concepten opgenomen. De RDF praktijk is het hanteren van begrijpbare namen voor concepten. Daarbij wordt de CamelCase / camelCase notatie van de namen voor zowel predicates (starten met lowercase) als concepten (starten met uppercase) gehanteerd.

Id_Naam	Label	Id_number	Unit	Definition
HoogteLeiding	Height	110003046	mm	De bij het materiaal gebruikelijke aanduiding van de hoogte van een leiding (bijvoorbeeld beton = binnenhoogte, kunststof = buitenhoogte)
Vervuilingsgraad	Vervuilingsgraad	111005558		Het percentage van de inwendige hoogte waarmee de leiding gevuld is met verontreiniging.
Activiteit	Activiteit	111001924		
Functionaliteit	Functionaliteit	111005395		
TijdstipMeting	Tijdstip meting	120001026		
ReinigenHoge_drukRioolput	Reinigen hoge-druk rioolput	111001132		
Deelreparatie	Deelreparatie	111000859		

3.3.1 Identificatie GWSW concepten

De conceptnaam (UID, kolom id_naam in GWSW_Lijsten) in het GWSW-model staat in CamelCase notatie. De UID wordt opgebouwd door de uitgeschreven conceptnaam te comprimeren (spaties en bijzondere tekens verwijderen) en in camelcase te plaatsen.

De uitgeschreven conceptnaam komt terug als object bij property rdfs:label. Binnen RDF is de property rdfs:label ook niet uniek per subject. Zoals gebruikelijk wordt in RDF elke vertaling in een apart label gezet.

Naast de conceptnaam kent het GWSW-model ook het identificatienummer (kolom `id_number` in `GWSW_Lijsten`) als unieke identificatie. Wel wordt er naar toegewerkt om in toekomstige versies alleen de UID als identificatie te gaan hanteren, dan komt het (Gellish-specifieke) `id_number` te vervallen.

De camelcase UID kunnen in sommige gevallen zeer lang worden. Bijvoorbeeld bij de concepten uit de EN13508-2, daarin zijn inspectie-waarnemingen en karakterisering van daarvan opgenomen. Vooral de bijbehorende keuzelijsten zijn in de standaard UID nauwelijks bruikbaar. In zo'n geval is er voor gekozen om voor de UID de EN-coderingen te hanteren. Omdat in de inspectiewereld is het gebruikelijk de EN-coderingen te hanteren. De UID blijft daarmee begrijpelijk en is gelijker tijd taalafhankelijk geworden.

3.4 Mapping tussen Native en GWSW

In de volgende tabel staat een voorbeeld van GWSW-concepten gelinkt aan een native database.

Tabel	Kolom	Inhoud	GWSW Id_name	GWSW Id_number	GWSW Unit
Grp	TypGrp	Code G	Afvalwatersysteem	110000001	
Grp	TypGrp	Code 1	Hemelwaterstelsel	110000006	
Grp	TypGrp	Code 0	GemengdStelsel	110000008	
Grp	TypGrp	Code R	Randvoorziening	110000050	
Grp	TypGrp	Code 2	Vuilwaterstelsel	110000106	
Grp	TypGrp	Code C	HorizontaalDrainagesysteem	110000108	
Grp	TypGrp	Code W	Wadi	110000108	
Grp	_wgb		StedelijkWatersysteem	110000004	
Grp	TypGrp	Code P	Grondwatermeetnet	111001112	
Grp Afm	Hgt		HoogteReservoir	110001065	mm
InspLei	_opdr		contractor	109999883	
InspLei	_proj		Rioleringsinspectieproject	110001534	
InspLei	_insplei		VisueelInspecterenVrijvervalLeiding	111000030	

De rechter kolommen bevatten de invulling vanuit `GWSW_Lijsten/Klassen`. Afhankelijk van de native datastructuur wordt deze lijst samengesteld met de corresponderende datatabellen en kolommen.

Geadviseerd wordt om in de mapping zowel naam als nummer te gebruiken, in toekomstige versies kan het nummer vervallen (zodra de naamgeving stabiel is, zie hoofdstuk 3.3.1).

Hier (en in voorbeelden) aandacht voor multi-typing: altijd functionele objecten, als 2^e typering de "uitvoerings"-objecten zoals "blinde put".

3.5 Relaties in een dataset

`rdfs:seeAlso` toevoegen voor URL naar bijlagen, figuren bij individuen e.d. (20210621).

In een dataset worden de volgende predicates gebruikt.

Predicate	Toelichting
<code>rdf:type of a</code>	Subject is van het type Object (klasse-naam)
<code>rdfs:label</code>	Subject heeft als naam Literal
<code>gsw:hasAspect</code>	Subject heeft als kenmerk Object
<code>gsw:hasValue</code>	Subject heeft als waarde Literal (subject is kenmerk)

gsw:hasReference	<i>Subject heeft als referentie Object (subject is kenmerk)</i>
gsw:hasInput	<i>Subject heeft als invoer Object</i>
gsw:hasOutput	<i>Subject heeft als uitvoer Object</i>
gsw:hasPart	<i>Subject heeft als deel Object</i>
gsw:isPartOf	<i>Subject is deel van Object</i>
gsw:hasConnection	<i>Subject heeft verbinding met Object (symmetrisch, subject en object uitwisselbaar)</i>
gsw:hasRepresentation	<i>Subject heeft als representatie Object</i>

Alle voor een dataset relevante relaties tussen de concepten staan in GWSW_Lijsten/Relaties:

OwnerOfProperty	Subject	Property	Object
Leiding	VrijvervalRioolleiding	hasAspect	Leidingorientatie
Leiding	VrijvervalRioolleiding	hasAspect	MateriaalLeiding
Leiding	VrijvervalRioolleiding	hasAspect	VormLeiding
Leiding	VrijvervalRioolleiding	hasAspect	Verbindingstype
VrijvervalRioolleiding	VrijvervalRioolleiding	hasAspect	Lining
VrijvervalRioolleiding	VrijvervalRioolleiding	hasAspect	AfvoerendOppervlak
VrijvervalRioolleiding	VrijvervalRioolleiding	hasConnection	Rioolput
VrijvervalRioolleiding	VrijvervalRioolleiding	hasConnection	Rioolput

Relaties kunnen geërfd worden van superklassen. In principe zijn alle relevante geërfde relaties per subject in de tabel opgenomen. Sorteren op property en vervolgens op subject geeft de, al dan niet geërfde, relaties per GWSW-concept.

4 Uitwisselformaat GWSW-OroX (Turtle-syntax)

Eerder geschetste ontwikkelingen gaven de koers al aan: het GWSW-OroX uitwisselformaat is gebaseerd op RDF, aansluitend op het Semantisch Web. Het GWSW-ontologie bevat de metagegevens en de restricties die gelden voor GWSW-concepten. Een GWSW-dataset bevat veel minder overhead dan het model, er kan volstaan worden met enkele header-gegevens (zoals GWSW-versie, exporterende applicatie).

Er zijn meerdere uitwisselvormen voor RDF. Die zijn allen onderling uitwisselbaar, alleen de syntax verschilt. GWSW-OroX wordt in het zogenaamde Turtle-formaat beschreven.

De overwegingen bij RDF Turtle op een rij:

- De syntax is helder en ontworpen voor triples. In het alternatieve RDF/XML formaat kan subject als xml-attribuut van xml-element <RDF:Description> en als xml-element onder <RDF:Description> voorkomen. Dat maakt het complex. XML sluit als structuur niet goed aan, is oorspronkelijk niet voor triples bedoeld.
- RDF Turtle heeft geen beperking in URI's (lokale naamgeving/karakters);
- RDF Turtle heeft compacte syntax voor toevoegen van eenheden en taal;
- RDF Turtle wordt algemeen gebruikt, de Turtle-syntax komt ook terug in de SPARQL syntax (de querytaal voor Triplestores);
- Voor het alternatieve RDF/XML formaat zou gebruik van XML-parsers een voordeel kunnen zijn maar die blijken nauwelijks bruikbaar;

4.1 Syntax Turtle-bestand:

Gebruik voor de vulling van een GWSW-OroX Turtle bestand de UTF-8 karakterset.

Bovenaan in het bestand wordt de namespace-prefix vermeld, analoog aan namespace-alias voor xml/xsd formaten:

```
@prefix rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:     <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:    <http://www.w3.org/2002/07/owl#> .
@prefix xsd:      <http://www.w3.org/2001/XMLSchema#> .
@prefix skos:     <http://www.w3.org/2004/02/skos/core#> .
@prefix geo:      <http://www.opengis.net/ont/geosparql#> .
@prefix gws:      <http://data.gws.nl/1.4/totaal/> .
@prefix bim:      <http://data.gws.nl/bim/juinen/#> .
```

Deze prefixdefinities zijn in feite ook triples (ze bestaan uit drie componenten afgesloten met een ".").

4.1.1 Prefixes voor een GWSW-dataset

In de gaande praktijk staat de keuze voor de prefix vrij. Het is wel gebruikelijk om prefixes zo min mogelijk te variëren. Consequent gebruik van prefixes is ook van belang bij federated queries, waarbij datasets gecombineerd worden ondervraagd.

In het GWSW-OroX gebruiken we (vanaf versie 1.4) de prefix: [gws](#) om te verwijzen naar de gehanteerde versie van de GWSW-ontologie.

De URL waarnaar de prefix verwijst kan wijzigen afhankelijk van de beschikbare versie van het GWSW. Alle versies vanaf GWSW 1.3 worden ondersteund door de GWSW Server.

Vanaf 12 jul 2019 kan het GWSW-OroX verwijzen naar de URL: <http://data.gwswnl/1.5/totaal/>
 Vanaf 1 jan 2018 kan het GWSW-OroX verwijzen naar URL: <http://data.gwswnl/1.4/totaal/>
 Vanaf 9 nov 2016 kan het GWSW-OroX verwijzen naar de URL: <http://data.gwswnl/1.3.1/totaal/>
 Vanaf 1 mei 2016 kan het GWSW-OroX verwijzen naar de URL: <http://data.gwswnl/1.3/totaal/>

In de voorbeelden wordt de prefix “bim:” gebruikt voor de individuals binnen een dataset. Dus voor bijvoorbeeld voor de fysieke objecten met hun kenmerken.

De overige prefixes zijn van algemene (RDF) aard.

4.1.2 Regels Turtle-syntax

Na de prefixdefinitie volgen de triples (“subject”-“predicate”-“object”). Hou hierbij de volgende regels aan:

- Triples separated by .
- URI’s in <> (*opm: URI’s vormen concept-identificaties/namen*)
- Literals in “ ”
- URI’s can be abbreviated by prefixed name spaces (*opm: dataset-toepassing*)
 - *opm: geen bijzondere tekens in korte URI gebruiken: @ ?*
- Three types of triple abbreviations:
 - Parallel: Same subject via ; separator
 - Parallel: Same subject-predicate via , separator
 - Sequential: End of one triple is start of the next triple via [...] grouping
 - Special
 - Typed literal add ^^xsd: ... (include xsd prefix)
 - Untyped literal can be multi-lingual via @ ...like @nl
 - Blank nodes via _:... or [...] grouping
 - Lists via / (of type: unordered list, ordered list, set of choices)
 - Reification via extra meta-node + 3 standard references
 - “a” is shorthand for “rdf:type”

Bron: Michel Böhm

Triples kunnen dus ook gebundeld (afgekort) worden:

1. het subject éénmalig noemen en vervolgens de bijbehoren predicates+objects
2. het subject+predicate éénmalig noemen en vervolgens de bijbehorende objects
3. geschakelde triples, het object wordt het subject van de volgende triple

Hou hierbij de volgende regels aan:

1. Parallel triples with common source
 - a1 b1 c1 .
 - a1 b2 c2 .
 - shorthand: a1 b1 c1 ; b2 c2 . (note also allowed: a1 b1 c1 ; b2 c2 . ; . (TBC-way)
2. Parallel triples with common source and predicate
 - a1 b2 c2 .
 - a1 b2 c3 .
 - shorthand: a1 b2 c2 , c3 .
3. Sequential triples where the end of one triple is the start of another triple
 - a1 b3 c4 .
 - c4 b4 c5 .
 - shorthand: a1 b3 [c4 b4 c5] .

