

Technisch Framework OWMS 3.5

Identificatie	http://standaarden.overheid.nl/owms/3.5/doc/technischframework.pdf
Titel	Technisch Framework OWMS 3.5
Informatietype	Richtlijn
Taal	nl-NL
Maker	Overheid heeft <i>Antwoord</i> [®]
Datum laatste wijziging	31-07-2008
Geldigheid	vanaf 01-08-2008
Locatie	Niet van toepassing

Inhoudsopgave

1	INLEIDING	3
1.1	LEESWIJZER.....	3
1.2	HET TECHNISCH FRAMEWORK	3
2	UITGANGSPUNTEN	4
2.1	DUBLIN CORE.....	5
3	XML-ARCHITECTUUR	5
4	AANSLUITEN BIJ DE XML-ARCHITECTUUR	6
4.1	COLLECTIE-EIGEN XSD AANMAKEN	6
4.2	TOEVOEGEN ANDERE METADATA	7
4.3	WIJZIGEN WAARDENBEREIK	8
4.4	VERIFICATIE GEBRUIKTE WAARDEN	11
4.5	XML-PARSERS	12
4.6	OFFLINE AANSLUITEN BIJ OWMS-XML	12
4.7	GEEN AUTOMATISCHE VALIDATIE IN 3.5	12
5	HTML-SYNTAX	13

1 Inleiding

De Overheid.nl Web Metadata Standaard (OWMS) beschrijft de metadata-eigenschappen waarmee ongestructureerde informatie van de Nederlandse overheid op een gestructureerde manier beschreven kan worden. Dit maakt het mogelijk om deze informatie in samenhang te ontsluiten, te presenteren en te vinden.

1.1 Leeswijzer

Dit document is onderdeel van de documentatie van OWMS versie 3.5. De documentatie van OWMS is geschreven voor metadata-experts die betrokken zijn bij de ontwikkeling van toepassingen van de standaard. Deze documentatie is niet specifiek geschreven voor mensen die metadata toekennen. Zij zullen meer hebben aan documentatie die bij een OWMS toepassing is geschreven: een Informatie Publicatie Model (IPM).

De documentatie bij OWMS 3.5 bestaat uit drie documenten:

A. [OWMS-3.5] Introductie OWMS 3.5 – Geeft overzicht over de componenten en organisatie van de standaard. Gebruik dit document als inleiding tot OWMS en om overzicht te krijgen over de componenten van OWMS.

B. [OWMS-MODEL] Conceptueel en semantisch model OWMS 3.5 – Geeft definities van de concepten die ten grondslag liggen aan OWMS. Gebruik dit document indien diepgaande kennis over de achtergronden van OWMS nodig is.

C [OWMS-TFW] Technisch Framework OWMS 3.5 – Geeft een beschrijving van de XML-schemadefinitie en andere technische componenten waarmee OWMS metadata in XML kan worden gevalideerd. . Gebruik dit document indien u een technische toepassing van OWMS ontwikkelt. Bijvoorbeeld een XML-schema-definitie bij een IPM.

Daarnaast is er on-line gestructureerde documentatie op detailniveau van de eigenschappen en elementen van OWMS op <http://standaarden.overheid.nl/owms/3.5/doc/>.

1.2 Het technisch framework

Om het mogelijk te maken om informatie, afkomstig van verschillende partijen en uit verschillende (software-)systemen, in samenhang te gebruiken is het van belang dat de schrijfwijze (syntax) en betekenis (semantiek) tussen die partijen en systemen wordt afgestemd. Deze afspraken worden samengebracht in OWMS.

Het Technisch Framework bestaat uit een nauwkeurige beschrijving van de syntactische afspraken. Het vormt als het ware de spellingsregels en de grammatica voor OWMS. De taal met de meeste voordelen om metadata vast te leggen en uit te wisselen is XML. XML is breed toegepast, platform-onafhankelijk en biedt voldoende structuur om nauwkeurige regels voor te formuleren. Het technisch framework is dan ook gedefinieerd in XML.

