

# GWSW Ontologie in RDF

## *Een beschrijving van het protocol GWSW-OroX*

Van: Stichting RIONED

### Versie historie

20190518: URL binnen rdfs:comment in apart predicate rdfs:seeAlso ondergebracht  
20181003: Subtitel toegevoegd: dit is het GWSW-OroX protocol  
20180305: Onderscheidende kenmerken uitgebreid voor activiteiten  
20171028: Samenvatting completer gemaakt  
20171023: Eerste gepubliceerde versie  
20171013: Eerste opzet document

### Inhoud

GWSW Ontologie in RDF .....	1
1 Inleiding.....	3
1.1 Leeswijzer en uitgangspunten.....	3
1.2 Gebruikte begrippen.....	3
1.3 Keuzen bij modellering .....	4
2 Samenvatting opzet GWSW-Ontologie in RDF .....	5
2.1 Explicite definitie: basis voor determinatie.....	5
2.1.1 Kwalificerende aspecten / onderscheidende kenmerken .....	5
2.1.2 Kwalificerende samenstelling.....	5
2.1.3 Intrinsieke aspecten / processed aspects .....	5
2.2 Metagegevens .....	5
2.3 Collecties, domeintabellen.....	6
2.4 Properties in de GWSW-Ontologie .....	6
2.5 Properties in Datasets.....	7
2.6 Validatie/inferencing met GWSW-Ontologie.....	7
3 Details van de GWSW semantiek .....	9
3.1 Concepten: naamgeving en annotaties.....	9
3.1.1 Naamgeving van concepten.....	9
3.1.2 Annotaties bij concepten.....	9
3.2 Kenmerken/aspecten – Class- of Property-central.....	10
3.2.1 Eigenschappen en predicaat .....	10
3.2.2 Minimum/maximum waarde, datatype.....	11
3.2.3 Intrinsieke aspecten .....	11
3.3 Kwalificaties.....	12
3.4 Samenstellingen (delen van, verbindingen).....	12
3.5 Cardinaliteit .....	13

3.6	Onderscheidende kenmerken.....	14
3.7	Collecties.....	15

# 1 Inleiding

Het W3C definieert standaarden voor het Semantisch Web met als basis de triple-vorm: de Subject-Predicate-Object constructie. Het basisprotocol dat hieraan ten grondslag ligt is RDF.

De term RDF wordt in deze notitie gebruikt voor de combinatie van meerdere protocollen: RDF, RDFS en OWL 2.

Het GegevensWoordenboek Stedelijk Water (GWSW), een ontwikkeling van de stichting RIONED, is oorspronkelijk ontwikkeld in de Gellish taal. Ook Gellish is een semantische modelleringstaal in het zogenaamde ORO (Object-Relatie-Object) formaat. In het najaar van 2015 is het GWSW omgezet naar RDF.

In de "backend" situatie gebruikt het GWSW-projectteam het Gellish formaat nog steeds voor ontologie-ontwerp en -beheer. Via een geautomatiseerd proces wordt echter altijd gepubliceerd conform de RDF semantiek.

*Met dank aan:*

Daan Oostinga (Semmtech)

Mike Henrichs (Semmtech)

Michel Böhms (TNO)

Matthé van Koetsveld (CIM Architects)

Linda van den Brink (Geonovum)

## 1.1 Leeswijzer en uitgangspunten

Bij de uitwerking van deze tekst is er van uitgegaan dat de lezer bekend is met de principes en semantiek van RDF/RDFS/OWL 2 en het uitwisselformaat Turtle.

In de voorbeelden en in de praktijk (bij uitwisseling van GWSW-gegevens) gebruiken we het Turtle-formaat. Voor de concepten binnen de GWSW-Ontologie hanteren we in de voorbeelden de prefix "gsw:". Voor individuals in een dataset wordt de prefix "bim:" gebruikt.

In dit hoofdstuk vind u de begrippen en uitgangspunten bij de modellering in RDF. In het volgende hoofdstuk wordt samenvattend de opzet van het RDF model beschreven. In het laatste hoofdstuk vind u de gedetailleerde uitwerking van het RDF model.

## 1.2 Gebruikte begrippen

RDF, RDFS, OWL 2

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

Ontologie

Datastructuur die alle relevante entiteiten en hun onderlinge relaties en regels binnen een domein bevat. (bron: Wikipedia)

Individual

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

## Property

Voor de relatie (tussen subject en object) zijn meerdere namen gebruikelijk ("predicate", "property name"), we hanteren in dit document "property".