Bron: Michel Böhm

4.2 Naamgeving GWSW-OroX-bestand

Voor bestanden met gegevens conform de Turtle-syntax is de extensie ttl gebruikelijk, RDF-georiënteerde applicaties zijn hierop afgestemd. Om daarnaast te markeren dat een turtle-bestand conform de GWSW-OroX specificaties is opgezet is de extensie orox.ttl voorgeschreven. Dus een GWSW-OroX-bestand met gegevens van rioolstelsel Drup in de gemeente Juinen kan als volgt worden genoemd:

[Juinen_Drup.orox.ttl](#)

4.3 Toegepaste syntax GWSW-OroX

Voorbeeld van een GWSW-OroX met namespaces en enkele triples:

```
@prefix rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:     <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:    <http://www.w3.org/2002/07/owl#> .
@prefix xsd:      <http://www.w3.org/2001/XMLSchema#> .
@prefix skos:     <http://www.w3.org/2004/02/skos/core#> .
@prefix geo:      <http://www.opengis.net/ont/geosparql#> .
@prefix gws:      <http://data.gws.nl/1.4/totaal/> .
@prefix bim:      <http://data.gws.nl/bim/juinen/#> .

bim:Stelsel_1    rdfs:label      "DM" .
bim:Stelsel_1    rdf:type        gws:GemengdStelsel .
bim:Stelsel_1    gws:hasPart     bim:Knp0980 .
bim:Knp0980      rdfs:label      "0980" .
bim:Knp0980      rdf:type        gws:Inspectieput .
```

Het gedeelte met de "bim:"-triples kan ook ingekort worden geschreven, de subjecten worden dan na een ";" niet in de volgende triple herhaald. Dat ziet het er als volgt uit:

```
bim:Stelsel_1    rdfs:label      "DM" ;
                  rdf:type        gws:GemengdStelsel .
bim:Stelsel_1    gws:hasPart     bim:Knp0980 .
bim:Knp0980      rdfs:label      "0980" ;
                  rdf:type        gws:Inspectieput .
```

De URI [bim:Stelsel_1](#) is een dataset-individual, de URI [gws:GemengdStelsel](#) verwijst naar een GWSW-concept (zie GWSW_Lijsten/Klassen).

De URI [bim:Stelsel_1](#) is noodzakelijk om het individu te kunnen onderscheiden. Voor deze URI kan het beste een begrijpbare naam worden gekozen. Het label kan de door de beheerder gebruikte naam of codering te bevatten.

4.3.1 Gebruik URI, Labels, Literals

Er zijn in RDF Turtle meerdere mogelijkheden voor het gebruik van labels en URI's. Voor het GWSW-OroX houden we uniform het volgende aan:

- De URI dient in het GWSW-OroX alleen voor identificatie binnen de Turtle-syntax, voor de eindgebruiker heeft het geen betekenis;
- Let op, voor de URI geldt een specifieke karakterset. Tekens zoals `/`, `.` kunnen niet in de individual-namen (die deel uitmaken van de URI, bijvoorbeeld `bim:Stelsel_1`) gebruikt worden. Ook kunnen de gereserveerde scheidingstekens in een URI (zoals `?`, `&`, `#`) niet toegepast worden. Geadviseerd wordt om voor de URI's de GWSW-conceptnamen met een volgnummer te hanteren. Bijvoorbeeld een inspectieput krijgt de URI `"bim:Inspectieput_100"`. De karaktergroep a-z, A-Z, 0-9, `_` is daarnaast altijd bruikbaar.
- De labels bij de concepten Stelsel (`"DM"`), Rioolput (`"0987"`) en Leiding (`"0987-0989"`) dienen de gangbare identificatie (namen) voor de eindgebruiker/beheerder te bevatten;
- In het label staat dus de gangbare objectnaam. Als bijvoorbeeld de rioolputnaam `"0987"` in meerdere stelsels binnen hetzelfde beheergebied maar in verschillende stelsels voorkomt blijft deze via de samenstelling in het GWSW-OroX (zie paragraaf 5.3) eenduidig te identificeren.
- Het voorkomen van bijzondere karakters in tekst-literals (de stringwaarde bij [hasValue](#) zoals een label-waarde) is veelal geen probleem. Er is een enkele uitzondering omdat de string-literal in Turtle tussen aanhalingstekens staat. Om die reden moeten in de string voorkomende aanhalingstekens `"` vervangen worden door `\`. Dus label [den "Haag" wijk](#) wordt in het GWSW-OroX uitgeschreven als [den \"Haag\" wijk](#). In triplevorm:

```
bim:DenHaagWijk    rdfs:label          "den \"Haag\" wijk" .
```

4.3.2 Kenmerken van individuals

Voor kenmerken van individuals (zoals bijvoorbeeld het materiaal van put `bim:Knp0980`) is niet persé een URI nodig. In die gevallen kan ook een zogenaamde "blank node" worden gebruikt. Die worden aangeduid met `_:nnn` (nnn = unieke naam). De volgende mogelijkheden zijn er:

Aspecten met URI aanduiden:

<code>bim:Knp0980</code>	<code>rdf:type</code> <code>rdfs:label</code>	<code>gws:Inspectieput ;</code> <code>"0980" ;</code>
<code>bim:Mat_Knp0980</code>	<code>gws:hasAspect</code>	<code>bim:Mat_Knp0980 .</code>
<code>bim:Mat_Knp0980</code>	<code>rdf:type</code> <code>gws:hasReference</code>	<code>gws:MateriaalPut .</code> <code>gws:Beton .</code>

Aspecten via blank node aanduiden:

<code>bim:Knp0980</code>	<code>rdf:type</code> <code>rdfs:label</code>	<code>gws:Inspectieput ;</code> <code>"0980" ;</code>
<code>_:Materiaal_111</code>	<code>gws:hasAspect</code> <code>rdf:type</code> <code>gws:hasReference</code>	<code>_:Materiaal_111 .</code> <code>gws:MateriaalPut ;</code> <code>gws:Beton .</code>

De volgende (sequentiële) methode heeft de voorkeur vanwege leesbaarheid en omvang. Aspecten niet benoemen, alleen het type en de waarde aangeven:

```
bim:Knp0980      rdf:type      gsw:Inspectieput ;
                  rdfs:label    "0980" ;
                  gsw:hasAspect
[
                  rdf:type      gsw:MateriaalPut ;
                  gsw:hasReference gsw:Beton
] .
```

Waarde-toekenning

De elementen van collecties (domeintabellen) zijn in het GWSW als aparte concepten benoemd. Als kenmerken (zoals [MateriaalPut](#)) de waarde van een concept krijgen, wordt de relatie [hasReference](#) gebruikt. Als kenmerken een "datatype" als waarde krijgen, dan geldt de relatie [hasValue](#).

Voorbeelden waarde-toekenning:

```
bim:Knp0980      rdf:type      gsw:Inspectieput ;
                  rdfs:label    "0980" ;
                  gsw:hasAspect
[
                  rdf:type      gsw:MateriaalPut ;
                  gsw:hasReference gsw:Beton
] ,
[
                  rdf:type      gsw:BreedtePut ;
                  gsw:hasValue   4000 (of "4000"^^xsd:integer)
] .
```

In dit voorbeeld is ook de afkorting voor de combinatie subject+predicate gebruikt. De waarde voor twee aspecten is ingevuld bij de combinatie [bim:Knp0980 gsw:hasAspect](#).

De breedte van de put wordt uitgedrukt in [mm]. In GWSW_Lijsten/Klassen bevat de kolom "Unit" de te hanteren eenheid in GWSW-OroX bestanden.

Specificeren datatypes

Datatypes bij waarden kunnen in veel gevallen automatisch afgeleid worden uit de syntax. Voor datum/tijd datatypes is echter een specifieke typering nodig, de RDF-validatie zal daar anders een foutmelding op geven.

```
bim:Knp0980      rdf:type      gsw:Inspectieput ;
                  gsw:hasAspect
[
                  rdf:type      gsw:YearCameIntoOperation ;
                  gsw:hasValue   "1980"^^xsd:gYear
] ,
[
                  rdf:type      gsw:DateOfInspectionManhole ;
                  gsw:hasValue   "2013-01-05"^^xsd:date
] ,
[
                  rdf:type      gsw:TimeOfInspectionManhole ;
                  gsw:hasValue   "11:50:00"^^xsd:time
] .
```

In de volgende tabel is aangegeven hoe de eenheden binnen de GWSW Ontologie zijn vertaald naar datatypes:

Unit	Datatype
%	xsd:integer
1/h	xsd:decimal
1/min	xsd:decimal
degC	xsd:integer
dm3	xsd:decimal
dm3/s	xsd:decimal
h	xsd:integer
m	xsd:decimal
m/dag	xsd:decimal
m2	xsd:decimal
m3/h	xsd:decimal
mm	xsd:integer
mm/h	xsd:decimal
mm/min	xsd:decimal
yyyymmdd	xsd:date
yyyy	xsd:gYear
pcs	xsd:integer
ppm	xsd:integer
hhmmss	xsd:time
-	xsd:decimal
number	xsd:integer

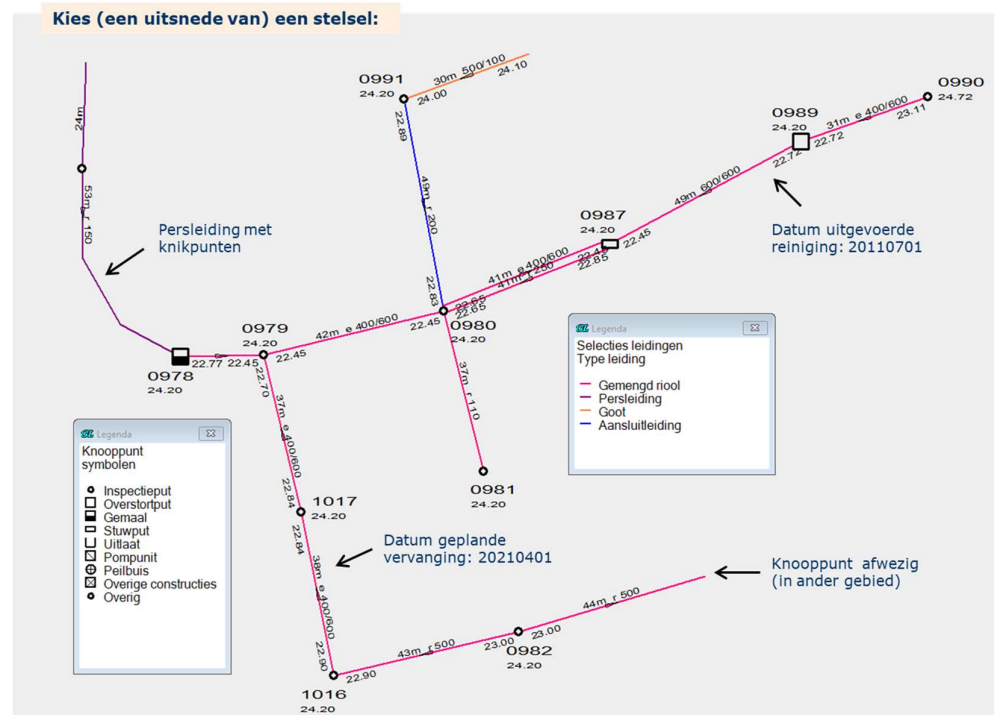
5 Dataset conform GWSW-Basis: Gestructureerd opbouwen

Op basis van een eerdere presentatie (uit de “Gellish-tijd”, december 2013) bouwen we in de volgende hoofdstukken stapsgewijs een dataset.