OWMS is een algemene overheidsbrede standaard. Voor specifieke domeinen, ook wel informatiecollecties genoemd, kan het nodig zijn om specifieke beperkingen of uitbreidingen op OWMS te maken. De manier waarop informatie van een collectie van metadata voorzien wordt leggen we vast in een Informatie Publicatie Model (IPM). Dit wordt ook wel een toepassing van OWMS op een informatiecollectie genoemd.

Dit document is primair bedoeld om uit te leggen hoe een formele definitie van een IPM gemaakt kan worden conform de OWMS syntax voor XML.

Aansluiting van een collectie op OWMS betekent dat de collectie op de OWMS-elementen ontsloten kan worden, omdat deze aanwezig zijn en OWMS-waarden hebben. Er zijn twee manieren waarop een collectie OWMS kan implementeren: in XML en in HTML.

Voor XML is een zogenaamd technisch framework gebouwd, bestaande uit een structuur van xsd's, schematron regels en een template.xsd waarmee een XML-schema voor de eigen collectie kan worden gemaakt. De collectie wordt dan (mede) in XML ontsloten, waarbij de XML voldoet aan een OWMS-conform schema.

In HTML wordt OWMS opgenomen als metatags. Het is dan echter niet mogelijk de aanvullende restricties en structuur te reguleren. Hoe OWMS met behulp van HTML moet worden geïmplementeerd wordt beschreven in de laatste paragraaf.

Het technisch framework OWMS stelt leveranciers van content in staat om applicaties te bouwen waarmee die content volgens OWMS van metadata kan worden voorzien. Het kan bovendien codeerschema's leveren die als waardelijst in een applicatie kan worden gebruikt en er kan tegen het framework gevalideerd worden.

2 Uitgangspunten

Het framework voldoet aan de volgende uitgangspunten:

- Het moet samenhang brengen tussen informatie uit verschillende informatiecollecties. Deze samenhang wordt bereikt doordat de metadata in alle collecties dezelfde gemeenschappelijke eigenschappen gebruiken en gemeenschappelijke waarden.
- Het dient voor de gebruikers (zij die collecties ontsluiten en zij die de content aanleveren) eenvoudig zijn om het framework toe te passen binnen hun eigen collectie.

Dit heeft het volgende tot gevolg:

- De voorgeschreven syntax moet de markup van de contentmetadata minimaliseren.
- De waardebereiken moeten eenvoudig uitbreidbaar zijn.
- Er moet standaard een validatieservice geleverd worden.
- De validatie moet liberaal zijn in acceptatie van content. (Postel's law)

Bij het realiseren van deze eisen in een technische oplossing maken we pragmatische keuzen.

Het technisch framework definieert de syntax van:

- De OWMS-kern: een set van acht verplichte eigenschappen.
- De OWMS-mantel: een ruime set van optionele eigenschappen.
- Codeerschema's voor deze eigenschappen (zijnde: syntax codeerschema of een gecontroleerde waardelijst)

Een toepassing heet OWMS-conform te zijn als het de OWMS-kern overneemt volgens de semantische definitie van OWMS en de implementatie volgens het technisch framework. Eventueel mag dit worden uitgebreid met nul of meer eigenschappen uit de OWMS-mantel of de Dublin Core set. Daarnaast staat het een toepassing vrij om eigen eigenschappen of waardebereiken te definiëren.

2.1 Dublin Core

Net als voor de semantiek is voor de opzet van het technisch framework allereerst onderzocht wat de ontwikkelingen op dat vlak bij de Dublin Core architecten op dit moment brengen. De DC-Architecten hebben in 2005 een conceptueel model voor Dublin Core metadata gedefinieerd: het Dublin Core Abstract Model (DCAM). Het DCAM kan worden vertaald naar verschillende implementatietalen. De DC-Architecten zijn bezig met vertalingen naar XML. Omdat deze specificaties door het DCMI nog in ontwikkeling zijn en we voor OWMS op korte termijn een technisch framework nodig hebben, maken we hierin onze eigen keuzes. Een andere reden om niet op de specificatie van Dublin Core te wachten is dat deze neigen naar te grote vrijheid aan de kant van de contentleverancier. Hierdoor kan het voornaamste doel van OWMS in gevaar komen, het creëren van samenhang tussen verschillende collecties van overheidsinformatie. De Dublin Core specificaties leiden bovendien tot een syntax die weliswaar academisch correct is, maar die ook moeilijk te begrijpen is. Dit vormt een belemmering voor de acceptatie en correcte implementatie van OWMS door overheidsorganisaties.

