# Issuelijst

Deze lijst bevat verschillende problemen die opgepakt moeten worden. Er is een onderverdeling gemaakt naar probleemeigenaar.

## OP

### Verwijzingen naar GIO’s

Voor de modellering van de GIO’s is het noodzakelijk dat we kunnen verwijzen naar de FeatureCollection en naar onderliggende FeatureMembers. Ik verwacht daarom, dat ik een JOIN-id kan aangeven om te koppelen naar het informatieobject en een geo-id om te koppelen naar de FeatureCollection of FeatureMember. Op dit moment geef je met <InformatieobjectRef> in het besluit aan naar welke join-id je verwijst, maar niet naar welke geo-id. De vraag is dus: Hoe verwijs ik vanuit mijn besluit naar een afzonderlijk FeatureMember in een GIO?

<informatieobjectRefs>

<informatieobjectRef>/join/id/regdata/gm0037/2019/loc\_ABC@2019-01-16;3520</informatieobjectRef>

<informatieobjectRef>/join/id/regdata/gm0037/2019/loc\_DEF@2019-01-16;3520</informatieobjectRef>

<informatieobjectRef>/join/id/regdata/gm0037/2019/loc\_ZA@2019-01-16;3520</informatieobjectRef>

<informatieobjectRef>/join/id/regdata/gm0037/2019/loc\_ZB@2019-01-16;3520</informatieobjectRef>

<informatieobjectRef>/join/id/regdata/gm0037/2019/loc\_speelhal@2019-04-10;3520</informatieobjectRef>

</informatieobjectRefs>

### Opbouw van GIO’s

De opbouw van InformatieObjectVersie is als volgt:

* ExpressionIdentificatie
* InformatieObjectMetadata
* GeoInformatie
* Achtergrond

#### ExpressionIdentificatie

Meer een vraag dan een issue is of het hier gaat over de identificatie van het informatieobject of van het besluit waarbij het hoort. In de huidige voorbeeldbestanden is uitgegaan van het laatste.

#### InformatieObjectMetadata

Deze bevat de benodigde informatie en de koppeling naar het gml-bestand. Niet echt een issue maar een schoonheidsdingetje: element heeftBestanden heeft als onderliggend element heeftBestand (één of meer) en daaronder element Bestand (exact één). Element heeftBestand is eigenlijk overbodig.

<InformatieObjectMetadata xmlns="https://standaarden.overheid.nl/stop/imop/data/">

<eindverantwoordelijke>/join/id/stop/gemeente/gm0037</eindverantwoordelijke>

<maker>/join/id/stop/gemeente/gm0037</maker>

<noemer>Bedrijf categorie 2</noemer>

<officieleTitel>/join/id/regdata/gm0037/2019/loc\_ABC@2019-01-16;3520</officieleTitel>

<publicatieinstructie>TeConsolideren</publicatieinstructie>

<soortInformatieobject>/join/id/stop/informatieobject/gio\_001</soortInformatieobject>

<!-- Koppeling naar het gml-bestand -->

<heeftBestanden>

<heeftBestand>

<Bestand>

<bestandsnaam>Bedrijf\_categorie\_2.gml</bestandsnaam>

<hash>7d0cc69907509a96e85ecfb85c7036d92facd71d71e6e57cabab3d17ad584d352cddaf0a23efd6e2fcb98196f4d20d9e9c9dc17a21e880309ae0b2a145b20dee</hash>

</Bestand>

</heeftBestand>

</heeftBestanden>

</InformatieObjectMetadata>

<GeoInformatie xmlns="https://standaarden.overheid.nl/stop/imop/geo/">

<!-- GeoInformatie is leeggelaten, omdat dit in het gml-bestand staat -->

</GeoInformatie>

#### GeoInformatie

Element GeoInformatie is een verplicht element dat de gml zelf bevat. Dit is in 0.98-kern verhuisd naar het gml-bestand. In een later voorstel voor de modellering van Wilko is de koppeling naar het gml-bestand aangepast.

...