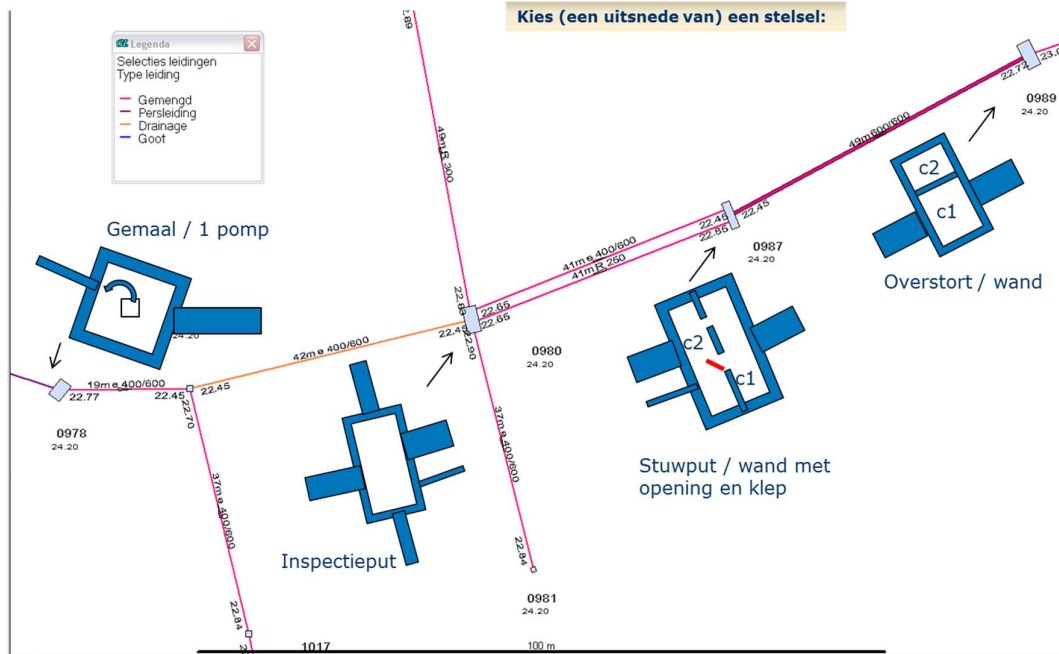
5.1 Het voorbeeld-stelsel

In dit document wordt vaak verwezen naar het voorbeeldbestand (zie hoofdstuk Uitgangspunten). In het bestand zijn hiervoor opmerkingen toegevoegd die altijd het formaat “#OpmNN: Xxxxx” hebben. NN is het optionele volgnummer (kan ontbreken), Xxxxx is de opmerking.

Het volgende schema representeert het stelsel in het voorbeeldbestand:



De detailconstructies:



5.2 Structureren van het GWSW-OroX-bestand

Vaak is de brondatabase ontworpen als relationele database met aparte tabellen voor de stelsels, putten, leidingen enzovoort. Zie ook de toelichting in hoofdstuk 3. Zo'n opbouw wijkt sterk af van de GWSW-OroX-structuur.

GWSW_Lijsten is daarom een belangrijk handvat voor het (geautomatiseerde) ontwerp van de omzettingsssoftware. Een voorbeeld van zo'n aanpak in een aantal stappen:

1. Selecteer in de brondatabase de **stelsels** als hoogste compositieobject.
 - 1.1. Gebruik GWSW_Lijsten/Soortenboom om per stelsel het juiste type te bepalen. Zoek daarbij het **bladerobject**, het meest specialiseerde type (zonder relevante subtypes, zie hst 5.3). Maak daarvan een mapping-tabel.
2. Herhaal stap 1.1 voor de te onderscheiden types van **putten** en **leidingen**, bepaal de meest gespecialiseerde types en maak een mapping-tabel.
3. Herhaal stap 1.1 voor de te onderscheiden **kenmerken** bij de stelsels, putten en leidingen (de "kolommen" van een databasetabel) en maak een mapping-tabel.
4. Vervolgens per stelsel in de brondatabase:
 - 4.1. Geef het stelsel een individual-naam en vermeld het type met de **rdf:type** relatie in het GWSW-OroX (zie hst 5.3)
 - 4.2. Plaats de naam van het stelsel met de **rdfs:label** relatie in het GWSW-OroX (zie hst 4.3 en 5.3)
 - 4.3. Zoek in GWSW_Lijsten/Relaties de **hasAspect** relaties (1:n niveaus). Zoek de relevante kenmerktypes op basis van de mapping-tabel uit stap 3.
 - 4.3.1. Vermeld het **type** en de **waarde** van de **kenmerken** in het GWSW-OroX (zie hst 4.3 en 5.5)
 - 4.4. Zoek in GWSW_Lijsten/Relaties de **hasPart** relaties (1 niveau) bij het stelseltype
 - 4.5. Selecteer de **hasPart** relatie met de eerder (in stap 2) bepaalde bladerobjecten van **putten**
 - 4.6. Vervolgens per gevonden **puttype** :

- 4.6.1. Doorloop de brondatabase op de putten binnen het stelsel
- 4.6.2. Doorloop de stappen 4.1 t/m 4.4 voor de put (“stelsel” wordt “put”)
- 4.6.3. Doorloop eventueel ook het equivalent van stap 4.5, zoek de **hasPart** relatie (1 niveau) met constructieonderdelen en compartimenten.
- 4.7. Selecteer de **hasPart** relatie met de eerder (in stap 2) bepaalde bladerobjecten van **leidingen**
- 4.8. Vervolgens per gevonden **leidingtype** :
 - 4.8.1. Doorloop de brondatabase op de leidingen binnen het stelsel
 - 4.8.2. Doorloop de stappen 4.1 t/m 4.4 voor de leiding (“stelsel” wordt “leiding”)

Met deze stappen is het grootste deel van de gegevens al omgezet naar een GWSW-OroX.

Zoals genoemd beschrijft de GWSW Ontologie de relaties tussen alle relevante entiteiten (fysieke objecten, activiteiten en ruimtes) in het stedelijk water beheer. Bij de omzetting van de brondatabase naar het GWSW-OroX zijn daarom een aantal specifieke bewerkingen nodig. Deze zijn beschreven in de volgende paragrafen.

5.2.1 Locatiegegevens / ruimtes

Locatiegegevens zoals straatnaam, wijknaam zijn in brondatabases vaak als kenmerk gedefinieerd. In het GWSW worden deze locaties als concepten van het type “ruimte” beschouwd. In GWSW_Lijsten/Relaties hebben “ruimtes” een **hasPart** relatie met putten, leidingen, stelsel. Zie voor de details hst 5.4.

5.2.2 Activiteiten

In het GWSW zijn geplande en uitgevoerde activiteiten concepten die een fysiek object als input hebben. Als in de brondatabase een leiding het kenmerk “datum inspectie” heeft dan wordt in het GWSW-OroX de inspectie als apart concept (van het type Activiteit) beschreven. In GWSW_Lijsten/Relaties hebben activiteiten een **hasAspect** relatie met bijvoorbeeld de “datum van uitvoering” en maximaal één relatie **hasInput** met de betreffende put of leiding. Zie voor de details hst 5.4.

5.2.3 Netwerk / topologie

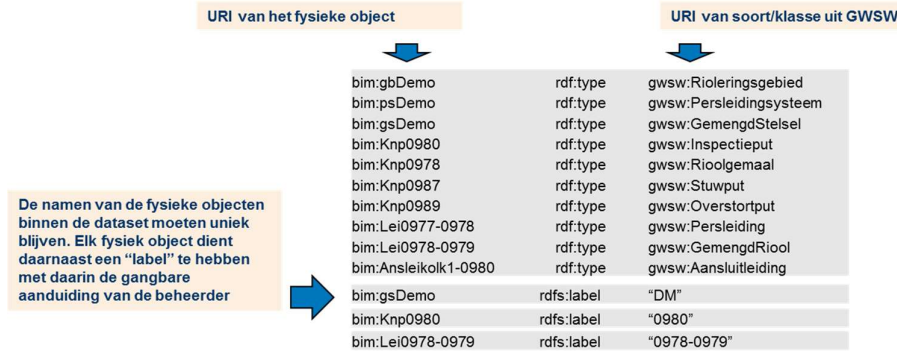
De topologie per put of leiding is als kenmerk (**hasAspect**) van het type **TopologischElement** bij putten en leidingen opgenomen en komen dus aan bod in stap 4.3. Verbindingen tussen putten en leidingen (het “netwerk”) worden altijd via kenmerken van het type **TopologischElement** beschreven. Doorloop daarvoor GWSW_Lijsten/Relaties op de relatie **hasConnection** (1 niveau). Zie voor de details hst 5.6.

5.3 Classificeren: indelen in soorten

Zie ook #Opm04 in voorbeeldbestand.

Elk fysiek object krijgt een unieke naam (URI) en wordt geclassificeerd: wat voor soort object het is. Het GWSW beschrijft deze soorten eenduidig.

Een dataset bevat de beschrijving van een fysiek stelsel. De ORO feiten zijn onderverdeeld in een aantal hoofdgroepen



5.3.1 Detailniveau soortenboom

De soortenboom van het GWSW wordt via een globaal niveau (bijvoorbeeld soort "Fysiek object") naar een gedetailleerd niveau (bijvoorbeeld soort "Trottoirkolk" is van het type "Fysiek object") opgebouwd. Het zal duidelijk zijn dat voldoende detail-niveau nodig is voor een bruikbare dataset. In de volgende tabellen is voor de belangrijkste soorten het gewenste detail-niveau aangegeven. Meer detailniveau (bijv. "Trottoirkolk" in plaats van "Kolk") dan in de tabellen opgenomen kan altijd, minder detail is ongewenst.

Bladerobjecten

Specificeer instanties van concepten altijd zoveel mogelijk. Gebruik het subtype als dat bekend is.

Dat lijkt soms overbodig, een "vrijverval leiding" als deel van een gemengd stelsel specificeert de leiding impliciet immers al voldoende. Maar bij het hanteren van bladerobjecten kan bij ondervragen (SPARQL) van de database zowel de specialisatie ("bladerobject") als het supertype (via inferencing) direct benaderd worden.

Het GWSW kent zogenoemde conformiteitsklassen, die specificeren de kwaliteitseisen voor een bepaalde toepassing. Per conformiteitsklasse kan ook het benodigde detailniveau van de typering variëren. In de volgende tabellen staan de meest gebruikte type-vereisten.

Te gebruiken stelseltypes

Type stelsel	Supertype
Gemengd stelsel	Vrijverval rioolstelsel
Hemelwaterstelsel	Vrijverval rioolstelsel
Vuilwaterstelsel	Vrijverval rioolstelsel
Drukriolering	Mechanisch rioolstelsel
VacuüMRIOLERING	Mechanisch rioolstelsel
Mechanisch drainageselsel	Drainageselsel
Vrijverval drainageselsel	Drainageselsel
Persleidingsstelsel	Transportstelsel

Vrijverval transportstelsel	Transportstelsel
-----------------------------	------------------

Te gebruiken leidingtypes

Type	Supertype
Drain	Leiding
Duiker	Leiding
Mantelbuis	Leiding
Aansluitleiding	Rioolleiding
Drukleiding	Rioolleiding
Vacuümleiding	Rioolleiding
Vuilwaterriool	Vrijverval rioolleiding
Hemelwaterriool	Vrijverval rioolleiding
Infiltratieriool	Vrijverval rioolleiding
Overstortleiding	Vrijverval rioolleiding
Stuwrioolleiding	Vrijverval rioolleiding
Zinker	Vrijverval rioolleiding
Bergingsleiding	Vrijverval rioolleiding
Bergbezinkleiding	Vrijverval rioolleiding
Persleiding	Mechanische transportleiding
Spoelleiding	Mechanische transportleiding
Transportrioolleiding	Vrijverval transportleiding
Goot	Open leiding

Te gebruiken puttypes

Type	Supertype
Aansluitput	Put
Drainageput	Put
Filterput	Put
Slokop	Put
Beerput	Put
Infiltratieput	Put
Kolk	Put
Aansluitput	Put
Doorspoelput	Rioolput
Inspectieput	Rioolput
Kruisingsput	Rioolput
Zinkerput	Rioolput
Stuwput	Rioolput
Lozingsput	Rioolput
Externe overstortput	Overstortput
Interne overstortput	Overstortput

Pompunit	Pompput
Vacuümpompstation	Pompput
Rioolgemaal	Gemaal

Te gebruiken reservoirtypes

Type	Supertype
Bergingsbassin	Reservoir
Bergingsvijver	Reservoir
Bergbezinkbassin	Reservoir
Bezinkbassin	Reservoir
Vacuümpopslagtank	Reservoir
Infiltratiebassin	Infiltratiereservoir
Infiltratiegreppel	Infiltratiereservoir
Infiltratieveld	Infiltratiereservoir
Wadi	Infiltratiereservoir
Helofytenfilter	Zuiveringsreservoir
IBA	Zuiveringsreservoir
Septictank	Zuiveringsreservoir

5.4 Samenstelling: decompositie

Geldt ook voor aansluitingen:

Stelsel bevat aansluiting.

Aansluiting bevat aansluitleidingen, ontstopingsputten, inlaten enzovoort.

Zie ook #Opm03 in voorbeeldbestand.