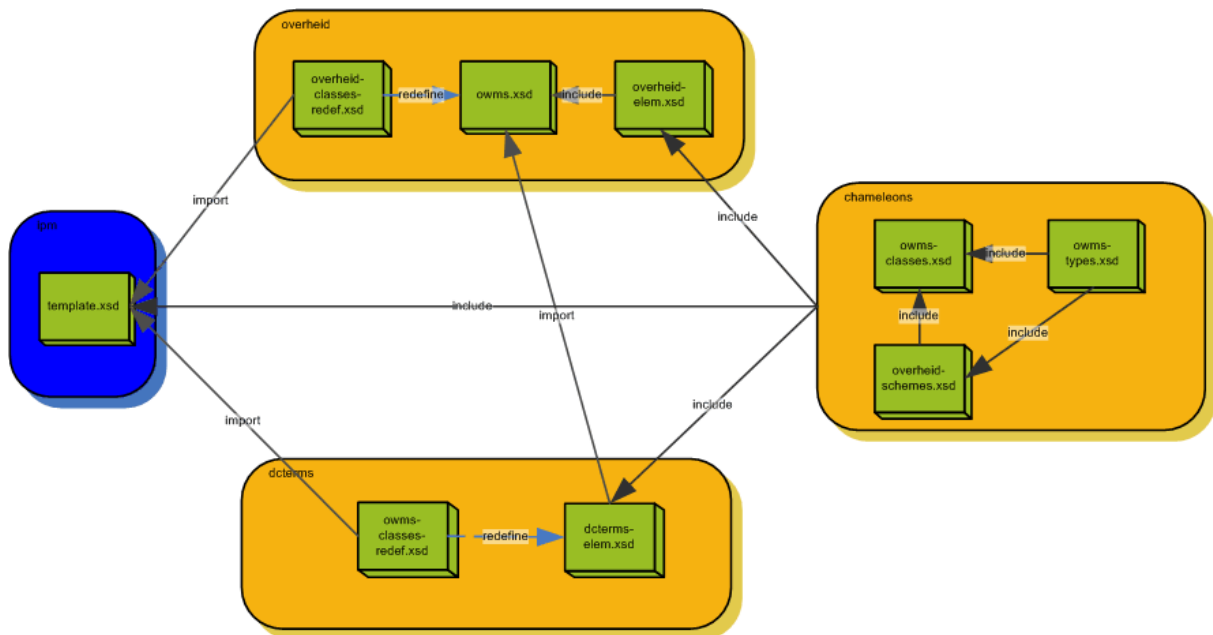
3 XML-architectuur

Een collectie kan aansluiten bij OWMS door gebruik te maken van de xsd-structuur, zoals die beschikbaar is op de OWMS-server. De collectie maakt een eigen template.xsd, waarin elementen van de verschillende basisxsd's worden geïmporteerd. Hiermee wordt afgedwongen dat de template voldoet aan OWMS. Daarnaast staat het de collectie vrij aan deze template eigen elementen en waarden toe te voegen.

De xml-architectuur kent drie pijlers:

- Een **centrale laag** waarin alle bouwstenen van OWMS zijn gedefinieerd. Er zijn vier componenten:
 - o Cameleons (namespacevrij): drie xsd-schema's waarin een aantal typen (owms-types), klassen (owms-classes) en waarden (owms-schemes) zijn gedefinieerd die in de verschillende namespaces hergebruikt (en geherdefinieerd) kunnen worden.
 - o Dcterms-elem met daarin de elementen die vanuit DC zijn overgenomen in OWMS, met als namespace dcterms (<http://purl.org/dc/terms/>)
 - o Overheid-elem met daarin de elementen die aanvullend zijn gedefinieerd in OWMS om met de metadatering beter aan te kunnen sluiten bij de specialistische context van overheidsinformatie (bijv. authority), met als namespace overheid (<http://metadata.overheid.nl/owms/terms/>)
 - o owms-elem, het centrale xsd, waarin de elementen uit overheid-elem en dcterms-elem zijn opgenomen en verdeeld in een kerngroep van acht en een mantelgroep voor de overige elementen
- Een **redefinelaag** waarin de classes uit de centrale laag opnieuw worden aangeboden, zodanig dat zij voor deze specifieke collectie geherdefinieerd kunnen worden. Het is bijvoorbeeld mogelijk extra waarden toe te staan voor standaardelementen of juist waarden niet toe te staan. Er zijn twee componenten:
 - o owms-classes-redef (namespace dcterms)
 - o overheid-classes-redef (namespace overheid)

- De **collectielaag** waarin de template van de betreffende collectie is opgenomen, die de xsd's uit de redefinelaag omvat en – voor zover nodig – de xsd's uit de centrale laag. Er is één component:
 - o template.xsd: het xml-schema van de collectie (collectie-eigen namespace)



4 Aansluiten bij de xml-architectuur

Om een XML-collectie aan te laten sluiten bij de OWMS-architectuur moet deze voldoen aan collectiespecifieke xsd die gebaseerd is op de template.xsd van OWMS. Hieronder wordt beschreven hoe deze template.xsd kan worden omgevormd in een eigen xsd.

4.1 Collectie-eigen xsd aanmaken

- a. Kopieer template.xsd en hernoem naar collectienaam.xsd

van	template.xsd
naar	{ipmnaam}.xsd
voorbeeld	vac.xsd

- b. Hernoem de template namespace naar collectie namespace

van	xmlns="http://standaarden.overheid.nl/template/"targetNamespace="http://standaarden.overheid.nl/template/"
naar	xmlns="http://standaarden.overheid.nl/{ipmnaam}/"targetNamespace="http://standaarden.overheid.nl/{ipmnaam}/"
voorbeeld	xmlns=http://standaarden.overheid.nl/vac/ targetNamespace=http://standaarden.overheid.nl/vac/

- c. Geef een versienummer aan het vocabularium

van	version="X.X"
naar	version="{X.X}"
voorbeeld	version="1.0"

d. Hernoem in de xsd de volgende elementen:

- root-elem

van	<root-elem>
naar	<{ipmnaam}>
voorbeeld	<vac>

- indien gewenst, templatemetadate

e. Voeg het collectie-eigen xsd model toe in het element body en stel body verplicht

van	<xs:element ref="body" minOccurs="0"/>
naar	<xs:element ref="body"/>
voorbeeld	<xs:element name="body"> <xs:complexType> <xs:sequence> <xs:element name="content" type="xs:string"/> </xs:sequence> </xs:complexType> </xs:element>

Het collectienaam.xsd bevat nu de OWMS-kernelementen en de collectie-xsd.

4.2 Toevoegen andere metadata

Naast OWMS-kern metadata kan er nog andere metadata worden toegevoegd:

- Uit de dcterms/overheid namespace (OWMS-mantel)
Om alle elementen optioneel op te nemen kan het beste de referentie naar de owms-mantel geuncomment worden. Om alleen specifieke elementen op te nemen, kan het beste worden gerefereerd, volgens <xs:element ref="dcterms:valid"/>.

van	<!-- <xs:group ref="overheid:owmsmantel" minOccurs="0" maxOccurs="unbounded"/> -->
naar	<xs:group ref="overheid:owmsmantel" minOccurs="0" maxOccurs="unbounded"/>
voorbeeld	<xs:element ref="dcterms:valid"/>

- Collectieeigen metadata
Eigen elementen worden opgenomen onder 'templatemeta'. Hernoem dit element, stel het verplicht en voeg het model met eigen elementen toe.