</InformatieObjectMetadata>

<Bestand xmlns="https://standaarden.overheid.nl/stop/imop/data/">

<bestandsnaam>Bedrijf\_categorie\_2.gml</bestandsnaam>

<hash>7d0cc69907509a96e85ecfb85c7036d92facd71d71e6e57cabab3d17ad584d352cddaf0a23efd6e2fcb98196f4d20d9e9c9dc17a21e880309ae0b2a145b20dee</hash>

</Bestand>

<Achtergrond xmlns="https://standaarden.overheid.nl/stop/imop/data/">

...

#### Achtergrond

Element Achtergrond wordt gebruikt om informatie over de achtergrond mee te geven. Deze bevat element bronverwijzing die een geo-id van een FeatureMember kan bevatten. Klopt dit? Zo ja, dan zou ik die verwachten in de verwijzing in het besluit om te kunnen verwijzen naar onderliggende geometrieën (zie ‘Verwijzingen naar GIO’s’).

<Achtergrond xmlns="https://standaarden.overheid.nl/stop/imop/data/">

<bronactualiteit>2019-01-01</bronactualiteit>

<bronbeschrijving>top10nl</bronbeschrijving>

<brontype>achtergrond</brontype>

<bronverwijzing>#50EA019E-3C96-4631-A76C-35BCF4D7AB6D</bronverwijzing>

<nauwkeurigheid>10</nauwkeurigheid>

</Achtergrond>

### Toevoegen van waarden aan een open waardelijst

Op dit moement zijn verschillende waardelijsten gedefinieerd voor OP-metadata. Hoe verloopt de procedure om waarden toe te voegen? Ik zie in de waardelijsten geen haakje om de definitie van de waarde op te slaan. Verder heb ik geen mogelijkheid om een nieuwe waarde en de bijbehorende definitie op te geven.

### FormeleInhoud versus FormeleDivisie

Volgens de documentatie bij OP is FormeleInhoud bedoeld als het structuurelement dat de daadwerkelijk juridische tekst bevat in de vrijetekststructuur, analoog aan Artikel in de artikelstructuur. Het kan echter ook gebruikt worden als containerelement analoog aan element Inhoud in de artikelstructuur. Dan is de inhoud van FormeleInhoud volledig betekenisloos. In het onderstaande voorbeeld is een hoofdstuk aangegeven waarbij FormeleInhoud alleen een inleidende zin bevat en de onderliggende paragrafen de daadwerkelijke (juridische) informatie. Hier is FormeleDivisie het structuurelement dat de informatie bevat en FormeleInhoud het containerelement voor de tekst.

<FormeleDivisie eId="fdvs\_3" wId="gm0599\_1\_\_fdvs\_3">

<Kop>

<Opschrift>Knooppunten</Opschrift>

</Kop>

<FormeleInhoud eId="fdvs\_3\_\_fcontent\_1" wId="gm0599\_1\_\_fdvs\_3\_\_fcontent\_1">

<Al>De volgende paragrafen geven de knooppunten in de regio Rotterdam aan.</Al>

</FormeleInhoud>

<FormeleDivisie eId="fdvs\_3\_\_fdvs\_1" wId="gm0599\_1\_\_fdvs\_3\_\_fdvs\_1">

<Kop>

<Al>Knooppunt Alexander</Al>

</Kop>

<FormeleInhoud eId="fdvs\_3\_\_fdvs\_1\_\_fcontent\_1" wId="gm0599\_1\_\_fdvs\_3\_\_fdvs\_1\_\_fcontent\_1">

<Al>Tekst</Al>

</FormeleInhoud>

</FormeleDivisie>

<FormeleDivisie eId="fdvs\_3\_\_fdvs\_2" wId="gm0599\_1\_\_fdvs\_3\_\_fdvs\_2">

<Kop>

<Al>Knooppunt Hart van Zuid</Al>

</Kop>

<FormeleInhoud eId="fdvs\_3\_\_fdvs\_2\_\_fcontent\_1" wId="gm0599\_1\_\_fdvs\_3\_\_fdvs\_2\_\_fcontent\_1">

<Al>Tekst</Al>

</FormeleInhoud>

</FormeleDivisie>

...