Specificeer de samenstelling van het stelsel. Eerst de decompositie, welk fysiek object is onderdeel van welk ander fysiek object

URI van het fysieke object		
bim:gbDemo	gws:hasPart	bim:psDemo
bim:psDemo	gws:hasPart	bim:Knp0977
bim:psDemo	gws:hasPart	bim:Lei0977-0978
bim:gbDemo	gws:hasPart	bim:gsDemo
bim:gsDemo	gws:hasPart	bim:Knp0978
bim:gsDemo	gws:hasPart	bim:Lei0978-0979

Zie het vorige blad voor de objectsoorten:

gbDemo = rioleringsgebied

psDemo = persleidingstelsel = onderdeel van gbDemo

gsDemo = gemengd stelsel = onderdeel van gbDemo

bim:Knp0978 = inspectieput = onderdeel van gsDemo

Lei0978-0979 = gemengd riool = onderdeel van gsDemo

De decompositie passen we toe op stelselniveau, maar ook op de opbouw van bijvoorbeeld putconstructies (zie hoofdstuk 5.7 voor de uitwerking):

Zie ook #Opm05 in voorbeeldbestand.

De decompositie kan doorgevoerd worden tot op het niveau van constructieonderdelen

URI van het fysieke object

bim:Knp0978	rdf:type	gsw:Rioolgemaal
bim:Knp0978_cmp1	rdf:type	gsw:Compartment
bim:Knp0978	gsw:hasPart	bim:Knp0978_cmp1
bim:Knp0978_pmp1	rdf:type	gsw:Pomp
bim:Knp0978_cmp1	gsw:hasPart	bim:Knp0978_pmp1

Zie ook de vorige bladen:

bim:Knp0978 = Rioolgemaal

bim:Knp0978 heeft als onderdeel pompkelder bim:Knp0978_cmp1

bim:Knp0978_cmp1 heeft als onderdeel pomp bim:Knp0978_pmp1

5.4.1 Samenstelling: Ruimte / Gebied / Ondergrond

Het GWSW beschrijft ook de samenstelling van concepten van het type Ruimte. Voor deze relaties wordt ook [hasPart](#) gehanteerd. Er hoeft hierbij echter geen sprake te zijn van een compositie zoals in het vorige hoofdstuk.

Een ruimte kan een andere ruimte bevatten maar ook een fysiek object. Daarmee geven we aan dat een stelsel of een leiding in een bepaalde ruimte ligt.

Voorbeelden van het type ruimte zijn:

- Rioleringsgebied
- District
- Kern
- Straat
- Ondergrond

De straat of het district wordt dus niet als [hasAspect](#) opgenomen bij het fysieke object, maar als volgt:

In de GWSW-Ontologie

gsw:Gebied	rdf:type	gsw:Ruimte .
gsw:Rioleringsgebied	rdf:type	gsw:Gebied .
gsw:Straat	rdf:type	gsw:Ruimte .
gsw:Ondergrond	rdf:type	gsw:Ruimte .
enzovoort		

In de dataset:

Zie ook #Opm02 in voorbeeldbestand.

bim:RiolGebied_A	rdf:type	gws:Rioleringsgebied .
bim:RiolGebied_A	gws:hasPart	bim:Stelsel_A .
bim:Stelsel_A	rdf:type	gws:GemengdStelsel . (het rioleringsgebied is hiermee bij het stelsel bekend)
bim:Straat_A	rdf:type	gws:Straat .
bim:Straat_A	gws:hasAspect	[
	rdf:type	gws:Straatnaam ;
	gws:hasValue	"straatnaam"
]		
bim:Straat_A	gws:hasPart	bim:Lei0989-0990 .
bim:Lei0989-0990	rdf:type	gws:Leiding . (straatnaam is hiermee bij de leiding bekend)
bim:Ondergrond_A	rdf:type	gws:Ondergrond .
bim:Ondergrond_A	gws:hasAspect	[
	rdf:type	gws:Grondsoort ;
	gws:hasReference	gws:Klei
]		
bim:Ondergrond_A	gws:hasPart	bim:Stelsel_A . (alle stelsel-leidingen/putten liggen in klei)

5.4.2 Samenstelling: Bestaansvorm / Activiteiten

De samenstelling van deze concepten is als volgt opgezet:

Een bestaansvorm (bijvoorbeeld een organisatie) kan als uitvoer ([hasOutput](#)) of als invoer ([hasInput](#)) een Activiteit hebben. Een activiteit kan een project zijn, in dat geval heeft de bestaansvorm Opdrachtgever als uitvoer het Project (is controller, bestuurder), de Opdrachtnemer heeft als invoer het Project (is uitvoerder).

Het project (de activiteit) kan deelactiviteiten ([hasPart](#)) hebben en als uitvoer ([hasOutput](#)) of als invoer ([hasInput](#)) een Fysiek Object.

In de GWSW-Ontologie

gws:Opdrachtgever	rdf:type	gws:Bestaansvorm .
gws:Opdrachtnemer	rdf:type	gws:Bestaansvorm .
gws:Inspecteur	rdf:type	gws:Opdrachtnemer .
gws:InspectieProject	rdf:type	gws:Activiteit .
gws:InspecterenLeiding	rdf:type	gws:Activiteit . (ingekorte conceptnaam)
gws:InspectieProject	gws:hasPart	gws:InspecterenLeiding .

In de dataset:

De activiteit InspecterenLeiding wordt in de praktijk verder gespecialiseerd tot VisueelInspecterenVrijvalLeiding
Zie ook #Opm06 in voorbeeldbestand.

bim:Opdr_A	rdf:type	gws:Opdrachtgever .
bim:Opdr_A	rdfs:label	"opdrachtgevernaam" .
bim:Insp_A	rdf:type	gws:Inspecteur .
bim:Insp_A	rdfs:label	"inspectiebedrijfsnaam" .
bim:Proj_A	rdf:type	gws:Inspectieproject .
bim:Opdr_A	gws:hasOutput	bim:Proj_A . (opdrachtgever doet aansturing van het project)
bim:Insp_A	gws:hasInput	bim:Proj_A . (opdrachtnemer doet uitvoering van het project)
bim:Proj_A_lei	rdf:type	gws:InspecterenLeiding .
bim:Proj_A	gws:hasPart	bim:Proj_A_lei . (de deelactiviteit binnen het project)
bim:Lei0989-0990	rdf:type	gws:Leiding .
bim:Proj_A_lei	gws:hasInput	bim:Lei0989-0990 . (de input van de deelactiviteit - projectdefinitie)
bim:Proj_A_lei	gws:hasOutput	bim:Lei0989-0990 . (de output van de deelactiviteit - inspectieresultaat)

De registratie van de reinigingsdatum bij een leiding in compacte vorm:

Zie ook #Opm07 in voorbeeldbestand.

```

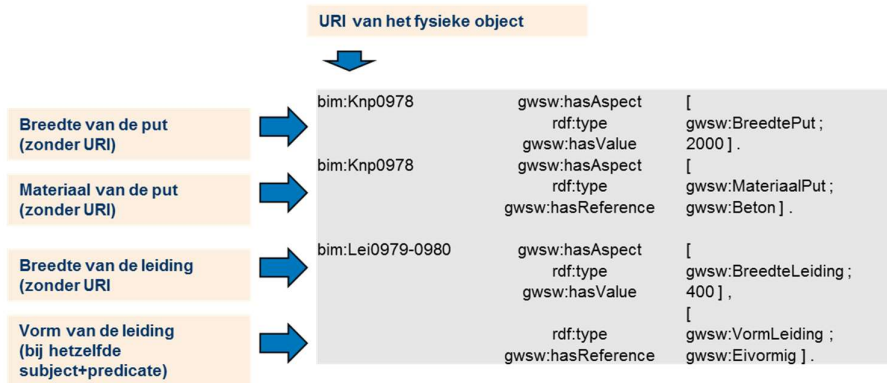
bim:Proj_A_lei      rdf:type      gsw:ReinigenHoge_drukLeiding ; (de activiteit)
                    gsw:hasInput  bim:Lei0989-0990 ;           (de input van de activiteit)
                    gsw:hasAspect
[
  rdf:type      gsw:DatumMaatregel ;
  gsw:hasValue  "2015-01-31"^^xsd:date
].

```

5.5 Specificaties: kenmerken en waarden

Zie ook #Opm10 in voorbeeldbestand.

De objecten zijn geclassificeerd en ingedeeld. Nu de kenmerken van de objecten specificeren



Zie ook hoofdstuk 4.3.2 voor het uitschrijven van kenmerken conform de Turtle-syntax.

Afmetingen

Zowel het kenmerk diameter als de kenmerken breedte/hoogte kunnen bij een put of leiding vermeld worden. In het geval van een rond object kunnen diameter of hoogte+breedte of alleen hoogte of breedte gevuld worden.

Uitwerken: Metagegeven Inwinning, temporeel, bij meervoudige kenmerken

Hoe ermee omgaan, inclusief "ontwerp" status (in GWSW v1.4.1 vervalt "ontwerp bob" enz.)

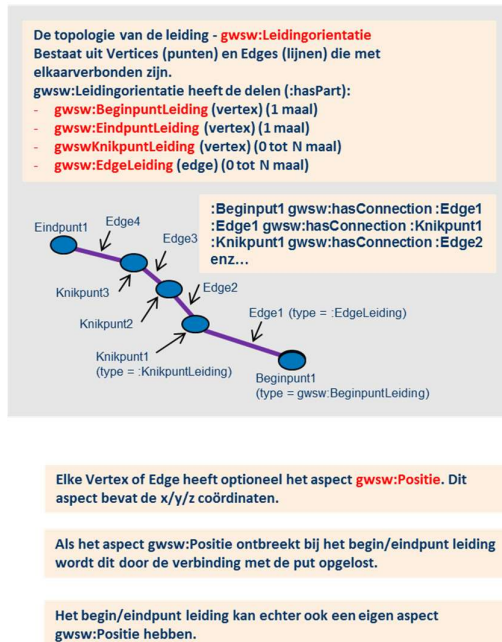
5.6 Samenhang: Topologie en geometrie

5.6.1 Topologie en geometrie in GWSW Ontologie

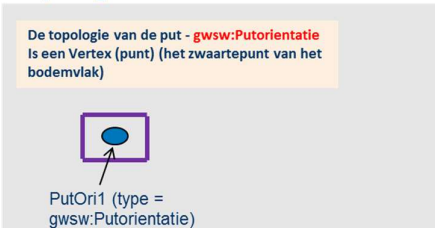
De topologie en geometrie van Stedelijk Water concepten is gedetailleerd in het GWSW gemodelleerd.

De topologie beschrijft de opbouw van concepten en de onderlinge samenhang. De geometrie beschrijft de lokatie en afmetingen van de concepten.

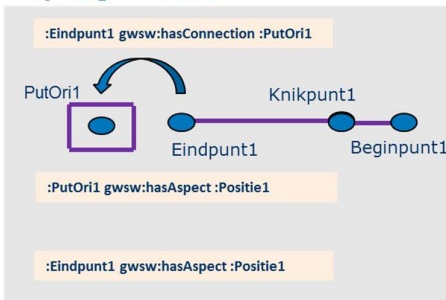
Topologie Leiding



Topologie Put



Topologie Stelsel



Zie ook #Opm08 (3x) in voorbeeldbestand.

Door aan elke vertex vervolgens een positie (x, y, z) te koppelen kan de geometrie van elk GWSW concept in detail worden beschreven.

5.6.2 Notatie Topologie en Geometrie

In het GWSW-OroX is gekozen voor een korte notatie van de topologie en geometrie. De notatie is vooral voor geknikte leidingen (curves) veel compacter en leesbaarder. Daarnaast sluiten we hiermee aan op de GeoSPARQL querytaal. Deze taal is in 2012 uitgebracht door het Open Geospatial Consortium (OGC). GeoSPARQL beschikt over allerlei “spatial” functies, zoals het berekenen van oppervlaktes en afstanden tussen geometrieën. De alternatieve geometrie maakt directe toepassing van GeoSPARQL-functionaliteit op het GWSW-OroX mogelijk.

In de korte GWSW-OroX-notatie wordt het topologische element “edge” (en de verbindingen ermee) niet beschreven:

- In de leiding-topologie worden alleen de vertices BeginpuntLeiding en EindpuntLeiding opgenomen, dat geldt ook voor geknikte leidingen
- De vertices voor knikpunten (via edges verbonden) worden dus niet opgenomen. Knikpunten gelden voor de beschrijving van geknikte leidingen (curves) of buitengrenzen (polygonen). Deze elementen worden impliciet in de geometrie van de leiding beschreven.

5.6.3 Het kenmerk Geometrie

De geometrie voor punten, lijnen en vlakken (putten, leidingen, buitengrenzen) wordt beschreven in één kenmerk met daarin een geometrie-string volgens de GML-definitie. Het gehanteerde coördinaatsysteem is EPSG:7415 (x/y conform EPSG:28992 (=RD), z conform EPSG:5709 (=NAP)).