## CE

De afkorting CE wordt gebruikt voor Class Expressions (in Description Logics "complex concepts"). CE's worden ondermeer gevormd door Classes te binden aan Property Expressions. Met Class Expressions kunnen we onder andere restricties benoemen voor Properties waarmee concepten/klassen worden onderscheiden.

## Dataset

Een dataset bevat de beschrijving van een fysiek stedelijk water systeem, de "individuals" op basis van het GWSW. De term ABox wordt ook gebruikt: "assertion components" binnen een ontologie. Voor het model (de concepten) wordt dan de term TBox gebruikt: "terminological components".

## 1.3 Keuzen bij modellering

Voor de modellering is uitgegaan van het RL (Rule Language) profiel. Dit profiel gebruikt nagenoeg alle OWL 2 semantiek en is toereikend voor het modelleren van de GWSW-Ontologie.

Voor de definitie van klassen, eigenschappen, datatypen en restricties kunnen verschillende benaderingen gekozen worden. De volgende uitgangspunten zijn gehanteerd:

- Bij het ontwerp van het GWSW in RDF wordt het oorspronkelijke Gellish-modellering zonder informatieverlies getransformeerd naar het RDF-model.
- In het verlengde daarvan: Voor de indeling in soorten, de vaststelling van de taxonomie, wordt de onderscheidende definitie zo expliciet mogelijk beschreven.
- Bij de modellering is rekening gehouden met de reasoner-prestaties, de benodigde rekenkracht voor de inferencing verschilt sterk per gekozen oplossing. Voor deze afweging is de Pellet reasoner versie 2.3.1 (voor OWL 2 RL) gebruikt.

## 2 Samenvatting opzet GWSW-Ontologie in RDF

### 2.1 Explicite definitie: basis voor determinatie

Voor de indeling in soorten, de bepaling van de taxonomie, wordt de onderscheidende definitie zo expliciet mogelijk beschreven. Determinerend kan daarmee (de naam van) een soort worden bepaald. Verschillende elementen in de ontologie spelen hierbij een rol, die zijn beschreven in de volgende paragrafen.

#### 2.1.1 Kwalificerende aspecten / onderscheidende kenmerken

De onderscheidende kenmerken specificeren de soorten, de GWSW ontologie hanteert de volgende:

- Doel (waarvoor)
- Toepassing (waarin)
- Functie (wat doet het)
- Uitvoering (hoe)
- Structuur (waaruit)

Meer specifiek voor activiteiten:

- Technologie (werkwijze, eisen)
- Resultaat
- Mechanisme (waarmee)

In RDF wordt het kwalificerend aspect beschreven door een CE met een restrictie op de property [hasAspect](#) gecombineerd met een restrictie op de property [hasReference](#).

#### 2.1.2 Kwalificerende samenstelling

De structuur wordt voor wat betreft de samenstelling expliciet beschreven door een CE met een restrictie op de property [hasPart](#) gecombineerd met de benoeming van de cardinaliteit. De cardinaliteit beschrijft het aantal voorkomens van een property tussen twee soorten.

#### 2.1.3 Intrinsieke aspecten / processed aspects

Afhankelijk van de soort kunnen kenmerken worden specialiseerd. Die intrinsieke kenmerken horen dan exclusief bij een soort. De CE beschrijft een restrictie op de property [hasAspect](#) in combinatie met het gerelateerd kenmerktype.

## 2.2 Metagegevens

Een ORO-feit (één excelregel) in het backend Gellish-model bevat een serie metagegevens zoals Cardinaliteit Links/Rechts, UoM, Brondefinitie, Datum Begin/Wijziging. Die worden als volgt in het RDF-model meegenomen:

De Cardinaliteit wordt via CE's in RDF uitgedrukt. De cardinaliteit kan in twee richtingen gelden, daarvoor is voor de relevante properties een inverse geïntroduceerd. Ook voor deze omgekeerde property geldt dan via de CE een restrictie op cardinaliteit.

De UoM, Brondefinitie en Datums worden als annotaties bij de concepten opgenomen.

De Taalgemeenschap wordt als extra naam bij de concepten (met property `rdfs:label`) vermeld.