van	<xs:element name="templatemeta" minOccurs="0">
naar	<xs:element name="{ipmnaam}meta" >
voorbeeld	<xs:complexType> <xs:sequence> <xs:element name="prioriteit"> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="laag"/> <xs:enumeration value="medium"/> <xs:enumeration value="hoog"/> </xs:restriction> </xs:simpleType> </xs:element>

- Metadata uit andere namespaces (bijv. Inspire)

van	<pre><xs:element name="othermeta" minOccurs="0"> <xs:complexType> <xs:sequence> <!-- TO DO (optional): add other metadata like for instance inspire. --> </xs:sequence> </xs:complexType> </xs:element></pre>
naar	<pre><xs:element name="othermeta" minOccurs="0"> <xs:complexType> <xs:sequence> <xs:element ref="foaf:topic"/> <!-- TO DO (optional): add other metadata like for instance inspire. --> </xs:sequence> </xs:complexType> </xs:element></pre>
voorbeeld	<pre><othermeta> <foaf:topic>Test</foaf:topic> </othermeta></pre>

4.3 Wijzigen waardenbereik

Tot slot is het nog mogelijk om de waardebereiken van de OWMS controlled vocabularies aan te passen. De werkwijze is afhankelijk van de namespace waarin het bijbehorende element zich bevindt:

- namespace dcterms

Kopieer owms-classes-redef.xsd lokaal en laat de schemalocation pointer in de collectienaam.xsd de wijzen naar deze lokale file.

van	<pre><xs:import namespace="http://standaarden.overheid.nl/owms/terms/" schemaLocation="http://standaarden.overheid.nl/owms/3.5/xsd/owms- classesredef.xsd"/></pre>
naar	<pre><xs:import namespace="http://standaarden.overheid.nl/owms/terms/" schemaLocation="owms-classes-redef.xsd"/></pre>

Pas de enumeraties in deze lokale file aan.

van	<pre><xs:simpleType name="LocationScheme"> <xs:restriction base="LocationScheme"> <xs:enumeration value="overheid:PostcodeHuisnummer"/> <xs:enumeration value="overheid:Gemeente"/> <xs:enumeration value="overheid:Provincie"/> <xs:enumeration value="overheid:Waterschap"/> </xs:restriction> </xs:simpleType></pre>
naar	<pre><xs:simpleType name="LocationScheme"> <xs:restriction base="LocationScheme"> <xs:enumeration value="overheid:PostcodeHuisnummer"/> <xs:enumeration value="overheid:Gemeente"/> </xs:restriction> </xs:simpleType></pre>

- namespace overheid

Kopieer overheid-classes-redef.xsd lokaal en laat de schemalocation pointer in de collectienaam.xsd de wijzen naar deze lokale file.

van	<code><xs:import namespace="http://standaarden.overheid.nl/owms/terms/" schemaLocation="http://standaarden.overheid.nl/owms/3.5/xsd/overheid-classesredef.xsd"/></code>
naar	<code><xs:import namespace="http://standaarden.overheid.nl/owms/terms/" schemaLocation="overheid-classes-redef.xsd"/></code>

Pas de enumeraties in deze lokale file aan.

van	<code><xs:simpleType name="BeslissendOrgaanScheme"> <xs:restriction base="BeslissendOrgaanScheme"> <xs:enumeration value="overheid:BestuursorgaanDeelgemeente"/> <xs:enumeration value="overheid:BestuursorgaanGemeente"/> <xs:enumeration value="overheid:BestuursorgaanMinisterie"/> <xs:enumeration value="overheid:BestuursorgaanProvincie"/> <xs:enumeration value="overheid:BestuursorgaanRegionaalSamenwerkingsorgaan"/> <xs:enumeration value="overheid:BestuursorgaanWaterschap"/> </xs:restriction> </xs:simpleType></code>
naar	<code><xs:simpleType name="BeslissendOrgaanScheme"> <xs:restriction base="BeslissendOrgaanScheme"> <xs:enumeration value="overheid:BestuursorgaanDeelgemeente"/> <xs:enumeration value="overheid:BestuursorgaanGemeente"/> </xs:restriction> </xs:simpleType></code>