Geometrie is altijd een kenmerk van de topologische elementen. Er worden drie soorten geometrie-kenmerken (subtypes van **Geometrie**) in het GWSW-OroX gebruikt:

Concept: **gws:Punt** (0-dimensionaal)
 Kenmerk van: **gws:Putorientatie**
 GML-notaties: Met hoofdelement **gml:Point** of **gml:MultiPoint** met één of meer **gml:pos** elementen.

Bijvoorbeeld:

```
<gml:Point xmlns:gml="http://www.opengis.net/gml/3.2"><gml:pos>
168508.00 442706.30
</gml:pos></gml:Point>^^geo:gmlLiteral
```

Concept: **gws:Lijn** (1-dimensionaal)
 Kenmerk van: **gws:Leidingorientatie**
 GML-notatie: Met hoofdelement **gml:LineString**, **gml:MultiLineString**, **gml:Curve** of **gml:MultiCurve** met één of meer **gml:posList** elementen. Bijvoorbeeld:

```
<gml:LineString xmlns:gml="http://www.opengis.net/gml/3.2"> <gml:posList
srsDimension="2">
168508.00 442706.30 168510.00 442716.30 168511.00 442706.30 168512.00
442736.30
</gml:posList></gml:LineString>^^geo:gmlLiteral
```

Concept: **gws:Buitengrens** (2-dimensionaal), subtype van **gws:Vlak**
 Kenmerk van: **gws:Putorientatie**, **gws:Gebiedorientatie**, **gws:Bouwwerorientatie**
 GML-notatie: Met hoofdelement **gml:Polygon**, **gml:MultiPolygon**, **gml:Surface** of **gml:MultiSurface** met één of meer **gml:posList** elementen. Bijvoorbeeld:

```
<gml:Polygon xmlns:gml="http://www.opengis.net/gml/3.2">
<gml:exterior><gml:LinearRing><gml:posList srsDimension="3">
168508.00 442706.30 22.45 168510.00 442716.30 22.45 168511.00 442706.30 22.45
168512.00 442736.30 22.45 168508.00 442706.30 22.45
</gml:posList></gml:LinearRing></gml:exterior></gml:Polygon>^^geo:gmlLiteral
```

Gebruik binnen de geometrie-literal voor de gml-elementen altijd de namespace **gml**: .

Let op de notatie \" voor gebruik van aanhalingstekens binnen een literal, zie hoofdstuk 4.2.1.

Het attribuut **srsDimension** is binnen het element **gml:posList** verplicht. Voor de drie geometrie-types is de z-waarde optioneel. Met het attribuut **srsDimension** wordt aangegeven of de z-waarde is meegegeven (waarde \"2\" = nee, waarde \"3\" = ja). De z-waarde beschrijft in de meeste gevallen de binnenonderkant van het concept, zie voor de details GWSW_Lijsten.

Niveauwaarde / z-waarde

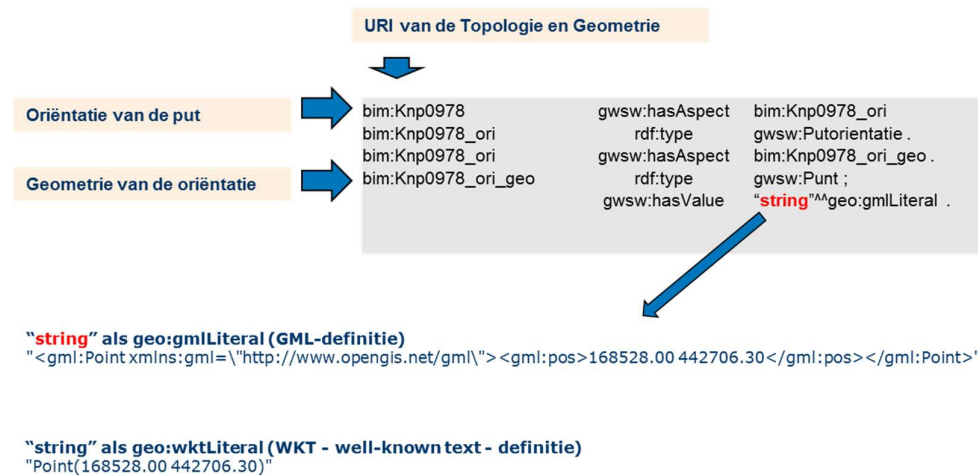
Naast het kenmerk **Geometrie** zijn de in de stedelijk-water-wereld gebruikelijke niveau-aanduidingen als kenmerk bij de topologische elementen opgenomen, bijvoorbeeld:

- **gws:BobBeginpuntLeiding**
- **gws:BobEindpuntLeiding**

Zie GWSW_Lijsten voor de relevante niveauwaarden en het voorbeeld in hoofdstuk 5.4.5. Indien bekend moeten deze niveauwaarden altijd als kenmerk opgenomen worden, optioneel kunnen ze als z-waarde in de GML-notatie meegenomen worden (zie hiervoor). Voor dat laatste kan bijvoorbeeld gekozen worden om het horizontale verloop van een persleiding (ter plaatse van knikpunten) weer te geven.

5.6.4 Put – Topologie en Geometrie

De topologie en geometrie van (subtypes van) :Put is gebaseerd op een vertex (punt)



Zie ook #Opm14 in voorbeeldbestand.

5.6.5 Leiding – Topologie en Geometrie

In native databases komt het veel voor dat de geometrie van een vrijerval-leiding ontbreekt maar wel impliciet beschreven wordt door de verbinding met putten (die wel een geometrie hebben). Voor de presentatie van een parallelleidingen die op dezelfde put aansluiten en voor pers- en drukleidingen blijft de eigen geometrie zo wie zo onontbeerlijk.

In het GWSW-OroX wordt (vanaf versie 1.4) de geometrie van de leiding altijd meegeven, de leiding wordt als apart geo-object beschouwd. Dat verbetert ook de query-efficiëntie aanzienlijk.

De topologie van de leiding beschrijft, los van de geometrie, de koppeling met de putten:

Een leiding verbonden met putten

Topologie put	→	bim:Knp0979_ori bim:Knp0979	rdf:type gsw:hasAspect	gsw:Putorientatie bim:Knp0979_ori
Topologie put	→	bim:Knp0980_ori bim:Knp0980	rdf:type gsw:hasAspect	gsw:Putorientatie bim:Knp0980_ori
Topologie leiding (begin-/eindpunt)	→	bim:Lei0979-0980_ori bim:Lei0979-0980_ori_bgp bim:Lei0979-0980_ori_enp bim:Lei0979-0980 bim:Lei0979-0980_ori bim:Lei0979-0980_ori	rdf:type rdf:type rdf:type gsw:hasAspect gsw:hasPart gsw:hasPart	gsw:Leidingorientatie gsw:BeginpuntLeiding gsw:EindpuntLeiding bim:Lei0979-0980_ori bim:Lei0979-0980_ori_bgp bim:Lei0979-0980_ori_enp
Verbinding leiding met putten	→	bim:Lei0979-0980_ori_bgp bim:Lei0979-0980_ori_enp	gsw:hasConnection gsw:hasConnection	bim:Knp0979_ori bim:Knp0980_ori

Bij de relatie gsw:hasConnection is de rol subject / object (links / rechts) niet voorgeschreven

Zie ook #Opm08 in voorbeeldbestand.

De hoogteligging van de leiding:

De hoogteligging van een leiding wordt als kenmerk aan de topologie gekoppeld

	→	URI van de Topologie	
Oriëntatie / beginpunt	→	bim:Lei0979-0980_ori_bgp	gsw:hasAspect rdf:type
B.o.b. beginpunt	→	bim:Lei0979-0980_ori_enp	gsw:hasAspect rdf:type
			[gsw:BobBeginpuntLeiding; 22.45], [gsw:BobEindpuntLeiding; 22.46] ,

Zie ook #Opm12 in voorbeeldbestand.

Het kenmerk geometrie wordt toegevoegd:

De leidingtopologie blijft gelijk, de geometrie met knikpunten wordt toegevoegd	→	URI van de Topologie	
Oriëntatie leiding	→	bim:Lei0979-0980_ori	gsw:hasAspect rdf:type
Geometrie van de oriëntatie	→		gsw:hasValue [gsw:Lijn ; "string"^^geo:gmlLiteral] .

"string" als geo:gmlLiteral (GML-definitie)

"<gml:LineString xmlns:gml="http://www.opengis.net/gml/"><gml:posList srsDimension="2">168508.00 442706.30 168510.00 442716.30 168511.00 442706.30 168512.00 442736.30</gml:posList></gml:LineString>"

Zie ook #Opm13 in voorbeeldbestand.

5.6.6 Samenhang: grondwaterstand en maaiveldhoogte

Als er bij een individual naar een grondwaterpeil of maaiveldhoogte verwezen wordt, gebeurt dit ook via een topologische beschrijving:

Zie ook #Opm09 in voorbeeldbestand.

Ook de verwijzing naar externe geometrie-objecten wordt via de topologie beschreven

Topologie maaiveld	→	bim:Maaiveld_A	rdf:type	gsw:Maaiveldorientatie
Kenmerk en waarde maaiveldhoogte	→	bim:Maaiveld_A	gsw:hasAspect	bim:Maaiveld_A_hgt
	→	bim:Maaiveld_A_hgt	rdf:type	gsw:Maaiveldhoogte
	→	bim:Maaiveld_A_hgt	gsw:hasValue	24.20
Verbinden aan topologie put	→	bim:Maaiveld_A	gsw:hasConnection	bim:Knp0980_ori
Idem voor grondwaterstand. Te koppelen aan meerdere putten (geldt ook voor maaiveld)	→	bim:GwsPunt_A	rdf:type	gsw:Grondwatermeetpunt
	→	bim:GwsPunt_A	gsw:hasAspect	bim:GwsPunt_A_Niv
	→	bim:GwsPunt_A_Niv	rdf:type	gsw:Grondwaterniveau
	→	bim:GwsPunt_A_Niv	gsw:hasValue	22.10
	→	bim:GwsPunt_A	gsw:hasConnection	bim:Knp0979_ori
	→	bim:GwsPunt_A	gsw:hasConnection	bim:Knp0980_ori

5.7 Hydraulische constructies (conformiteitsklasse GWSW-Hyd)

Een GWSW-OroX dataset kan alle netwerk componenten voor hydraulische modellering bevatten. De module GWSW-Basis versie 1.4 bevat enkele toevoegingen op dit vlak. De conformiteitsklasse GWSW-Hyd bevat de kwaliteitseisen voor de gegevens en is onderdeel van GWSW-Basis.

In dit hoofdstuk werken we een aantal voorbeelden uit van hydraulische constructies. Voor deze constructies zoals overstortputten en rioolgemalen hanteren we de term Kunstwerken in de toelichting (gebruikelijk in de modelleringswereld).

