Integratie van namespaces

Ontwerp voor het integreren van namespaces in Fresnel Forms.

|  |
| --- |
| **Alex Mekkering** |
| Revisiehistorie |

|  |  |
| --- | --- |
| **Datum** | **Beschrijving** |
| 2015-01-11 | Initiële versie |
| 2015-01-17 | Ontwerp bijgewerkt |
| 2015-02-20 | Revisiehistorie bijgewerkt |
| 2015-05-13 | Uiteindelijke versie |

# Doel

Het doel van dit document is om te bepalen hoe we namespaces van elementen uit de bronontologie kunnen integreren in Fresnel Forms. Het is hierbij uitdrukkelijk niet de bedoeling om het huidige ontwerp in zijn geheel te herzien; slechts de delen die belang zijn voor het correct integreren van URI namespaces worden aangepakt, mede om de tijdsbesteding voor deze taak niet uit de hand te laten lopen.

# Inleiding

Op dit moment wordt vanuit de bronontologie bepaald voor welke OWL Classes en Properties lenzen dienen te worden gecreëerd op de gebruikersinterface. Hierbij wordt echter bij de benaming van zowel de lenzen als de properties geen rekening gehouden met de namespace van de betreffende Class of Property.

Wanneer verschillende properties dezelfde naam hebben, maar een andere namespace, kunnen deze hierdoor niet van elkaar onderscheiden worden op de GUI. Door rekening te houden met de namespace van een Class of Property, kan dit onderscheid wel gemaakt worden aangezien de namespace en de naam (fragment identifier) van een resource de bijbehorende URI vormen en de URI van een RDF resource altijd uniek is. (The Internet Engineering Task Force, 2015)

In dit document wordt, na een korte uitleg over het toepassen van prefixes, eerst een analyse uitgevoerd van de huidige implementatie waarin de problemen erin worden benoemd. Daarna wordt een oplossing voor deze problemen geboden in de vorm van een aangepast ontwerp en uitleg over het dynamisch bepalen van namen op basis van URI’s.

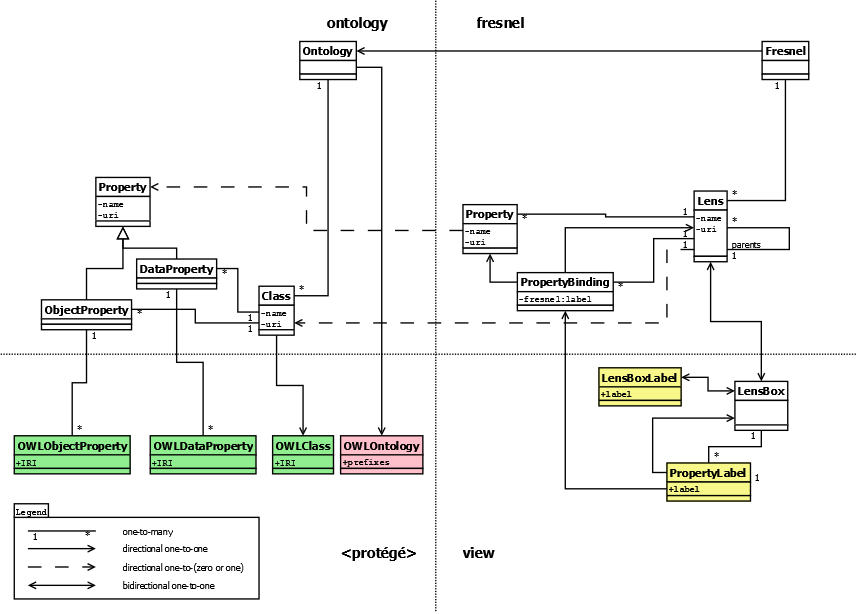
## Het gebruik van prefixes

Volledige URI’s hebben meestal een lengte die lastig weer te geven is in een gebruikersinterface. Binnen de meeste ontologiëen zijn echter ook prefixes aanwezig welke een verkorte weergave definiëren voor de namespace van een URI. Door, in plaats van de volledige namespace, deze prefixes te gebruiken, blijven namen op de gebruikersinterface kort en overzichtelijk.