</FormeleDivisie>

In de volgende uitwerking van dezelfde tekst is op hoofdstukniveau FormeleDivisie het structuurelement dat de informatie bevat en FormeleInhoud het containerelement, maar op paragraafniveau is FormeleInhoud het structuurelement.

<FormeleDivisie eId="fdvs\_3" wId="gm0599\_1\_\_fdvs\_3">

<Kop>

<Opschrift>Knooppunten</Opschrift>

</Kop>

<FormeleInhoud eId="fdvs\_3\_\_fcontent\_1" wId="gm0599\_1\_\_fdvs\_3\_\_fcontent\_1">

<Al>De volgende paragrafen geven de knooppunten in de regio Rotterdam aan.</Al>

</FormeleInhoud>

<FormeleInhoud eId="fdvs\_3\_\_fcontent\_2" wId="gm0599\_1\_\_fdvs\_3\_\_fcontent\_2">

<Kop>

<Al>Knooppunt Alexander</Al>

</Kop>

<Al>Tekst</Al>

</FormeleInhoud>

<FormeleInhoud eId="fdvs\_3\_\_fcontent\_3" wId="gm0599\_1\_\_fdvs\_3\_\_fcontent\_3">

<Kop>

<Al>Knooppunt Hart van Zuid</Al>

</Kop>

<Al>Tekst</Al>

</FormeleInhoud>

...

</FormeleDivisie>

Verzoek is om een eenduidige structurering af te dwingen. Het is eenvoudig op te lossen door te kiezen voor FormeleDivisie als structuurelement en FormeleInhoud als containerelement, analoog aan de artikelstructuur. Dan kan element Kop binnen FormeleInhoud vervallen, sterker nog, je zou FormeleInhoud kunnen vervangen door het generieke element Inhoud.

Bij OW zijn de consequenties aangegeven van de huidige interpretatievrijheid.

### Keuze tussen artikelstructuur en vrijetekststructuur

Op het moment moet je binnen een regeling kiezen tussen de artikelstructuur en de vrijetekststructuur. Er zijn ook instrumenten die beide combineren zoals het projectbesluit en het programma. Op het moment is een combinatie van beide structuren niet mogelijk.

## OW

### Optimalisatie van schema’s

De OW-schema’s kunnen op verschillende manieren geoptimaliseerd worden. Het gebruik van Imvertor heeft grote invloed op de efficiëntie van de opgeleverde schema’s. Afgezien van de gml-schema’s zijn er 21 schema’s en net zoveel namespaces. Dit aantal kan drastische teruggebracht worden.

#### Elementen generiek maken

Veel elementen zijn nu specifiek gemaakt, terwijl ze eigenlijk generiek zijn. Bijvoorbeeld element Activiteit heeft een ActiviteitRef, Locatie heeft LocatieRef enzovoorts. Door alleen al deze ref-elementen generiek te maken kunnen we acht van de 21 schema’s en namespaces saneren. Naar verwachting zou je met drie á vier schema’s en namespaces kunnen volstaan.

#### Versie-aanduiding uit namespace verwijderen

In namespaces zit nu een versie-aanduiding. Schemawijzigingen zijn daardoor per definitie niet backward-compatible. Het is elke keer noodzakelijk om bestaande content bij te werken naar de laatste versie-aanduiding.

#### xs:choice versus xs:sequence

Veel onderliggende elementen in een object zijn gedefinieerd met xs:sequence in plaats van xs:choice. De gebruikelijke werkwijze is, dat xs:sequence wordt gebruikt als de volgorde relevant is. In ons geval is de volgorde niet relevant.

### Attribuut rkow:RegeltekstId op objecten

Op dit moment zijn er twee concurrerende mechanismen om vanuit een object te achterhalen in welke regeltekst ‘die is ontstaan’. Dit kan door terug te herleiden welke regelteksten verwijzen naar het object, maar ook met attribuut rkow:RegeltekstId. Dit heeft nadelen en risico’s:

* Het is nooit handig om twee concurrerende mechanismen te gebruiken, omdat er daardoor onderlinge inconsistenties kunnen ontstaan.
* Er is een lastige transformatie nodig om te achterhalen in welke regelteksten een object ‘ontstaat’. Dit kunnen er ook meerdere zijn, terwijl er maar één aangegeven kan worden.