5.7.1 Netwerk: knooppunten en verbindingen

Putten, leidingen en andere constructies wordt voor een modelberekening geschematiseerd naar twee topologische elementen:

- **Knooppunt:** koppelt één of meerdere verbindingen
- **Verbindingen:** geleidt het afvalwater en heeft een stromingsrichting

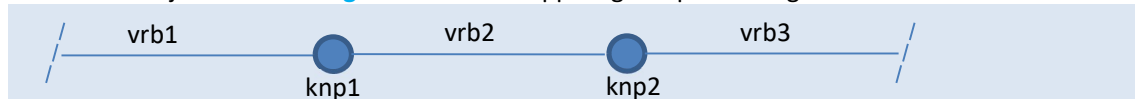
1. Voor een **Inspectieput** met aansluitende vrijverval leidingen is dat helder:



Verbindingen vrb1, vrb2, vrb3 beschrijven de **Vrijverval leiding**

Knooppunten knp1, knp2 beschrijven de **Inspectieput**

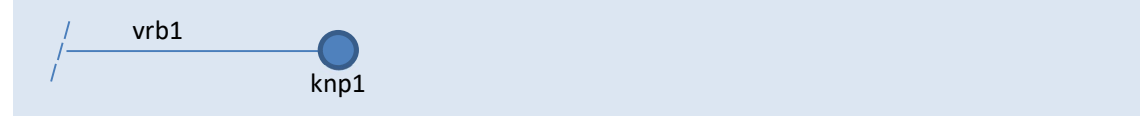
2. Hetzelfde bij een **Verbindingsstuk** voor de koppeling van persleidingen:



Verbindingen vrb1, vrb2, vrb3 beschrijven de **Persleiding**

Knooppunten knp1, knp2 beschrijven het **Verbindingsstuk**

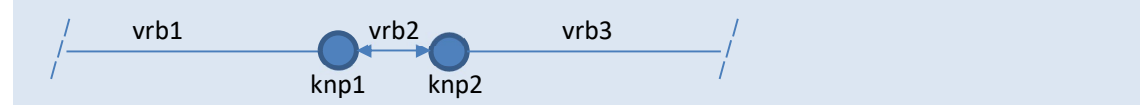
3. Een **Uitlaatpunt** wordt als volgt geschematiseerd:



Verbinding vrb1 beschrijft de **Leiding** (vrijverval of mechanisch)

Knooppunt knp1 beschrijft het **Uitlaatpunt**

4. Een **Pomp**, **Wand** of **Doorlaat** wordt als volgt geschematiseerd:

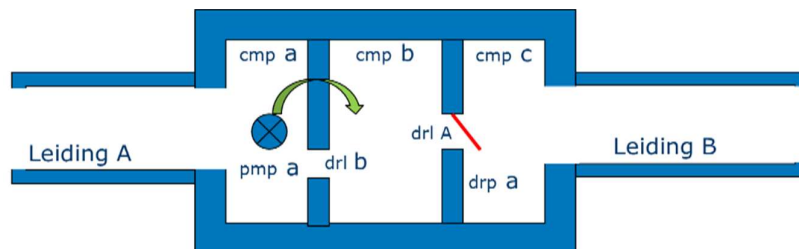


Verbinding vrb1 en vrb3 beschrijven de **Leiding**

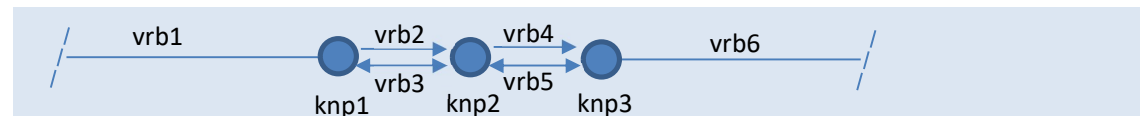
Verbinding vrb2 beschrijft de **Pomp**, **Wand** of **Doorlaat** (met een stromingsrichting)

Knooppunten knp1 en knp2 beschrijven (delen van) het **Kunstwerk**.

5. Meerdere **Pompen**, **Wanden** en **Doorlaten** in een kunstwerk worden als volgt geschematiseerd:



Deze constructie wordt:



Verbinding 1 en 6 beschrijven de **Leiding**, gekoppeld aan knp1 en knp3

Verbinding 2 beschrijft de **Pomp** (stroming in één richting), gekoppeld aan knp1 en knp2

Verbinding 3 beschrijft de **Doorlaat** (stroming in beide richtingen), gekoppeld aan knp1 en knp2

Verbinding 4 beschrijft de **Doorlaat** (stroming in één richting), gekoppeld aan knp2 en knp3

Verbinding 5 beschrijft de **Wand** (stroming in beide richtingen), gekoppeld aan knp2 en knp3

Knooppunt 1, 2 en 3 beschrijven (delen van) het **Kunstwerk**.

5.7.2 Netwerk: Topologie en geometrie

Knooppunten - zijn 0 dimensionaal

Knooppunt is een punt. De volgende GWSW-concepten worden beschreven door knooppunten:

- **Put** (met alle subtypes: inspectieput, infiltratieput, ...)
 - Heeft topologisch kenmerk **Putoriëntatie** (= subtype van Knooppunt)
- **Bouwwerk** (met alle subtypes: rioolgemaal, uitlaatconstructie, ...)
 - Heeft topologisch kenmerk **Bouwwerkoriëntatie** (= subtype van Knooppunt)
- **Compartment** (met subtype: natte pompkamer)
 - Heeft topologisch kenmerk **Compartmentoriëntatie** (= subtype van Knooppunt)
- **Aansluitpunt** (met alle subtypes: lozingspunt, uitlaatpunt, overnamepunt, ...)
 - Is topologisch element (= subtype van Knooppunt)
- **Hulpstuk** (met alle subtypes: verbindingsstuk, afsluitstuk, ...)
 - Hebben topologisch kenmerk **Hulpstukoriëntatie** (= subtype van Knooppunt)

Verbindingen - zijn 1 dimensionaal

Verbinding is een lijn en heeft als onderdelen het **Beginpunt** en **Eindpunt**. De volgende GWSW-concepten worden beschreven door verbindingen:

- **Leiding** (met alle subtypes)
 - Heeft topologisch kenmerk **Leidingoriëntatie** (= subtype van Verbinding)
- **Wand** (met alle subtypes: overstortdrempel, schildmuur, stuwmuur)
 - Heeft topologisch kenmerk **Onderdeeleriëntatie** (= subtype van Verbinding)
- **Doorlaat** (met alle subtypes: Opening in wand, Wervelventiel, Afsluiter, Klep)
 - Heeft topologisch kenmerk **Onderdeeleriëntatie** (= subtype van Verbinding)
- **Pomp** (heeft geen subtypes)
 - Heeft topologisch kenmerk **Onderdeeleriëntatie** (= subtype van Verbinding)

Netwerk: koppeling van Knooppunten aan Verbindingen

De verbindingen zijn altijd via het begin- en/of eindpunt verbonden aan de knooppunten. Daarmee wordt de topologie van het netwerk in de eerder genoemde voorbeelden volledig beschreven.

Alle typen verbindingen (die een **Leiding**, **Wand**, **Doorlaat** of **Pomp** beschrijven) kunnen koppelen aan alle typen knooppunten (die een **Put**, **Bouwwerk**, **Compartiment**, **Aansluitpunt** of **Hulpstuk** beschrijven).

Als een verbinding een externe lozing heeft, bijvoorbeeld een pomp in een eindgemaal waarbij het verpompte water uit het systeem verdwijnt, zal het gekoppelde knooppunt een **Uitlaatpunt** (een subtype van Aansluitpunt) zijn.

Kunstwerken (overstortputten, stuwputten, rioolgemalen, ...)

De constructies die een **Wand**, **Doorlaat** of **Pomp** bevatten hoeven voor de netwerkberekening niet gemodelleerd te worden. Dus een overstortput kan beschreven door alleen de compartimenten te benoemen, die vervolgens verbonden zijn via de overstortdrempel.

Ook in het GWSW-OroX is de benoeming van deze constructie geen noodzaak. In veel native databases worden kunstwerken als een koppeling van knooppunten beschreven. In zo'n geval zullen de gekoppelde knooppunten veelal van het type "compartiment" zijn.

Als de kunstwerken wel in het GWSW-OroX zijn opgenomen kan er semantiek worden toegevoegd zoals stelselnaam, maaiveldhoogte en natuurlijk type constructie. De verbindingen **Wand**, **Doorlaat**, **Pomp** en het knooppunt **Compartiment** worden dan in het GWSW-OroX een deel van het kunstwerk.

Een **Uitlaatpunt** is in het GWSW-OroX ook een (topologisch) kenmerk van de uitlaatconstructie.

5.7.3 Hydraulische constructies

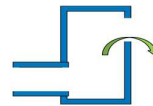
De principes zijn in de vorige hoofdstukken beschreven. De inrichting van het GWSW-OroX leunt sterk op de eerdere behandelde onderwerpen Decompositie en Topologie (zie hoofdstuk 5.4 en 5.6). Hierna volgen een aantal voorbeelden, voornamelijk gebaseerd op het Didactisch Stelsel v2.0. Zie ook https://apps.gwsw.nl/item_hyddef.

5.7.3.1 Overstortput:



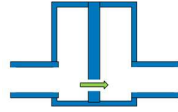
De overstortput bestaat uit 2 Compartimenten en een Wand <i>Deze omvattende constructie optioneel opnemen</i>	→	bim:Overstortput_331	rdf:type	gwsw:Overstortput ;
			gwsw:hasPart	bim:Compartiment_334 ;
			gwsw:hasPart	bim:Compartiment_335 ;
			gwsw:hasPart	bim:Overstortdrempel_336 .
		bim:Compartiment_334	rdf:type	gwsw:Compartiment .
		bim:Compartiment_335	rdf:type	gwsw:Compartiment .
		bim:Overstortdrempel_336	rdf:type	gwsw:Overstortdrempel .
De Compartimenten zijn Knooppunten: topologie = punt	→	bim:Compartiment_334	gwsw:hasAspect	bim:Compartiment_334_ori .
		bim:Compartiment_335	gwsw:hasAspect	bim:Compartiment_335_ori .
		bim:Overstortdrempel_336	gwsw:hasAspect	
De Wand is Verbinding: topologie = beginpunt / eindpunt	→	[rdf:type	gwsw:Onderdeelorientatie ;
			gwsw:hasPart	_:BeginpuntOnderdeel_338 ;
			gwsw:hasPart	_:EindpuntOnderdeel_339 ;
];		
De Wand heeft kenmerken zoals drempelbreedte	→	[gwsw:hasAspect	
			rdf:type	gwsw:Dempelbreedte ;
			gwsw:hasValue	5000 ;
].		
De Verbinding "Wand" is gekoppeld aan de compartimenten.	→	_:BeginpuntOnderdeel_338	rdf:type	gwsw:BeginpuntOnderdeel .
		_:EindpuntOnderdeel_339	gwsw:hasConnection	bim:Compartiment_334_ori .
			rdf:type	gwsw:EindpuntOnderdeel .
			gwsw:hasConnection	bim:Compartiment_335_ori .
De aansluitende leidingen zijn vervolgens aan compartimenten gekoppeld				

5.7.3.2 Overstortput met externe lozing:



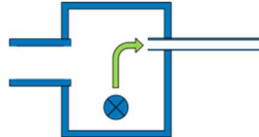
De overstortput bevat alleen een Wand	→	bim:Overstortput_331	rdf:type	gwsw:Overstortput ;
			gwsw:hasPart	bim:Overstortdrempel_336 .
De putorientatie is het Knooppunt: topologie = punt	→	bim:Overstortput_331	gwsw:hasAspect	bim:Overstortput_331_ori
		bim:Overstortput_331_ori	rdf:type	gwsw:Putorientatie .
		bim:Overstortput_331_ori_geo	gwsw:hasAspect	bim:Overstortput_331_ori_geo .
		bim:Overstortput_331_ori_geo	rdf:type	gwsw:Put ;
Het uitlaatpunt definieert de locatie van externe lozing	→	bim:UitlaatPunt_335	gwsw:hasValue	"string"^^geo:gmlLiteral .
		bim:Overstortdrempel_336	rdf:type	gwsw:UitlaatPunt .
		bim:Overstortdrempel_336	rdf:type	gwsw:Overstortdrempel .
De Wand is Verbinding: topologie = beginpunt / eindpunt	→	[gwsw:hasAspect	
			rdf:type	gwsw:Onderdeelorientatie ;
			gwsw:hasPart	_:BeginpuntOnderdeel_338 ;
			gwsw:hasPart	_:EindpuntOnderdeel_339 ;
];		
De Wand heeft kenmerken zoals drempelbreedte	→	[gwsw:hasAspect	
			rdf:type	gwsw:Dempelbreedte ;
			gwsw:hasValue	5000 ;
].		
De Verbinding "Wand" is gekoppeld aan de put en het uitlaatpunt	→	_:BeginpuntOnderdeel_338	rdf:type	gwsw:BeginpuntOnderdeel .
		_:EindpuntOnderdeel_339	gwsw:hasConnection	bim:Overstortput_331_ori .
			rdf:type	gwsw:EindpuntOnderdeel .
			gwsw:hasConnection	bim:UitlaatPunt_335 .