De Collection Of Facts speelt een belangrijke rol in het GWSW- model, hiermee wordt aangegeven welk ORO-feit bij welke discipline/module (GWSW-Basis, GWSW-RIB, enz.) hoort. Binnen RDF wordt

dit onderscheid gemaakt door aparte RDF-modellen per (combinatie van) van Collection Of Facts te maken. Het geldende "GWSW-filter" is altijd met de property rdfs:label in de ontologie opgenomen.

## 2.3 Collecties, domeintabellen

Alle collectie-leden zijn in de GWSW topologie opgenomen als individuen met annotaties. RDF beschrijft de enumeratie van individuals per collectie. Hoofdstuk 3.7 bevat de details.

## 2.4 Properties in de GWSW-Ontologie

De tabellen beschrijft de opgenomen properties. De toepassing van properties (per klasse) is in de GWSW-Ontologie vaak aan regels gebonden door middel van een Class Expression (CE). In de volgende tabel is dat aangegeven ("CE").

Property	Inverse property	Type	Omschrijving
owl:versionInfo			<i>Subject (ontologie) heeft versieomschrijving Literal</i>
rdf:type			<i>Subject is van het type Object (Klasse-naam)</i>
rdfs:subClassOf			<i>Subject is van het subtype Object (Klasse-naam)</i>
rdfs:label			<i>Subject heeft als naam Literal (ook vertalingen, dan meerdere rdfs:label properties) (annotatie)</i>
skos:altLabel			<i>Subject heeft als synoniem Literal (ook vertalingen, dan meerdere rdfs:altLabel properties) (annotatie)</i>
skos:hiddenLabel			<i>Subject heeft als id Literal (annotatie, alleen gebruikt voor referentieobjecten, individuals zoals collectie-elementen)</i>
skos:notation			<i>Subject heeft als code/afkorting Literal (annotatie)</i>
skos:definition			<i>Subject heeft als definitie Literal (definitie zonder bron-referentie) (annotatie)</i>
rdfs:isDefinedBy			<i>Subject is gedefinieerd door Literal (definitie met bron-referentie) (annotatie)</i>
rdfs:seeAlso			<i>Subject heeft aanvullende informatie op Literal (URL) (annotatie)</i>
rdfs:comment			<i>Subject heeft als commentaar Literal (annotatie)</i>
owl:inverseOf			<i>Subject-property is de inverse van Object-property</i>
hasUnit		owl:AnnotationProperty	<i>Subject heeft als eenheid Literal (annotatie)</i>
hasDateStart		owl:AnnotationProperty	<i>Subject heeft als begindatum Literal (annotatie)</i>
hasDateChange		owl:AnnotationProperty	<i>Subject heeft als wijzigingsdatum Literal (annotatie)</i>
hasAspect	isAspectOf	owl:ObjectProperty	<i>CE beschrijft restrictie op cardinaliteit: Bij subject mag property hasAspect 0-n maal of min 0-n en max 1-n maal voorkomen</i>
hasValue		owl:DatatypeProperty owl:FunctionalProperty	<i>CE beschrijft restrictie op object: Bij subject met property hasValue mogen alleen objecten van een bepaald datatype voorkomen. Bij het datatype kunnen vervolgens restricties op inhoud worden meegegeven.</i>
hasReference		owl:ObjectProperty owl:FunctionalProperty	<i>CE beschrijft restrictie op object: Bij subject met property hasReference mogen alleen objecten van een bepaalde klasse (collectie) voorkomen</i>
hasInput	isInputOf	owl:ObjectProperty	<i>CE beschrijft restrictie op cardinaliteit: Bij subject mag property hasInput 0-n maal of min 0-n en max 1-n maal voorkomen</i>
hasOutput	isOutputOf	owl:ObjectProperty	<i>CE beschrijft restrictie op cardinaliteit: Bij subject mag property hasOutput 0-n maal of min 0-n en max 1-n maal voorkomen</i>
hasPart	isPartOf	owl:ObjectProperty	<i>CE beschrijft restrictie op cardinaliteit: Bij subject mag property hasPart 0-n maal of min 0-n en max 1-n maal voorkomen</i>

<a href="#">hasConnection</a>		owl:ObjectProperty owl:SymmetricProperty	CE beschrijft restrictie op cardinaliteit: Bij subject mag property hasConnection 0-n maal of min 0-n en max 1-n maal voorkomen
<a href="#">hasRepresentation</a>		owl:ObjectProperty owl:FunctionalProperty	

Inverse properties zijn nodig om verschillen in cardinaliteit bij omgekeerde relaties te kunnen definiëren. Ze worden alleen gebruikt bij object-properties waarvan het type niet symmetrisch ([hasConnection](#)) of functioneel ([hasRepresentation](#)) is.