- namespace eigen collectie

Vervang in collectienaam.xsd de include door een redefine en herdefinieer het waardebereik.

van	<code><xs:include schemaLocation="http://standaarden.overheid.nl/owms/3.5/xsd/overheidschemes.xsd"/></code>
naar	<code><xs:redefine schemaLocation="http://standaarden.overheid.nl/owms/3.5/xsd/overheid-schemes.xsd"> <xs:simpleType name="AgentScheme"> <xs:restriction base="AgentScheme"> <xs:enumeration value="overheid:Adviescollege"/> </xs:restriction> </xs:simpleType> </xs:redefine></code>

Er kunnen waarden worden weggelaten, toegevoegd of een combinatie van beide.

- Inperken van het waardebereik

van	<code><xs:simpleType name="BeslissendOrgaanScheme"> <xs:restriction base="BeslissendOrgaanScheme"> <xs:enumeration value="overheid:BestuursorgaanDeelgemeente"/> <xs:enumeration value="overheid:BestuursorgaanGemeente"/> <xs:enumeration value="overheid:BestuursorgaanMinisterie"/> <xs:enumeration value="overheid:BestuursorgaanProvincie"/> <xs:enumeration value="overheid:BestuursorgaanRegionaalSamenwerkingsorgaan"/> <xs:enumeration value="overheid:BestuursorgaanWaterschap"/> </xs:restriction> </xs:simpleType></code>
naar	<code><xs:simpleType name="BeslissendOrgaanScheme"> <xs:restriction base="BeslissendOrgaanScheme"></code>

	<pre><xs:enumeration value="overheid:BestuursorgaanDeelgemeente"/> <xs:enumeration value="overheid:BestuursorgaanGemeente"/> <xs:enumeration value="overheid:BestuursorgaanMinisterie"/> </xs:restriction> </xs:simpleType></pre>
--	---

- Uitbreiden van het waardenbereik

van	<pre><xs:simpleType name="BeslissendOrgaanScheme"> <xs:restriction base="BeslissendOrgaanScheme"> <xs:enumeration value="overheid:BestuursorgaanDeelgemeente"/> <xs:enumeration value="overheid:BestuursorgaanGemeente"/> <xs:enumeration value="overheid:BestuursorgaanMinisterie"/> <xs:enumeration value="overheid:BestuursorgaanProvincie"/> <xs:enumeration value= "overheid:BestuursorgaanRegionaalSamenwerkingsorgaan"/> <xs:enumeration value="overheid:BestuursorgaanWaterschap"/> </xs:restriction> </xs:simpleType></pre>
naar	<pre><xs:simpleType name="BeslissendOrgaanScheme"> <xs:restriction base="BeslissendOrgaanScheme"> <xs:enumeration value="overheid:BestuursorgaanDeelgemeente"/> <xs:enumeration value="overheid:BestuursorgaanGemeente"/> <xs:enumeration value="overheid:BestuursorgaanMinisterie"/> <xs:enumeration value="overheid:BestuursorgaanProvincie"/> <xs:enumeration value= "overheid:BestuursorgaanRegionaalSamenwerkingsorgaan"/> <xs:enumeration value="overheid:BestuursorgaanWaterschap"/> </xs:restriction> <xs:enumeration value= "{ipmnamespaceprefix}:BeslissendOrgaanBuurtComité"/> </xs:simpleType></pre>
voorbeeld	<pre><xs:simpleType name="BeslissendOrgaanScheme"> <xs:restriction base="BeslissendOrgaanScheme"> <xs:enumeration value="overheid:BestuursorgaanDeelgemeente"/> <xs:enumeration value="overheid:BestuursorgaanGemeente"/> <xs:enumeration value="overheid:BestuursorgaanMinisterie"/> <xs:enumeration value="overheid:BestuursorgaanProvincie"/> <xs:enumeration value= "overheid:BestuursorgaanRegionaalSamenwerkingsorgaan"/> <xs:enumeration value="overheid:BestuursorgaanWaterschap"/> </xs:restriction> <xs:enumeration value= "qa:BeslissendOrgaanBuurtComité"/> </xs:simpleType></pre>