5.7.3.3 Stuwput:



De Stuwput bestaat uit 2 Compartimenten en een Doorlaat <i>Deze omvattende constructie optioneel opnemen</i>	→	bim:Stuwput_126	rdf:type	gsw:Stuwput ;
			gsw:hasPart	bim:Compartiment_129 ;
			gsw:hasPart	bim:Compartiment_130 ;
			gsw:hasPart	bim:Doorlaat_131 .
De Compartimenten zijn Knooppunten: topologie = punt	→	bim:Compartiment_129	rdf:type	gsw:Compartiment .
		bim:Compartiment_130	rdf:type	gsw:Compartiment .
		bim:Doorlaat_131	rdf:type	gsw:Doorlaat .
De Doorlaat heeft kenmerken zoals doorlaatsniveau	→	bim:Compartiment_129	gsw:hasAspect	_:Compartiment_129_ori .
		bim:Compartiment_130	gsw:hasAspect	_:Compartiment_130_ori .
		bim:Doorlaat_131	gsw:hasAspect	gsw:Doorlaatsniveau ;
			rdf:type	gsw:Doorlaatsniveau ;
			gsw:hasValue	0.00 ;
De Doorlaat is Verbinding: topologie = beginpunt / eindpunt	→	bim:Doorlaat_131	gsw:hasAspect	gsw:Onderdeelorientatie ;
			rdf:type	gsw:Onderdeelorientatie ;
			gsw:hasPart	_:BeginpuntOnderdeel_134 ;
			gsw:hasPart	_:EindpuntOnderdeel_135 ;
De Verbinding "Doorlaat" is gekoppeld aan de compartimenten.	→	_:BeginpuntOnderdeel_134	rdf:type	gsw:BeginpuntOnderdeel .
		_:EindpuntOnderdeel_135	gsw:hasConnection	_:Compartiment_129_ori .
			rdf:type	gsw:EindpuntOnderdeel .
			gsw:hasConnection	_:Compartiment_130_ori .

5.7.3.4 Riolgemaal met persleiding:



Het Riolgemaal bevat een Pomp gekoppeld aan een Persleiding	→	bim:Riolgemaal_77	rdf:type	gsw:Riolgemaal ;
			gsw:hasPart	bim:Pomp_85 .
De Bouwwerkorientatie is een Knooppunt: topologie = punt	→	bim:Pomp_85	rdf:type	gsw:Pomp .
		bim:Riolgemaal_77	gsw:hasAspect	_:Bouwwerk_81_ori .
		_:Bouwwerk_81_ori	rdf:type	gsw:Bouwwerkorientatie .
De Pomp is Verbinding: topologie = beginpunt / eindpunt	→	bim:Pomp_85	gsw:hasAspect	gsw:Onderdeelorientatie ;
			rdf:type	gsw:Onderdeelorientatie ;
			gsw:hasPart	_:BeginpuntOnderdeel_87 ;
			gsw:hasPart	_:EindpuntOnderdeel_88 ;
De Pomp heeft kenmerken zoals aanslagniveau	→		gsw:hasAspect	gsw:AanslagniveauBovenstrooms ;
			rdf:type	gsw:AanslagniveauBovenstrooms ;
			gsw:hasValue	0.50 ;
De Persleiding is Verbinding: topologie = beginpunt / eindpunt	→	bim:Persleiding_89	rdf:type	gsw:Persleiding ;
			gsw:hasAspect	gsw:Leidingorientatie ;
			rdf:type	gsw:Leidingorientatie ;
			gsw:hasPart	_:BeginpuntLeiding_90 ;
			gsw:hasPart	_:EindpuntLeiding_91 ;
Het beginpunt van de Persleiding is verbonden met een Verbindingsstuk	→	bim:Verbindingsstuk_92	rdf:type	gsw:Verbindingsstuk ;
		_:Verbindingsstuk_92_ori .	gsw:hasAspect	_:Verbindingsstuk_92_ori .
			rdf:type	gsw:Hulpstukorientatie ;
De Verbinding "Pomp" is gekoppeld aan het Riolgemaal en het Verbindingsstuk	→	_:BeginpuntOnderdeel_87	gsw:hasConnection	_:BeginpuntLeiding_90 .
		_:EindpuntOnderdeel_88	rdf:type	gsw:BeginpuntOnderdeel ;
			gsw:hasConnection	_:Bouwwerk_81_ori .
			rdf:type	gsw:EindpuntOnderdeel ;
			gsw:hasConnection	_:Verbindingsstuk_92_ori .

5.7.3.5 Uitlaatconstructie:

De Uitlaatconstructie heeft als kenmerk het Uitlaatpunt. Het Uitlaatpunt is een Vertex, met een Punt-geometrie.



Leidingen zijn vervolgens aan het Uitlaatpunt gekoppeld



```
bim:Uitlaatconstructie_65      rdf:type      gsw:Uitlaatconstructie ;
                                gsw:hasAspect  _:UitlaatPunt_69 .
_:UitlaatPunt_69              rdf:type      gsw:UitlaatPunt ;
                                gsw:hasAspect
[
                                rdf:type      gsw:Punt ;
                                gsw:hasValue  "string"^^geo:gmlLiteral .
] .
_:EindpuntLeiding_655         gsw:hasConnection  _:UitlaatPunt_69 .
```

"string" als geo:gmlLiteral (GML-definitie)

```
"<gml:Point xmlns:gml="http://www.opengis.net/gml"><gml:pos>400.00 500.00 -1.00</gml:pos>
</gml:Point>"^^geo:gmlLiteral .
```

De Z-waarde van de Uitlaatpunt-geometrie definieert de buitenwaterstand, het lozingsniveau (in dit geval -1.00 m NAP)

Uitlaatpunt is een zelfstandig knooppunt binnen het netwerk

Bij elke vorm van uitlaat is de aanwezigheid van een uitlaatpunt bepalend voor de vaststelling van het hydraulische netwerk. Een uitlaatpunt (met supertype Knooppunt) bestaat altijd zelfstandig in het netwerk, anders dan bijvoorbeeld een bouwwerkoriëntatie hoeft het geen kenmerk van een object te zijn. Als een leiding koppelt aan een uitlaatpunt zonder bijbehorende constructie (uitlaatconstructie, overstortput) dan heeft de leiding vrije uitstroming bij het gekoppelde begin- of eindpunt. Geef in de dataset zo'n uitlaatpunt wel een naam (*rdfs:label*), die wordt gebruikt in de HydX-export. Geef bij voorkeur elk uitlaatpunt een naam, ook als het een kenmerk van een uitlaatconstructie is.

5.7.3.6 Bergbezinkbassin:



De randvoorziening heeft een leegloopvoorziening (van compartiment 2 naar 1, ronde buis \varnothing 200 mm met terugslagklep in compartiment 1). Compartiment 3 heeft een vrije lozing naar het buitenwater. Waterstand NAP -2,20m. In compartiment 2 is op 1,00 m voor overstortdrempel 2 een duikschot aangebracht

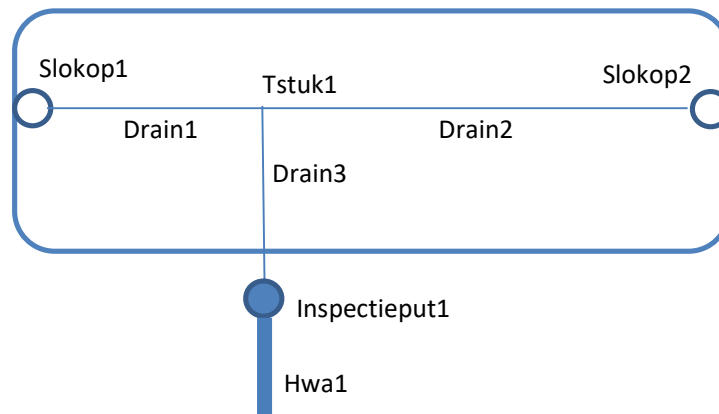
Bouwwerken met meerdere hydraulische elementen zoals dit Bergbezinkbassin bouwen we op uit:

- Compartimenten
- Stuwconstructies
- Overstortconstructies

Het bassin bestaat uit een stuwconstructie, een overstortconstructie en drie compartimenten	→	bim:Bassin1	rdf:type	gsw:Bergbezinkbassin ;
			gsw:hasPart	bim:Cmp1 ;
			gsw:hasPart	bim:Cmp2 ;
			gsw:hasPart	bim:Cmp3 ;
			gsw:hasPart	bim:StuwCon1 ;
			gsw:hasPart	bim:OvsCon1 .
De Compartimenten zijn Knooppunten: topologie = punt	→	bim:Cmp1	rdf:type	gsw:Compartiment .
		bim:Cmp2	rdf:type	gsw:Compartiment .
		bim:Cmp3	rdf:type	gsw:Compartiment .
De stuwconstructie bestaat uit een wand en een doorlaat. ENZOVOORT: de stuwconstructie modelleren zoals Stuwput	→	bim:StuwCon1	rdf:type	gsw:Stuwconstructie .
		bim:OvsCon1	rdf:type	gsw:Overstortconstructie .
		bim:StuwCon1	gsw:hasPart	bim:Drp1 ;
		bim:Drp1	rdf:type	bim:Dr1 .
		bim:Dr1	rdf:type	gsw:Stuwmuur .
		bim:OvsCon1	rdf:type	gsw:OpeningInWand .
De overstortconstructie bestaat uit een wand. ENZOVOORT: de overstortconstructie modelleren zoals Overstortput	→	bim:OvsCon1	gsw:hasPart	bim:Drp2 .
		bim:Drp2	rdf:type	gsw:Overstortdrempel

De individuele hydraulische constructies binnen het bouwwerk worden vervolgens analoog aan overstortput en stuwput gemodelleerd, zie de betreffende hoofdstukken.

5.7.3.7 Wadi:



Onder de wadi ligt een drainage. Op deze drainage zijn twee 'slokopputten' aangebracht. De drainage voert af naar het aanliggende regenwaterriool. Bergend oppervlak: 1400 m², bergend vermogen: 1400 m³, infiltrerend vermogen: 15 mm/h.

De wadi heeft kenmerken zoals bergend vermogen.	→	bim:Wadi1	rdf:type	gsw:Wadi ;
			gsw:hasAspect	
			rdf:type	gsw:BergendVermogen ;
			gsw:hasValue	"1400"^^xsd:decimal ;
De wadi heeft bouwwerkoriëntatie met een buitengrens-geometrie (string = polygoon)	→	bim:Wadi1_ori	gsw:hasAspect	bim:Wadi1_ori .
			rdf:type	gsw:Bouwwerkoriëntatie ;
			gsw:hasAspect	
			rdf:type	gsw:Buitengrens ;
			gsw:hasValue	"string"^^geo:gmlLiteral ;
ENZOVOORT: De slokop-putten worden analoog aan andere putten verbonden aan leidingen (drains).	→	bim:Slokop1	rdf:type	gsw:Slokop .
		bim:Slokop2	rdf:type	gsw:Slokop .
		bim:Slokop1	gsw:hasAspect	
			rdf:type	gsw:Putorientatie ;
			gsw:hasConnection	bim:Drain1_ori_beginpunt ;

Een wadi kan in een GWSW dataset gedetailleerd worden beschreven (inclusief hydraulische eigenschappen en geometrie). Een wadi wordt echter niet als hydraulisch component in het GWSW-HydX formaat meegenomen. De bijbehorende drainage, met aansluitende putten, kan wel als netwerk in het HydX-formaat worden uitgedrukt.

5.7.3.8 Nog toevoegen ...

Nog toevoegen: Leiding met klep: knooppunt 1 = verbindingstuk, verbinding = doorlaat, knooppunt 2 = compartiment of put

Eindkap (afsluitstuk) bij uitlegger, verbindingstuk en zo worden hydraulisch (in HydX) vertaald naar knooppunt "Inspectieput": gekneveld en met minimale afmetingen.

6 Dataset conform GWSW-Basis: Gestructureerd inlezen

Er zijn verschillende wijzen van aanpak. Dit hoofdstuk beschijft als voorbeeld één van de mogelijkheden.

6.1 Selecteren/doorlopen triples vanuit het GWSW-OroX-bestand

- Gebruik dotNetRdf om een GWSW-OroX-bestand te parsen naar Triples (prestaties ok: 100000 feiten < 1 sec).
- Enkele methoden om de dataset (met triples) te doorzoeken:
 - Plaats de dataset in een relationele database, een tabel met de drie kolommen. Gebruik SQL om de dataset te ondervragen
 - Hou de dataset beschikbaar als triplestore (gebruik een library zoals dotNetRdf) en ondervraag de dataset met SPARQL.
 - De nodige queries beperken zich tot selecteren op subject, relatie of object. Voor die overzichtelijke functies kunnen ook interne containers (bijvoorbeeld <vector> of <map> van C++ STL) gebruikt worden. Met containerfuncties (zoals “find”) kan zo’n dataset effectief worden doorzocht.
- Selecteer op de relatie [rdf:type](#) om alle “individuals” in de dataset te vinden
- Selecteer op de relatie [rdfs:label](#) om de gangbare namen bij de individuals te achterhalen
- Selecteer op de relaties [hasPart](#) om de samenstelling van individuals te achterhalen. Bijvoorbeeld rioolstelsel “DM” heeft als deel rioolput “0980”.
- Selecteer op de relatie [hasAspect](#) om de kenmerken bij de individuals te lezen, zoek vervolgens op de relaties [hasValue](#) of [hasReference](#) om de bijbehorende waarde in te lezen
- Selecteer op de relatie [hasCollection](#) om de verbindingen tussen topologische individuals (middenpunt put, beginpunt leiding, eindpunt leiding) te achterhalen.