Voor het uitdrukken van CE's voorziet OWL 2 in een groot aantal (restrictie) properties. Daarmee kunnen we klassen expliciet onderscheiden, de GWSW Ontologie bevat de volgende :

Property	Toelichting
owl:onClass	Uitdrukken van cardinaliteit
owl:onProperty	Veelvuldig toegepast voor uitdrukken van klassen, vanwege "property central" principe.
owl:hasValue	Veelvuldig toegepast voor uitdrukken van klassen, vanwege "property central" principe.
owl:allValuesFrom	Uitdrukken van range bij waarden
owl:someValuesFrom	Uitdrukken van intrinsieke en onderscheidende kenmerken
owl:disjointWith	Uitdrukken van ruimtelijke <a href="#">hasPart</a> relatie
owl:unionOf	Uitdrukken van ruimtelijke <a href="#">hasPart</a> relatie
owl:qualifiedCardinality	Uitdrukken van verplichte properties
owl:maxQualifiedCardinality	Uitdrukken van maximum aantal properties
owl:minQualifiedCardinality	Uitdrukken van minimum aantal properties
owl:intersectionOf	Uitdrukken van onderscheidende kenmerken

Hoofdstuk 3 beschrijft de toepassing van deze properties.

## 2.5 Properties in Datasets

In datasets conform het GWSW worden de volgende properties gebruikt:

Property	Toelichting
rdf:type	<i>Subject is van het type Object (klasse-naam)</i>
rdfs:label	<i>Subject heeft als naam Literal</i>
<a href="#">hasAspect</a>	<i>Subject heeft als kenmerk Object</i>
<a href="#">hasValue</a>	<i>Subject heeft als waarde Literal (subject is kenmerk)</i>
<a href="#">hasReference</a>	<i>Subject heeft als referentie Object (subject is kenmerk)</i>
<a href="#">hasInput</a>	<i>Subject heeft als invoer Object</i>
<a href="#">hasOutput</a>	<i>Subject heeft als uitvoer Object</i>
<a href="#">hasPart</a>	<i>Subject heeft als deel Object</i>
<a href="#">hasConnection</a>	<i>Subject heeft verbinding met Object</i>
<a href="#">hasRepresentation</a>	<i>Subject heeft als representatie Object</i>

## 2.6 Validatie/inferencing met GWSW-Ontologie

Hier volgt een opsomming van de mogelijke inferences en validaties. In enkele gevallen is reasoning op basis van het UNA (Unique Name Assumption) principe nodig. De controle op cardinaliteit is beperkt vanwege het OWA (Open World Assumption) principe in RDF.

- Controle op hasReference-waarden binnen domein van collecties / keuzelijsten (UNA)
- Controle op correcte typering binnen samenstellingen via "[hasPart](#)".
  - Ruimte [hasPart](#) "Object". Object: alleen van de klasse Ruimte of FysiekObject

- FysiekObject **hasPart** "Object". Object: alleen van de klasse Ruimte of FysiekObject
- Inferencing: Individual-klasse wordt afgeleid uit intrinsiek aspect.
  - **hasAspect** BreedteLeiding => Individual = Leiding
- Inferencing: Individual-klasse wordt afgeleid uit onderscheidend kenmerk.
  - **hasAspect** Uitvoering + hasReference Klein => Individual = KleinObject
- Controle op correct gebruik datatype bij **hasValue**: decimal, string, integer, double, date, time, year.
- Controle op numerieke waarden binnen minimum maximum grenzen
- Cardinaliteit, aantal voorkomens per property boven het voor het type gedefinieerde maximum wordt gemeld (UNA)
  - ook "inverse"-cardinaliteit wordt in de reasoning meegenomen
  - minimum cardinaliteit en shall-relatie wel gemodelleerd, controle op strijdigheid met typering niet mogelijk (OWA)



## 3 Details van de GWSW semantiek

### 3.1 Concepten: naamgeving en annotaties

#### 3.1.1 Naamgeving van concepten

Het hanteren van begrijpbare namen voor concepten is de gangbare RDF praktijk. We gaan uit van camelCase of CamelCase notatie van de namen voor respectievelijk de properties (starten met lowercase) als de klassen (starten met uppercase).