Vervolgens kan eventueel voor resources die binnen de default namespace (:) vallen, gekozen worden om de dubbele punt niet te tonen. Hierdoor blijft de gebruikersinterface voor elementen die binnen de default namespace van de bronontologie bestaan, gelijk aan de huidige. Slechts elementen die buiten de default namespace vallen, krijgen dan een expliciete namespace (prefix).

# Analyse van de huidige implementatie

Om een idee te krijgen van hoe de namespaces, en prefixes daarvan, binnen de huidige implementatie het best bepaald kunnen worden is een analyse gemaakt van de huidige implementatie. De huidige implementatie is gebouwd op het volgende model:



Figuur 1: Huidige implementatie

Hierbij geldt het volgende:

* De plug-in is gebouwd rond de packages ontology, fresnel en view (en wat hulp-packages die hier niet benoemd worden).
* Klassen uit de package ontology raadplegen de Protégé OWLAPI interface.
* De gele klassen leveren de labels aan de gebruikersinterface. Deze worden afgeleid uit de gedefinieerde naam van de betreffende Lens of Property (met optioneel het fresnel:label van de PropertyBinding).
* De groene klassen leveren de URI’s welke de namespaces bevatten.
* De rode klasse levert de eventueel aanwezige prefixes.

Uit dit overzicht valt bovendien het volgende af te leiden:

* Klassen in de package ‘ontology’ representeren overeenkomstige klassen uit de OWL API interface van de actieve ontologie in Protégé.
* Klassen in de package ‘fresnel’ definiëren het model van de Fresnel lenzen zoals ze gebruikt worden binnen de plug-in. Elke lens is hierbij optioneel gekoppeld aan een klasse uit de ontology package en elke Property is optioneel gekoppeld aan een generieke Property uit de ontology package. Een PropertyBinding legt vervolgens de relatie tussen een Lens en een bijbehorende Property vast.
* Klassen in de package ‘view’ worden gebruikt om gegevens voor de GUI vast te leggen, waaronder de labels welke te zien zijn op het scherm. Elke Lensbox heeft een koppeling met een Lens uit de fresnel package en elk PropertyLabel heeft een koppeling met een PropertyBinding uit de fresnel package.
* Er zijn twee kapstok klassen:
  + In fresnel.Fresnel worden alle lenzen geregistreerd en een koppeling gelegd met de (representatie van) de bronontology ontology.Ontology.
  + In ontology.Ontology worden de (representaties van) de klassen uit de bronontologie geregistreerd en bovendien een koppeling gelegd met de OWL API interface naar de bronontologie zelf, zoals die vanuit Protégé aangeboden wordt.
* Ieder data object (ontology.Class, ontology.Property, fresnel.Property & fresnel.Lens) heeft naast een URI ook een naam.
* De naam van een object uit de fresnel klasse is bepalend voor het betreffende label op de GUI.
* De koppeling van Property en Lens klassen in de fresnel package naar klassen in de ontology package zijn optioneel. Dit is waarschijnlijk gedaan om zelf (op de GUI) Lenzen en Properties te kunnen creëren waarvoor geen klasse of property in de bronontologie bestaat.

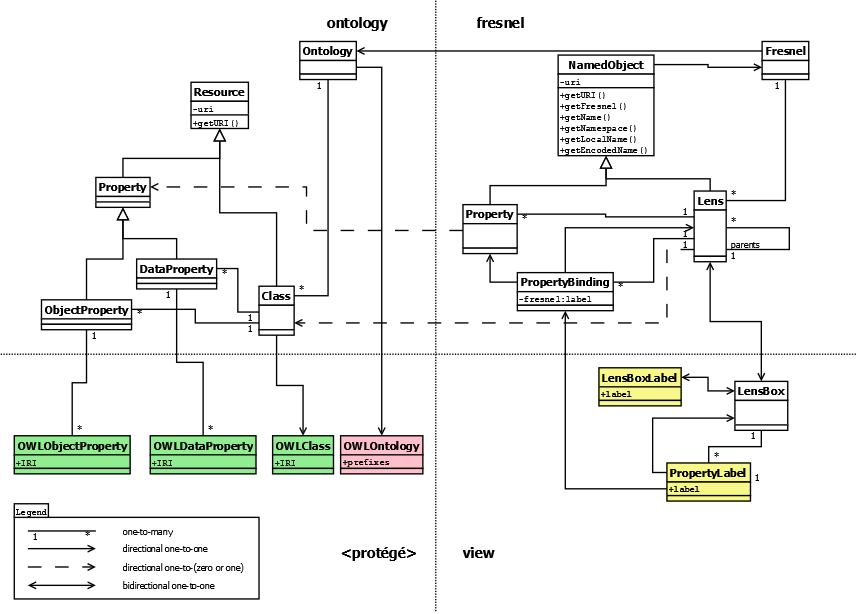
## Problemen met de huidige implementatie