- Combinatie van beide (inperken en uitbreiden waardenbereik)

van	<pre><xs:simpleType name="BeslissendOrgaanScheme"> <xs:restriction base="BeslissendOrgaanScheme"> <xs:enumeration value="overheid:BestuursorgaanDeelgemeente"/> <xs:enumeration value="overheid:BestuursorgaanGemeente"/> <xs:enumeration value="overheid:BestuursorgaanMinisterie"/> <xs:enumeration value="overheid:BestuursorgaanProvincie"/> <xs:enumeration value= "overheid:BestuursorgaanRegionaalSamenwerkingsorgaan"/> <xs:enumeration value="overheid:BestuursorgaanWaterschap"/> </xs:restriction> </xs:simpleType></pre>
naar	<pre><xs:simpleType name="BeslissendOrgaanScheme"> <xs:restriction base="BeslissendOrgaanScheme"></pre>

	<pre> <xs:enumeration value="overheid:BestuursorgaanDeelgemeente"/> <xs:enumeration value= "overheid:BestuursorgaanRegionaalSamenwerkingsorgaan"/> <xs:enumeration value="overheid:BestuursorgaanWaterschap"/> </xs:restriction> <xs:enumeration value= "{ipmnamespaceprefix}:BeslissendOrgaanBuurtComité"/> </xs:simpleType> </pre>
voorbeeld	<pre> <xs:simpleType name="BeslissendOrgaanScheme"> <xs:restriction base="BeslissendOrgaanScheme"> <xs:enumeration value="overheid:BestuursorgaanDeelgemeente"/> <xs:enumeration value= "overheid:BestuursorgaanRegionaalSamenwerkingsorgaan"/> <xs:enumeration value="overheid:BestuursorgaanWaterschap"/> </xs:restriction> <xs:enumeration value="qa:BeslissendOrgaanBuurtComité"/> </xs:simpleType> </pre>

4.4 Verificatie gebruikte waarden

Teneinde de waarden uit de cv's te kunnen verifiëren, wordt gebruik gemaakt van een aanvullend schematronbestand. Hiermee kan voor een XML-instance voor elementen uit OWMS met een gecontroleerd waardenbereik, worden vastgesteld of:

- de waarde voorkomt in de actuele versie van de betreffende controlled vocabulary
- de waarde van `overheid:postcodeHuisnummer` voldoet aan de gedefinieerde reguliere expressie
- voor datumvelden geldt: `startdatum <=einddatum`

Om schematron toe te passen, kan gebruik worden gemaakt van een schematron-template, analoog aan de xsd-template. Hierin is voor alle OWMS-elementen voor zover mogelijk de validatie gerealiseerd. Daarnaast kunnen er nog collectiespecifieke validatieregels worden toegevoegd. Dit sch-bestand wordt vervolgens met de iso schematron xsl samengevoegd tot een xsl voor deze specifieke collectie. Deze transformatie resulteert in een errorlijst.

Omdat er op dit moment nog geen open source schema aware schematron processor is, zijn er twee versies van de schematron template beschikbaar: een non schema aware en een schema aware. Deze laatste is beheersmatig een stuk eenvoudiger en bovendien beter leesbaar, omdat daarin gebruik kan worden gemaakt van de definities uit de xsd's. De eerste kent geen koppeling met de xsd's en somt daarom apart alle elementen op waarvoor validatieregels gelden.

Saxon-SA is een processor die schema aware aankant, voor non schema aware is de open source variant Saxon-B beschikbaar.

Schematron kan op de volgende wijze aan een collectie worden toegevoegd:

- schema aware
Kopieer `template_sa.sch` en hernoem naar `collectienaam_sa.sch`. Definieer de namespaceprefix voor de collectienamespace. Importeer vervolgens de `collectiexsd` in de sch-file. Voeg, voor zover nodig, schematronregels toe voor elementen/attributen uit de eigen collectie.
- non schema aware
Kopieer `template_nsa.sch` en hernoem naar `collectienaam_nsa.sch`. Definieer de namespaceprefix voor de collectienamespace. Voeg, voor zover nodig, schematronregels toe voor elementen/attributen uit de eigen collectie.