In het GWSW komen vertalingen voor, die worden als `rdfs:label` (voorzien van de taalcode) benoemd. Ook synoniemen komen veelvuldig in het GWSW voor, die worden benoemd met de relatie `skos:altLabel`.

Voor de property-namen wordt altijd de Engelse taal gebruikt.

In een dataset wordt via `rdf:type` verwezen naar de GWSW-conceptnaam:

<code>bim:Put1</code>	<code>rdf:type</code>	<code>gwsw:Put</code>
-----------------------	-----------------------	-----------------------

Voorkeursterm GWSW-concepten

Voor de voorkeursterm van GWSW-concepten geldt het volgende uitgangspunt:

Altijd de literal bij `rdfs:label` als voorkeursterm gebruiken. Als die meertalig is (er kunnen meerdere `rdfs:label` relaties zijn met een eigen taalaanduiding) geldt altijd de voorkeur `@nl` en daarna `@en` (van een GWSW-concept is minimaal een `@nl` of een `@en` versie aanwezig).

In het Gellish-backend is een nummer-identificatie (naast het unieke label) belangrijk. Dit unieke nummer wordt met de property `skos:hiddenLabel` benoemd, maar zal op termijn zijn waarde verliezen.

#### 3.1.2 Annotaties bij concepten

Een voorbeeld van gebruikte annotaties:

<code>gwsw:Put</code>	<code>rdf:type</code>	<code>owl:Class</code> ;
	<code>rdfs:label</code>	<code>"Put"@nl</code> ;
	<code>rdfs:subClassOf</code>	<code>:FysiekObject</code> ;
	<code>skos:definition</code>	<code>"Verticale waterdichte ..."@nl</code> ;
	<code>rdfs:isDefinedBy</code>	<code>"[IMGeo:1.0/2007] Gegraven of ..."@nl</code> ;
	<code>rdfs:seeAlso</code>	<code>"http://www.w3.org"</code> ;
	<code>rdfs:comment</code>	<code>"Toelichting bij put"</code> ;
	<code>skos:notation</code>	<code>"AAA"</code> .

Coderingen komen veel voor in het GWSW, bijvoorbeeld als taalafhankelijke aanduidingen van toestandsaspecten in de EN13508-2. Codes van concepten zijn object bij de property `skos:notation`. In het voorbeeld is een fictieve code "AAA" gebruikt.

Er kunnen meerdere codes per context voorkomen. Voor het reinigen van een leiding worden bijvoorbeeld andere codes gebruikt dan voor het inspecteren van een leiding. Om dat onderscheid te kunnen maken is in de GWSW-Ontologie een datatype aan de code toegevoegd. Dat datatype representeert het geldende notatie-schema.

```

:StartNodeReference skos:notation "AAB"^^:Dt_Notation_RIB . (inspecteren leiding)
:StartNodeReference skos:notation "GAB"^^:Dt_Notation_RRB . (reinigen leiding)
:Dt_Notation_RRB rdfs:label "Codering reinigen put/leiding"@nl ;
rdf:type rdfs:Datatype .

```

## 3.2 Kenmerken/aspecten – Class- of Property-central

### 3.2.1 Eigenschappen en predicaat

In RDF is het gebruikelijk om eigenschappen van een subject te benoemen in het predicate ("property-central"). Bijvoorbeeld:

```

bim:Put1 gsw:hasAspectPutHoogte 1000^^xsd:integer .