6.2 Putten inlezen

- Doorloop de dataset en selecteer op basis van de GWSW-Naam de aanwezige stelsels (toegestane soorten: gemengd rioolstelsel, drukriolering, ...)
- Doorloop de dataset en selecteer op basis van de relatie [hasPart](#) de onderdelen van het stelsel.
- Bepaal per onderdeel op basis van de GWSW-Naam of het een put is (toegestane soorten: inspectieput, overstortput, pompunit, ...)
- Doorloop de kenmerken per put (relatie [hasAspect](#)) en bepaal de bruikbaarheid: is het bekend of relevant binnen de doeldatabase
- Voor het inlezen van de putcoördinaten: Doorloop de kenmerken (relatie [hasAspect](#)) en zoek de bijbehorende Putoriëntatie, zoek vervolgens naar de Positie (relatie [hasAspect](#)), dan de X/Y aspecten (relatie [hasAspect](#)) en vervolgens de waarden (relatie [hasValue](#))

6.3 Leidingen inlezen

- Doorloop de dataset en selecteer op basis van de GWSW-Naam de aanwezige stelsels (toegestane soorten: gemengd rioolstelsel, drukriolering, ...)
- Doorloop de dataset en selecteer op basis van de relatie [hasPart](#) de onderdelen van het stelsel.
- Bepaal per onderdeel op basis van de GWSW-Naam of het een leiding is (toegestane soorten: persleiding, gemengd riool, ...)

- Doorloop de kenmerken per leiding (relatie [hasAspect](#)) en bepaal de bruikbaarheid: is het bekend of relevant binnen de doeldatabase
- De verbindingen tussen leidingen en putten zullen ook in de doeldatabase staan, vaak in de vorm van de beginputnaam en eindputnaam als attribuut bij de leiding. Om deze verbindingen te vinden: Doorloop de kenmerken (relatie [hasAspect](#)) en zoek de bijbehorende Leidingoriëntatie. Zoek vervolgens per Leidingoriëntatie het Beginpunt Leiding en Eindpunt Leiding (relatie: [hasPart](#)). Zoek bij het begin/eindpunt de koppeling (relatie [hasConnection](#), symetrisch) met de Putoriëntaties (of Compartimentoriëntaties). Bij de relatie [hasConnection](#) is de positie subject/object (links/rechts) willekeurig. Zoek vanuit de Putoriëntatie naar de Putnaam (relatie [hasAspect](#)).

7 Gereedschappen bij ontwikkeling GWSW-OroX export/import

Zoals genoemd komt op langere termijn met de inrichting van de Semantische Server door RIONED voor applicatiebouwers in het GWSW-OroX traject meer functionaliteit ter beschikking. Dit omvat:

- Ondervragen GWSW-ontologie (via eigen en standaard SPARQL-queries);
- Uploaden datasets als Turtle-bestand naar GraphDB endpoint;
- Onderhouden datasets via SPARQL of uploads;
- Valideren datasets (juiste syntax, restricties op basis GWSW-ontologie);
- Ondervragen dataset via SPARQL of browsing, eventueel in combinatie met GWSW-Model;
- Geautoriseerde toegang per dataset: editen en/of viewen.

Export vanuit een relationele database (RDB) of import naar een RDB kan rechtstreeks via een GWSW-OroX-bestand of via SPARQL-queries op een generieke Triplestore (in combinatie met alle soorten RDF-uitwisselformaten). Voor wat betreft de export uit een RDB is de syntax van de SPARQL-constructs complexer dan bijvoorbeeld rechtstreeks naar Turtle.

7.1 Export naar GWSW-OroX (Turtle / Triplestore)

Bij de implementatie van het GWSW-OroX in een beheersysteem zal vaak gestart worden met de export in Turtle-formaat. Bij de omzetting van de brondatabase naar een GWSW-OroX kunnen de volgende tools ondersteunen:

- Export vanuit Relationele database naar Triplestore
 - Mapping op GWSW_Lijsten: alle relevante GWSW-concepten en relaties
 - Overzichten van de GWSW-Ontologie via HTML-presentaties op <https://data.gwsw.nl>
 - Een tool als D2RQ, een server voor mapping SQL versus SPARQL, is voorsnag te generiek voor de gedetailleerde GWSW-Ontologie.
 - Zoals genoemd kan eerst een export via SPARQL-constructs naar een generieke Triplestore worden gedaan en van daaruit naar alle soorten RDF-formaten. Voor de verwerking van SPARQL-queries zijn er freeware-apps zoals dotNetRDF, voorbeelden van (freeware) SPARQL-endpoints zijn Jena-Fuseki en TopBraid.
 - Rechtstreeks vanuit de brondatabase met bijvoorbeeld dotNetRDF (zie hst 7.2.1) “serializing” naar het Turtle-formaat.
- Valideren geëxporteerde triplestore (in Turtle-formaat)
 - GWSW-Nulmeting gebruiken op <https://apps.gwsw.nl>, bijvoorbeeld in combinatie met conformiteitsklasse GWSW-MdsPlan. Nadere informatie te verkrijgen bij Stichting RIONED
 - Er zijn gecombineerde viewer/editors voor RDF-modellen beschikbaar. Voor het view- (en validatie-) aspect is relevant:
 - Protégé is een RDF-editor en –viewer. De Pellet reasoner draait als plug-in binnen Protégé.
 - Topbraid is de professionele variant van Protégé. Bevat bijvoorbeeld de HTML-export. De eerste ervaring met Protégé met heel grote datasets zijn niet goed, mogelijk is Topbraid een betere variant.
 - Voor gevorderden: de Pellet-reasoner toepassen, commandline based of als plug-in bij Protégé.

7.2 Import van GWSW-OroX-bestand (Turtle)

De importfunctie is natuurlijk ook van belang voor het testen van de export. Via de import van het eerder geëxporteerde GWSW-OroX bestand kan een cirkeltest worden uitgevoerd.

7.2.1 API voor RDF en SPARQL - dotNetRDF

Zoals besproken is het Turtle-bestand relatief eenvoudig van opbouw, maar het kan vanwege de mogelijke afkortingen vele vormen aannemen. Er zijn op het web gelukkig veel tools beschikbaar die bij de import (en export) kunnen assisteren. Voor de dotnet-omgeving is een goed voorbeeld het eerder genoemde dotNetRDF.

Toepassing dotNetRDF voor export

dotNetRDF kan ook toegepast worden voor de export naar een Turtle-formaat. Daarbij kunnen compressie-niveau's (bundeling van triples) worden ingesteld. Let er wel op dat sterke compressie mooie (turtle) bestanden levert maar veel rekenkracht vergt.

dotNetRDF is een open source pakket voor verwerken van RDF triples. Inclusief Wiki met uitgebreide handleiding: <https://bitbucket.org/dotnetrdf/dotnetrdf/wiki/Home>

Na download van de libraries (sources zijn niet nodig) kan in de dotnet-omgeving gerefereerd worden aan dotNetRDF.dll. Daarna is het een kleine stap een Turtle-bestand in te lezen en te verwerken. Als voorbeeld deze code (onder C++/CLR):

```
short cOro::ReadOrTtl(char *nam)
{
    // Lees een GWSW-Dataset Turtle-besand *nam
    // Namespace VDS::RDF: Referentie naar ...\\Utilities\\dotNetRDF\\Libraries\\Core\\net40\\dotNetRDF.dll
    // Load a Graph from a File
    VDS::RDF::Graph Ag = gCnew VDS::RDF::Graph();
    VDS::RDF::Parsing::TurtleParser Attlparser = gCnew VDS::RDF::Parsing::TurtleParser();
    Attlparser->Load(g, STRING(nam)); // Lees de dataset, zet in graph
    VDS::RDF::INamespaceMapper Answ = g->NamespaceMap; // De namespace mapper (gebruikte Turtle-prefixes)
    System::String AsPrf; // String met prefix ipv URI
    System::String ASubj, APred, AObj; // De uit te lezen triples (subject, predicate, object)
    VDS::RDF::Triple Atri;
    for each (tri in g->Triples) // Lees al de triples uit de Graph (= turtle bestand)
    {
        if (tri->Subject->NodeType == VDS::RDF::NodeType::Blank)
            Subj = tri->Subject->ToString();
        else
        {
            nsw->ReduceToQName(tri->Subject->ToString(), sPrf); // Zet URL om naar prefix (bim:, :, _:)
            if (sPrf == "") // Geen prefix = <URI>
                Subj = tri->Subject->ToString();
            else
                Subj = sPrf;
        }
        nsw->ReduceToQName(tri->Predicate->ToString(), sPrf); // Zet URL om naar prefix (bim:, :, _:)
        Pred = sPrf;
        if (tri->Object->NodeType == VDS::RDF::NodeType::Blank)
            Obj = tri->Object->ToString();
        else if (tri->Object->NodeType == VDS::RDF::NodeType::Literal) // Waarde (numeriek / string) (inclusief datatype: "1000"^^xsd:integer)
            Obj = tri->Object->ToString();
        else
        {
            nsw->ReduceToQName(tri->Object->ToString(), sPrf); // Zet URL om naar prefix (bim:, :, _:)
            if (sPrf == "") // Geen prefix = <URI>
                Obj = tri->Object->ToString();
            else
                Obj = sPrf;
        }
    }
    return 0;
}
```

Inlezen omvangrijke triplestores

De API dotNetRDF is niet afgestemd op grote triplestores. In dat geval kan het efficiënter zijn om alleen de parsing-functionaliteit van dotNetRDF te gebruiken en de triples in aparte geheugenlocaties op te slaan. Zie het volgende voorbeeld in C++/CLR, de ingelezen triples worden opgeslagen in de string-arrays vLOn (subject), vRelNm (predicate) en vROn (object).

```

public ref class cOro
{
private:
    cliext::vector<System::String^> ^vLOn, ^vROn, ^vRelNm;

public:
    short ReadOroTtl(char *nam);
    bool addTri(VDS::RDF::INamespaceMapper ^nsw, VDS::RDF::Triple ^tri);
};

public ref class MyHandler : VDS::RDF::Parsing::Handlers::GraphHandler
{
    // Subclass van DotNetRdf - GraphHandler (20160218)
    // Standaard GraphHandler doet intern veel meer met de parsing
    // daardoor bij grotere bestanden (Den Haag) OutOfMemory-exceptions [OOM] [DotNetRdf]
    // Override HandleTripleInternal, daarmee wordt de verdere verwerking van de gelezen triple afgevangen

public:
    VDS::RDF::IGraph ^g;
    cOro ^Oro;
    MyHandler(VDS::RDF::IGraph ^_g, cOro ^_Oro) : GraphHandler(_g, g(_g), Oro(_Oro) {});

    virtual bool HandleTripleInternal(VDS::RDF::Triple ^tri) override
    {
        return Oro->addTri(g->NamespaceMap, tri);
    }
};

short cOro::ReadOroTtl(char *nam)
{
    // Lees het Turtle-besand nam
    // Via maatwerk-GraphHandler triples afvangen

    VDS::RDF::Options::InternUris = false; // Spaart geheugen en tijd

    while (true)
    {
        VDS::RDF::Graph ^g = gcnew VDS::RDF::NonIndexedGraph(); // Memory-reductie: indexes zijn nodig voor SPARQL e.d.
        VDS::RDF::Parsing::TurtleParser ^ttlparser = gcnew VDS::RDF::Parsing::TurtleParser();
        MyHandler ^gh = gcnew MyHandler(g, this); // Graph via maatwerk handler vullen

        try
        {
            ttlparser->Load(gh, STRING(nam)); // Lees de dataset, zet in vectors
        }
        catch (System::Exception ^Exp)
        {
            // Vang de fout
        }
    }
    return 0;
}

bool cOro::addTri(VDS::RDF::INamespaceMapper ^nsw, VDS::RDF::Triple ^tri)
{
    // Voeg triple toe aan Oro-vectoren
    // Wordt aangeroepen vanuit maatwerk-Graph-handler

    // Zet zo nodig URL om naar prefix (bim:, :, _:) via
    // nsw->ReduceToQName(tri->Subject->ToString(), sPrf);

    vLOn->push_back(tri->Subject->ToString());
    vRelNm->push_back(tri->Predicate->ToString(sPrf));
    vROn->push_back(tri->Object->ToString());

    return true;
}

```