4.5 XML-parsers

De XML-architectuur werkt o.a. binnen de volgende parsers:

- Xerces-J
- Saxon SA / Saxon B
- MSXML 6.0

4.6 Offline aansluiten bij OWMS-XML

Bij het aanmaken van een IPM eigen schema op basis van het owms 3.5 template.xsd is het de bedoeling dat de ondersteunende xsd files (dcterms-elem.xsd, owms.xsd, overheid-elem.xsd, owms-types.xsd, overheid-schemes.xsd, overheid-classes.xsd) gelezen blijven worden van de server (<http://standaarden.overheid.nl/owms/3.5/xml/>). De schema optimalisaties in OWMS zijn backwards compatible, zodat altijd gebruik gemaakt kan worden van de laatste versie.

Mocht het nodig zijn het schema ook offline te kunnen gebruiken, dan worden alle xsd files lokaal gekopieerd en de paden tussen de verschillende files lokaal aangepast. Er wordt geadviseerd voor deze mapping gebruik te maken van een zgn. catalog file. Hiermee kan worden geregeld dat wanneer er verbinding mogelijk is, er gebruik wordt gemaakt van de online versie en bij offline de lokale kopieën worden aangeroepen, als fall back. Bovendien kan dan de mappenstructuur volledig worden overgenomen, zonder dat deze handmatig hoeft te worden aangepast. De meeste XML parsers bieden ondersteuning voor dit OASIS XML catalog mechanisme.

In de catalogus, catalog.xml, worden de mappings als volgt gedefinieerd:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE catalog PUBLIC "-//OASIS//DTD XML Catalogs V1.1//EN"
"http://www.oasis-open.org/committees/entity/release/1.1/catalog.dtd">
<catalog xmlns="urn:oasis:names:tc:entity:xmlns:xml:catalog">

<rewriteURI uriStartString=http://standaarden.overheid.nl/owms/3.5/xsd/
rewritePrefix="file:///LocalSchemes/xsd/" />
<!--bij rewritePrefix wordt het lokale pad ingevoerd -->
</catalog>
```

Hiermee wordt geregeld dat alle paden die beginnen met <http://standaarden.overheid.nl/owms/3.5/xsd/> geremapt moeten worden naar <file:///LocalSchemes/xsd/>

De XML-catalogspecificatie biedt nog meer mappingmogelijkheden. Zie voor een volledige behandeling: <http://www.oasis-open.org/committees/entity/spec-2001-08-06.html>.

4.7 Geen automatische validatie in 3.5

Het technisch framework voorziet in OWMS 3.5 nog niet in een volledig operationele validatieservice voor XML- berichten en XHTML-documenten. Het is voor contentleveranciers wel mogelijk op basis van het technisch framework hun eigen berichten te valideren en documenten te controleren. In de praktijk zullen zij dit doen met een IPM-schema, zoals bijvoorbeeld vac.xsd. Met een validatieservice kunnen XML-berichten gevalideerd worden tegen OWMS.

5 HTML-syntax

De HTML-syntax voor OWMS 3.5 gaat uit van de geldende HTML-syntax van versie 3.0. In de header van een HTML-document wordt een link-tag opgenomen die de namespaces declareert, in XHTML is de conventie de naam van een namespace met hoofdletters te schrijven.

```
<link rel="schema.DCTERMS" href="http://purl.org/dc/terms/" />
<link rel="schema.OVERHEID" href="http://overheid.nl/owms/terms/" />
```

Een OWMS-conforme metadatatag in XHTML heeft de volgende syntax:

```
<meta name="[EIGENSCHAP]" scheme="[CODEERSHEMA]" value="[WAARDE]" />
```

Een eigenschap of codeerschema wordt aangeduid met een naam bestaande uit namespace, dubbele punt, term voor de eigenschap of schema.

Voorbeeld

```
<meta name="DCTERMS.type" scheme="OVERHEID.Informatietype" value="vragen" />
```