```

Zo'n eigenschap/kenmerk kan ook als apart concept ("class-central") worden onderscheiden:

```

bim:Put1 gsw:hasAspect bim:Hgt1 .
bim:Hgt1 rdf:type gsw:PutHoogte ;
rdf:type 1000^^xsd:integer .
gsw:hasValue

```

De notatie (in turtle) blijft overzichtelijk, het object Hgt1 kan anoniem blijven (zonder URI) en wordt bijvoorbeeld gecombineerd met de specificatie van putmateriaal (zie hoofdstuk "Collecties"):

```

bim:Put1 gsw:hasAspect
[
rdf:type gsw:PutHoogte ;
gsw:hasValue 1000^^xsd:integer
],
[
rdf:type gsw:PutMateriaal ;
gsw:hasReference gsw:Beton .
]

```

In de GWSW Ontologie definieert voor veel kenmerken metagegevens zoals de "wijze van inwinning". Dat is volgens het class-central principe relatief eenvoudig te beschrijven:

```

bim:Put1 gsw:hasAspect
[
rdf:type gsw:PutHoogte ;
gsw:hasValue 1000^^xsd:integer ;
gsw:hasAspect
[
rdf:type gsw:Inwinning ;
gsw:hasAspect
[
rdf:type gsw:WijzeVanInwinning ;
gsw:hasReference gsw:Schatting ;
]] .
]
]

```

Property [hasAspect](#) is van het type owl:ObjectProperty.

Property [hasValue](#) is van het type owl:DatatypeProperty en owl:FunctionalProperty. De property rdf:value wordt niet toegepast omdat het geen owl-type is (en daardoor bijvoorbeeld OWL RL-reasoning beperkt wordt).

Property [hasReference](#) is van het type owl:ObjectProperty en owl:FunctionalProperty.

In de GWSW ontologie gaan we uit van het "class-central" principe. Deze oplossing biedt een uitgebreidere semantiek en heeft (daarom) de voorkeur:

- Door objectificering van het kenmerk kan het in meerdere relaties gebruikt:

- Het kenmerk kan met de relatie "hasInput" als object voor berekeningen staan
- Het kenmerk kan zelf kenmerken bevatten (met de relatie "hasAspect"), bijvoorbeeld metagegevens over de inwinning. In het GWSW komt dit veel voor (zie het eerdere voorbeeld).
- Als alternatief voor de property-central oplossing met rdfs:range kunnen met een CE als subklassering op subject + property "hasValue" restricties aan het datatype worden toegevoegd. (zie hst. 3.2.2)
- Als alternatief voor de property-central oplossing met rdfs:domain kunnen met een CE als subklassering op het subject restricties aan de combinatie predicate en object worden toegevoegd. (zie hst. 3.2.3).
- In een dataset kan naar believen het aspect als URI of anoniem (via een blank node) worden uitgeschreven.
- De uitgebreidere semantiek kan in een dataset met beperkte syntax worden beschreven (zie het eerdere voorbeeld)

### 3.2.2 Minimum/maximum waarde, datatype

Voor restricties op de kenmerk-waarde hanteren we:

```

gsw:hasAspect      rdf:type      owl:ObjectProperty .
gsw:hasValue       rdf:type      owl:DatatypeProperty ;
                   rdf:type      owl:FunctionalProperty . (waarde-relatie altijd max 1)
gsw:PutHoogte     rdfs:subClassOf gsw:Kenmerk ;
                   rdfs:label    "Put hoogte" ;
                   rdfs:subClassOf
                   [
                     rdf:type      owl:Restriction ;
                     owl:onProperty gsw:hasValue ;

```

Alleen een restrictie op het standaard datatype:

```

                   owl:allValuesFrom xsd:integer
                   ] .

```

Of restricties op min/max waarde met een GWSW-datatype:

```

gsw:dt_PutHoogte  owl:allValuesFrom gsw:dt_PutHoogte
                   ] .
                   rdf:type      rdfs:Datatype ; (typering verplicht in OWL RL)
                   rdfs:label    "Put hoogte - datatype" ;
                   owl:equivalentClass
                   [
                     rdf:type      rdfs:Datatype ;
                     owl:onDatatype xsd:integer
                     owl:withRestrictions
                     ( [xsd:minInclusive "0"^^xsd:integer] [xsd:maxExclusive "10000"^^xsd:integer] )
                   ] .