### Waardelijsten

Binnen OW worden waarden in een waardelijst nog steeds aangeduid als xs:string. Het mechanisme om waarden toe te voegen aan een waardelijst bestaat nog niet. Ik heb begrepen dat de stelselcatalogus waardelijsten gaat beheren. In ieder geval moeten we in OW een haakje hebben om een definitie van een nieuwe waarde in een waardelijst mee te geven. De verwijzing naar een waarde in OW zou dan een uri kunnen zijn.

### FormeleInhoud versus FormeleDivisie

Zie ook ‘FormeleInhoud versus FormeleDivisie’ bij OP voor de voorbeelden.

Nu wordt in OW FormeleInhoud gebruikt om annotaties aan te koppelen. Het is afhankelijk van de manier waarop het bevoegde gezag zijn tekst structureert of daadwerkelijk de goede tekst weergegeven gaat worden in de DSO-viewer. In het volgende voorbeeld willen we het hoofdstuk ‘Knooppunten’ annoteren met thema ‘Verkeer en vervoer’.

<!-- FormeleDivisie als structuurelement en FormeleInhoud als containerelement -->

<FormeleDivisie eId="fdvs\_3" wId="gm0599\_1\_\_fdvs\_3">

<Kop>

<Opschrift>Knooppunten</Opschrift>

</Kop>

<FormeleInhoud eId="fdvs\_3\_\_fcontent\_1" wId="gm0599\_1\_\_fdvs\_3\_\_fcontent\_1">

<Al>De volgende paragrafen geven de knooppunten in de regio Rotterdam aan.</Al>

</FormeleInhoud>

<!-- FormeleDivisie als structuurelement en FormeleInhoud als containerelement -->

<FormeleDivisie eId="fdvs\_3\_\_fdvs\_1" wId="gm0599\_1\_\_fdvs\_3\_\_fdvs\_1">

<Kop>

<Al>Knooppunt Alexander</Al>

</Kop>

<FormeleInhoud eId="fdvs\_3\_\_fdvs\_1\_\_fcontent\_1" wId="gm0599\_1\_\_fdvs\_3\_\_fdvs\_1\_\_fcontent\_1">

<Al>Tekst</Al>

</FormeleInhoud>

</FormeleDivisie>

<!-- FormeleInhoud als structuurelement -->

<FormeleInhoud eId="fdvs\_3\_\_fcontent\_2" wId="gm0599\_1\_\_fdvs\_3\_\_fcontent\_2">

<Kop>

<Al>Knooppunt Hart van Zuid</Al>

</Kop>

<Al>Tekst</Al>

</FormeleInhoud>

...

</FormeleDivisie>

In owVrijetekst.xml nemen we de volgende xml op:

<sl:stand>

<ow-dc:owObject>

<!-- FormeleInhoud of FormeleDivisie? -->

<vt:FormeleInhoud wIdRegeling="/akn/nl/bill/gm0599/2019/4515/nld@2019-10-01" wId="gm0599\_1\_\_fdvs\_3\_\_fcontent\_1">

