

OSLO UML2RDF OMZETTINGSREGELS

Versie /// 0.1

Publicatiedatum /// 17/9/2018

Documenthistoriek:

Versie	Opmerking	Datum	Auteur	Status
0.1		17/9/2018	Geert Thijs	Draft

informatie.vlaanderen@vlaanderen.be

////////////////////////////////////

INHOUD

Inhoud.....	3
1 Inleiding	4
2 Aanpak	5
2.1 Mapping.....	5
2.2 UML-first.....	5
2.3 Best practices.....	6
2.4 MDA	6
3 Gemeenschappelijke elementen	7
4 Omzetting UML2RDF	8
4.1 Identificatie.....	8
4.2 Package -> Ontology	10
4.3 Class -> OWLCLASS	10
4.4 Tagged values	10
5 Referenties.....	11

////////////////////////////////////

1 INLEIDING

OSLO gebruikt UML om de conceptuele en applicatiemodellen die het ontwikkelt te beschrijven, meer info daarover is te vinden in de OSLO-modelleringsregels (Thijs, 2018). Een model bestaat uit elementen zoals klassen, attributen, associaties etc en wordt grafisch voorgesteld dmv een UML-klassendiagram.

Bijkomend echter moeten de OSLO modellen worden beschreven in RDFS/OWL. Dat laat toe om data te combineren met de betekenis van de data (betekenis die door het model wordt voorgesteld). Dit is een basisprincipe van het semantisch web en maakt data op het web makkelijker vindbaar en interpreteerbaar.

RDFS en OWL zijn vocabularia met elementen om modellen (resp schema's en ontologieën) te beschrijven, zie (W3C, RDF Schema 1.1, 2014) en (W3C, OWL 2 Web Ontology Language, 2012). Het resultaat wordt genoteerd in RDF, een taal waarmee niet alleen modellen maar alle resources op het semantisch web worden beschreven.

De verzameling elementen die RDFS & OWL gebruiken om een model te documenteren is analoog aan die van UML maar stemt er toch niet 100% mee overeen. Vandaar dit document dat uitlegt welke UML-elementen we bij OSLO mappen op welke RDFS/OWL elementen.

De omzetting naar RDFS/OWL gebeurt op het niveau van conceptueel model (het zgn vocabularium of VOC). Een applicatiemodel (applicatieprofiel of AP) steunt echter op één of meerdere VOC's, dus impliciet geldt de omzetting naar RDFS/OWL ook voor de AP's.

[illegible]

2 AANPAK

De omzetting van UML naar RDFS/OWL steunt op volgende principes:

- Mapping op bestaande termen.
- Aan het UML-model wordt niet geraakt.
- Beste praktijken.
- Model Driven Architecture.

2.1 MAPPING

Belangrijk bij het opstellen van een model is het hergebruik van bestaande concepten. Als een concept al gedefinieerd is moet dat niet opnieuw gebeuren. Men kan bij publicatie van een nieuwe datastandaard verwijzen naar de bestaande standaarden waarop men zich baseert, maar op het semantisch web gebeurt zo iets op een formele manier. Niet alleen wordt het model beschreven in RDF/OWL en daarna gepubliceerd op het web in RDF, elk element van het model krijgt ook een URI als identicator. Zo kan een model-element en zijn betekenis steeds worden opgezocht en op dezelfde manier kan verwezen worden naar reeds bestaande elementen.

Om dit mogelijk te maken moet meta-informatie aan het model worden toegevoegd, de zgn mapping. Bv hergebruikt het VOC OSLO-Organisatie het concept PubliekeOrganisatie van ISA. ISA publiceert zijn modellen ook in RDFS/OWL en dit concept kreeg daar volgende URI: <http://data.europa.eu/m8g/PublicOrganisation>. Om meta-informatie aan een modelement (in dit geval een klasse) toe te voegen voorziet UML “tagged values”. OSLO maakt van dit mechanisme gebruik om een tag “uri” aan een modelement toe te voegen.

Sommige tags worden sowieso al verplicht opgelegd door de OSLO-modelleringsregels. Naast de uri-tag is er bv ook nog een tag “definition” (met de definitie vh modelement), een tag “usageNote” (met informatie over het gebruik vh concept) en een tag “label” (met de mensleesbare naam vh element). Meer daarover in de OSLO-modelleringsregels, zie (Thijs, 2018).

Echter niet alle tags dienen om de omzetting van UML naar RDFS/OWL te sturen, in dit document sommen we enkel de tags op die dat wel doen. Deze tags worden gedocumenteerd bij de beschrijving van specifieke omzetting (bv van Class naar OWLClass) of vermeld in het algemeen overzicht in §4.4.

2.2 UML-FIRST

Het tweede principe houdt in dat men het UML-model opstelt los van de omzetting ervan naar RDFS/OWL. Het model beschrijft een bepaald domein en het is niet de bedoeling dat semantiek daardoor verandert of verloren gaat. Zo is het bv niet nodig om een associatie voor te stellen door een attribuut omdat dat in RDFS/OWL wel zo is of om moet vermeden worden om een associatieklasse te modelleren als een veel-veel relatie omdat associatieklassen in RDFS/OWL niet bestaan..