```

### 3.2.3 Intrinsieke aspecten

Een intrinsiek aspect behoort specifiek (per definitie) bij een klasse.

gsw:PutHoogte	rdfs:comment rdfs:subClassOf	"Intrinsiek kenmerk" ; gsw:Hoogte .
---------------	---------------------------------	--

Vanwege inference-snelheid hier eenzijdige subklassering aangehouden. Alleen de restrictie als subklasse is voldoende ("sufficient versus necessary"), type-prolongatie van CE naar Put. Wordt alleen (op deze manier) gebruikt in de dataset

Via blank-node subklasse bij Put met restrictie op property:

[	rdf:type owl:onProperty owl:someValuesFrom rdfs:subClassOf	owl:Restriction ; gsw:hasAspect ; gsw:PutHoogte ; gsw:Put .
]		

Met deze definitie worden Putten onderscheiden op basis van het intrinsieke aspect Puthoogte, de individual hoeft in de dataset niet getypeerd te worden:

bim:Put1	gsw:hasAspect	
[	rdf:type gsw:hasValue	gsw:PutHoogte ; "1100"
].		

### 3.3 Kwalificaties

De restrictie-property owl:hasValue gebruiken we voor het kwalificeren van standaardwaardes van kenmerken (bijvoorbeeld voor versie-aanduidingen) .

gsw:GswVersie	rdf:type rdfs:subClassOf	owl:Class ;
	[ rdf:type owl:onProperty owl:hasValue ].	owl:Restriction ; gsw:hasReference ; gsw:Abc ;
gsw:Abc	rdfs:label rdf:type	"abc"^^xsd:string gsw:GswVersie . (wordt hiermee individual)

Met de restrictie-property owl:allValuesFrom worden concepten als kwalificatie benoemd.

gsw:Nodeld	rdf:type rdfs:subClassOf	owl:Class ;
	[ rdf:type owl:onProperty owl:allValuesFrom ].	owl:Restriction ; gsw:hasReference ; (Objectproperty) gsw:Rioolput

### 3.4 Samenstellingen (delen van, verbindingen)

De samenstelling-relaties komen voor in datasets en in de ontologie, in de ontologie worden alleen de restricties op deze relaties beschreven. De volgende relaties worden gebruikt:

hasPart  
hasInput

## hasOutput hasConnection

### Activiteiten

De relaties [hasInput](#) en [hasOutput](#) worden gebruikt voor de beschrijving van activiteiten en processen, een voorbeeld:

gsw:hasInput	rdfs:label rdf:type	"has as input" ; owl:ObjectProperty .
gsw:InspecterenPut	rdfs:label rdfs:subClassOf	"Inspecteren van een put"@nl, "Inspection manhole"@en ; gsw:Activiteit .

In een dataset:

bim:Insp1	rdf:type gsw:hasInput	gsw:InspecterenPut ; bim:Put1 .
-----------	--------------------------	------------------------------------

### Verbindingen

[hasConnection](#) wordt van het type owl:SymmetricProperty.

### Decompositie

[hasPart](#) wordt type owl:ObjectProperty en geldt voor zowel fysieke als ruimtelijke composities. Ruimtelijke composities ("bevatten" iets) via de property [hasPart](#) / [isPartOf](#) met restrictie op object-class:

gsw:Ruimte	rdf:type rdfs:label rdfs:subClassOf	owl:Class ; "Ruimte"@nl ;
	[ rdf:type owl:onProperty owl:allValuesFrom	owl:Restriction ; gsw:hasPart ;
	[ rdf:type owl:unionOf	owl:Class; (gsw:Ruimte gsw:FysiekObject) ;
	] ]	

In combinatie daarmee is in de ontologie expliciet gemaakt dat bijvoorbeeld individuals van het type FysiekObject altijd iets anders zijn dan die van het type Ruimte:

gsw:Ruimte	owl:disjointWith owl:disjointWith	gsw:Kenmerk ; gsw:FysiekObject . Enzovoort
------------	--------------------------------------	---

## 3.5 Cardinaliteit

Cardinaliteiten worden in CE's geborgd. Fictief voorbeeld van cardinaliteit voor objectproperty + objecttype:

```

gsw:Rioolstelsel    rdfs:subClassOf
[
  rdf:type          owl:Restriction ;
  owl:minQualifiedCardinality "1"^^xsd: nonNegativeInteger ;
  owl:onProperty  gsw:hasPart ;
  owl:onClass     gsw:Lozingspunt .
];
rdfs:subClassOf
[
  rdf:type          owl:Restriction ;
  owl:maxQualifiedCardinality "99"^^xsd: nonNegativeInteger ; (onbeperkt = geen maximum)
  owl:onProperty  gsw:hasPart ;
  owl:onClass     gsw:Lozingspunt .
].