<vt:identificatie>nl.imow-gm0599.formeleinhoud.2019000001</vt:identificatie>

</vt:FormeleInhoud>

</ow-dc:owObject>

</sl:stand>

<sl:stand>

<ow-dc:owObject>

<vt:Tekstdeel ow:regeltekstId="nl.imow-gm0599.formeleinhoud.2019000001">

<vt:identificatie>nl.imow-gm0599.tekstdeel.2019000001</vt:identificatie>

<vt:thema>verkeer en vervoer</vt:thema>

<vt:formeleInhoud>

<vt-ref:FormeleInhoudRef xlink:href="nl.imow-gm0599.formeleinhoud.2019000001"/>

</vt:formeleInhoud>

</vt:Tekstdeel>

</ow-dc:owObject>

</sl:stand>

Dit houdt in dat de DSO-viewer niet de verwijzing naar hoofdstuk ‘Knooppunten’ krijgt, maar de verwijzing naar de betekenisloze zin: ‘De volgende paragrafen geven de knooppunten in de regio Rotterdam aan.’ Hier moet OZON of de DSO-viewer op een of andere manier zelf concluderen dat thema ‘Verkeer en vervoer’ hoort bij de bovenliggende FormeleDivisie. Gaan we echter de onderliggende paragraaf ‘Knooppunt Hart van Zuid’ annoteren met thema ‘Verkeer en vervoer’, dan geeft FormeleInhoud wel de goede verwijzing. OZON of de DSO-viewer moeten dan zelf interpreteren dat ze NIET de bovenliggende FormeleDivisie moeten weergeven.

<sl:stand>

<ow-dc:owObject>

<!-- FormeleInhoud of FormeleDivisie? -->

<vt:FormeleInhoud wIdRegeling="/akn/nl/bill/gm0599/2019/4515/nld@2019-10-01" wId="gm0599\_1\_\_fdvs\_3\_\_fcontent\_2">

<vt:identificatie>nl.imow-gm0599.formeleinhoud.2019000001</vt:identificatie>

</vt:FormeleInhoud>

</ow-dc:owObject>

</sl:stand>

<sl:stand>

<ow-dc:owObject>

<vt:Tekstdeel ow:regeltekstId="nl.imow-gm0599.formeleinhoud.2019000001">

<vt:identificatie>nl.imow-gm0599.tekstdeel.2019000001</vt:identificatie>

<vt:thema>verkeer en vervoer</vt:thema>

<vt:formeleInhoud>

<vt-ref:FormeleInhoudRef xlink:href="nl.imow-gm0599.formeleinhoud.2019000001"/>

</vt:formeleInhoud>

</vt:Tekstdeel>

</ow-dc:owObject>

</sl:stand>

Doordat de tekst in de vrijertekststructuur op heel veel manieren gestructureerd kan worden, moeten OZON of de DSO-viewer gaan interpreteren wat bedoeld wordt, de bovenliggende FormeleDivisie of de FormeleInhoud. De noodzaak voor eigen interpretatie is er niet als alleen FormeleDivisie structuurelement is en FormeleInhoud wordt vervangen door element Inhoud.

## Synchronisatie OP en OW

### Gemeenschappelijke waardelijsten

OP en OW hebben op dit moment de waardelijsten Thema en Onderwerp gemeenschappelijk. Beide hebben een andere invulling van de waardelijsten. Hoe kunnen we zorgen dat deze lijsten synchroon blijven lopen?

### Aanlevering van gml-bestanden

In het SPITS-overleg is afgesproken dat er geen redundantie is in aanlevering van gml-bestanden. Er zijn nu nog veel onduidelijkheden over de vraag wat er precies aangeleverd moet worden. Omdat we nu eigenlijk alleen nog maar kijken naar initiële besluiten zijn we nog niet aangelopen tegen de volgende problemen:

* Bevoegde gezagen gaan wijzigingsbesluiten aanleveren aan de LVBB. De gml-bestanden zullen de geo-wijzigingen bevatten terwijl OZON de geconsolideerde gml-bestanden nodig heeft.
* De LVBB gaat de aangeleverde geo-wijzigingen conslideren en doorleveren aan OZON. Hoe kunnen de bevoegde gezagen verwijzen naar geo-id’s die nog niet bestaan?

## “Losse” schema’s

De schema’s laten ruimte voor allerlei soorten extra elementen die niet in het document thuis horen. Zo is het nog steeds mogelijk om activiteiten te annoteren in het OP gedeelte. Dit laat ruimte om allerlei rare dingen een document in te smokkelen.

Heel veel dingen zijn geregeld via waardes van attributen. Daarop wordt niet of nauwelijks gevalideerd. Ook daar is dus veel gekkigheid mogelijk.

Al met al is de standaard zo niet restrictief genoeg.

## Juridische werking

Normen zijn verhuisd naar OW. In OP zijn ze niet meer te annoteren. Daarmee hebben de normen geen juridische werking en zijn daarmee feitelijk uit het model gevallen.