Voor wie een substantiële mismacht verwacht tussen UML en RDF: in praktijk zijn er weinig elementen van het UML-metamodel voor klassen die niet omgezet kunnen worden in RDFS/OWL, bijna voor alle elementtypes is een oplossing voorhanden. Wel zijn daar soms tags voor nodig, bv om een associatieklasse correct in

3 GEMEENSCHAPPELIJKE ELEMENTEN

Volgende tabel geeft een overzicht van elementen in UML en het overeenstemmend element in RDFS/OWL.

[illegible]

Opmerking: de termen gebruikt voor OWL/RDFS-elementen zijn deze uit het Ontology Definition Metamodel vd OMG (zie §4).

////////////////////////////////////

4 OMZETTING UML2RDF

We illustreren de omzetting van UML- naar RDFS/OWL-elementen met voorbeelden. Voor een beter begrip worden deze grafisch voorgesteld in UML. Voor RDFS/OWL is dit mogelijk dankzij het Ontology Definition Metamodel (ODM) en het bijbehorende UML-profiel, zie (OMG, 2014). Naast de grafische weergave tonen we ook hoe het resultaat vd omzetting eruit ziet in RDF.

Opmerking: De tool waarmee de illustraties werden vervaardigd is Enterprise Architect (EA) van SPARX. ODM is daarin maar gedeeltelijk geïmplementeerd en dus gebruikten we de profielen verstrekt door de OMG zelf, zie (OMG, 2014).

4.1 IDENTIFICATIE

REGEL: Identificeer het element dmv een URI als volgt:

- Voor een eigen concept: Combineer de waarde vd package-tag “baseURI” met deze vd element tag “name” of indien deze ontbreekt met de lokale naam vh element.
- Voor een bestaand concept: Haal de URI vh element waarmee het betrokken element overeenstemt in een bestaand vocabularium uit de tag “uri”.

Vb eigen concept zonder name-tag:

UML	RDFS/OWL
<pre> classDiagram package MijnDomein { class OSLO-Adres { tags } class Adres } OSLO-Adres "tags" : https://data.vlaanderen.be/ns/adres# </pre>	<pre> classDiagram class MijnDomein { class Adres { <<owlClass>> tags } } Adres "tags" : https://data.vlaanderen.be/ns/adres#Adres </pre>

De URI van de owl-klasse Adres is `https://data.vlaanderen.be/ns/adres#Adres`, een combinatie van `https://data.vlaanderen.be/ns/adres#` & `Adres`, resp de waarde vd package-tag “baseURI” en de lokale naam vd klasse in UML.

In RDF-turtle:

```
<http://data.vlaanderen.be/ns/adres#Adres> a owl:Class.
```

////////////////////////////////////

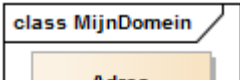
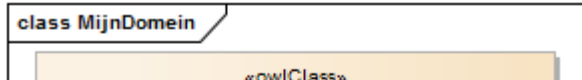
4.2 PACKAGE -> ONTOLOGY

()

4.3 CLASS -> OWLCLASS

REGEL: Zet een UML-klasse om in een owlClass.

Vb nieuwe klasse:

UML	RDFS/OWL
 <pre> classDiagram class MijnDomein { class Adres } </pre>	 <pre> classDiagram class MijnDomein { class Adres { identifier = https://data.vlaanderen.be/ns/adres#Adres tag } } </pre>

De UML-klasse Adres wordt een owl-klasse Adres.

In RDF-turtle:

```
<http://data.vlaanderen.be/ns/adres#Adres> a owl:Class.
```

Voor de samenstelling vd URI vd owl-klasse: zie §4.1.

Klassen met een uri-tag (aka klassen die al bestaan op het semantisch web) moeten niet worden omgezet, meer info hierover in §4.1.

4.4 TAGGED VALUES

()

[illegible]

5 REFERENCES

- Cranefield, S. (sd). *UML and the Semantic Web*.
- De Paepe, D. e. (2016). *Automated UML-Based Ontology Generation in OSLO*. Opgehaald van <https://biblio.ugent.be/publication/8531758/file/8531761.pdf>
- INSPIRE. (2017). *Guidelines for the RDF encoding of spatial data*. Opgehaald van http://inspire-eu-rdf.github.io/inspire-rdf-guidelines/#ref_cr_prop_multiplicity
- ISA. (sd). *Core Public Service Vocabulary*. Opgehaald van Core Public Service Vocabulary
- ISO 191150-2. (2015). *Rules for developing ontologies in the Web Ontology Language*.
- OMG. (2014). *Model Driven Architecture (MDA), MDA Guide rev 2.0*. Opgehaald van <https://www.omg.org/cgi-bin/doc?ormsc/14-06-01>
- OMG. (2014). *ODM Profiles*. Opgehaald van <https://www.omg.org/spec/ODM/About-ODM/>
- OMG. (2014). *Ontology Definition Metamodel*. Opgehaald van <file:///C:/Users/thijsge/Downloads/formal-14-09-02.pdf>
- Thijs, G. (2018). *OSLO modelleringsregels*.
- W3C. (2012). *OWL 2 Web Ontology Language*. Opgehaald van <https://www.w3.org/TR/owl2-overview/>
- W3C. (2014). *RDF Schema 1.1*. Opgehaald van <https://www.w3.org/TR/rdf-schema/>