```

Als de cardinaliteit niet beperkt is:

```

owl:minQualifiedCardinality    "0"^^xsd: nonNegativeInteger ;

```

Hiermee is wel gemarkeerd dat het aspect *relevant is voor* het subject.

Als de cardinaliteit verplicht voor een klasse:

```

owl:qualifiedCardinality      "1"^^xsd: nonNegativeInteger ;

```

Definiërende relaties

In het GWSW is beschreven welke relaties onderscheidend zijn voor de typering. Een rioolput moet bijvoorbeeld een deksel hebben om een echte rioolput te zijn. Daarvoor geldt ook een type-prolongatie van *Rioolput* naar *CE*, als iets een *Rioolput* is dan is het ook iets met een *Deksel*.

```

gsw:Rioolput    rdfs:subClassOf
[
  rdf:type          owl:Restriction ;
  owl:minQualifiedCardinality "1"^^xsd: nonNegativeInteger ;
  owl:onProperty  gsw:hasPart ;
  owl:onClass     gsw:Deksel .
].

```

Inversed property

Cardinaliteit kan tweezijdig worden beschreven, daarvoor zijn er omgekeerde relaties nodig.

```

gsw:isPartOf    rdfs:label          "has as part (inverse)" ;
                rdf:type          owl:ObjectProperty ;
                owl:inverseOf  gsw:hasPart .

```

Ook voor deze inversed property + object (was subject) wordt dan de cardinaliteit gedefinieerd.

### 3.6 Onderscheidende kenmerken

Een onderscheidend kenmerk wordt gemodelleerd met restricties binnen een CE. Bij benoeming van de CE als *equivalentClass* vragen deze restricties veel rekenkracht, daarom is hier ook voor een éénzijdige subtypering (*rdfs:subClassOf* ipv *owl:equivalentClass*) gekozen. Daarmee leveren we wel semantiek in ("sufficient", niet "necessary").

Definieer Onderscheidend Kenmerk

```

gsw:Uitvoering    rdfs:comment      "Onderscheidend kenmerk"@nl ;
                rdfs:subClassOf  gsw:Kenmerk .

```

Combineer het kenmerk en de waarde ervan in een CE

gsw:KleinObject [	rdfs:subClassOf rdf:type owl:onProperty owl:someValuesFrom [ owl:intersectionOf (gsw:Uitvoering [ rdf:type owl:onProperty owl:hasValue ]) ]	gsw:FysiekObject . owl:Restriction ; Via blank node: eenzijdige subklasse gsw:hasAspect ;  zowel klasse Uitvoering als CE-restrictie op hasReference
] gsw:Klein	rdfs:subClassOf rdfs:label rdf:type	gsw:KleinObject . "klein" ; gsw:Uitvoering . (wordt hiermee individual)

Als in de dataset een individual als volgt beschreven is kan een reasoner op basis van de ontologie afleiden dat KleinObject1 van het type KleinObject is:

bim:KleinObject1	gsw:hasAspect [ rdf:type gsw:hasReference ]	gsw:Uitvoering ; gsw:Klein .
------------------	---	---------------------------------

### 3.7 Collecties

Voor de modellering van collecties gebruiken we in RDF een enumeratie van individuals. Alle collectiemembers (elementen) zijn dus in de GWSW-topologie opgenomen als individuen met annotatieproperties.

gsw:PutMateriaal	rdfs:subClassOf rdfs:label rdfs:subClassOf [ rdf:type owl:onProperty owl:allValuesFrom ]	gsw:Kenmerk ; "Put materiaal" ;  owl:Restriction ; gsw:hasReference ; (FunctionalProperty + ObjectProperty) gsw:PutMatColl ;
gsw:PutMatColl	rdf:type rdfs:subClassOf [ rdf:type owl:oneOf ]	owl:Class ;  owl:Class ; (gsw:Betongsw:Pvc) (individuals)
gsw:Betong	rdfs:label rdf:type skos:hiddenLabel skos:notation	"betong" ; (annotatie: naam) gsw:PutMatColl ; (algemene typering) 111004801 ; (nummer-id van concept) "A" . (annotatie: code)

In de dataset verwijzen naar het individu:

```
bim:Put1      rdf:type      gsw:Put ;
              gsw:hasAspect
              [
                rdf:type      gsw:Put_mat ;
                gsw:hasReference
                gsw:Betón ;      (Objectproperty)
              ] .
```