De grootste problemen met de huidige implementatie zijn de volgende:

* De namen van data-objecten worden initieel bepaald als zijnde de fragment identifier van een URI. Hierbij is een object op basis van zijn naam niet per definitie meer uniek te benaderen; Voor verschillende namespaces kan immers dezelfde fragment identifier bestaan.
* Vanuit de klassen in de view package kunnen op dit moment de prefixes, welke aanwezig zijn in de OWLOntology interface, niet gebruikt worden om een verkorte weergave van een URI weer te geven. Voor de bijbehorende Lenzen en Properties is namelijk niet bekend op welke ontologie ze betrekking hebben; Vanuit klassen in zowel de fresnel als de ontology klasse is de kapstokinstantie op dit moment niet te herleiden.
* Doordat de namen van de verschillende objecten gebruikt worden om de betreffende Lens- , Group- en Format-instanties in de fresnel doelontologie te bepalen is op dit moment niet gegarandeerd dat de benoemde elementen erin altijd een unieke Lens, Property, Format of Group binnen de namespace van de doelontologie beschrijven.

# Oplossing

De oplossing voor de problemen, zoals hierboven beschreven, bestaat uit een implementatie van het volgende gewijzigde ontwerp:



Figuur 2: Aangepast ontwerp

Hierbij zijn de volgende wijzigingen aangebracht:

* De statische namen van alle objecten zijn verwijderd.
* De URI’s van objecten zijn eenduidig bepaald.
  + Klassen in de package ontology zijn nu afgeleid van één basisklasse Resource zoals gedefinieerd aan de hand van ‘Figure 4.1 Subclass relationships between OWL and RDF/RDFS’ uit (Antoniou & van Harmelen, 2008).
  + Klassen in de package fresnel zijn nu afgeleid van de klasse NamedObject.
* Waar namen nodig zijn (binnen de fresnel package), worden ze in de klasse NamedObject uniek bepaald uit de URI van een object waarbij waar mogelijk gebruik gemaakt wordt van de beschikbare prefixes om de naam kort en bondig te houden.
* Om prefixes uit de bronontologie te kunnen gebruiken voor een verkorte weergave, heeft elke NamedObject instantie nu kennis van het Fresnel model waarin het bepaald is. Dit Fresnel model weet op zijn beurt op welke bronontologie het van toepassing is.
* Voor het definiëren van unieke elementen in de fresnel doelontologie is een methode getEncodedName in klasse NamedObject opgenomen die per Lens, Group en Format een unieke naam binnen de namespace van de doelontologie moet bepalen.

## Het bepalen van namen

De namen van objecten binnen de plug-in worden nu als volgt bepaald op basis van de URI van het betreffende object:

1. Als voor een URI geen prefix gevonden kan worden in de bronontologie, dan wordt de volledige URI gebruikt als naam.
2. Als de namespace van een URI gelijk is aan die van de default namespace van de bronontologie, dan wordt de fragment identifier gebruikt als naam (deze is immers uniek binnen de default namespace.
3. Als voor een URI een geprefixte versie bepaald kan worden (buiten de default namespace), dan wordt deze gebruikt als naam.

### Het bepalen van namen voor Lenses, Groups en Formats in de doelontologie

Omdat de elementen in de fresnel doelontologie zich allen binnen de default namespace van de doelontologie bevinden, is het voor het bepalen van de namen hiervan van groot belang om ook hier te zorgen dat deze voor elk element uniek is. Om dit goed te kunnen doen, dienen ook deze namen, welke fragment identifiers worden in de default namespace van de doelontologie, gebaseerd te zijn op de URI’s van de objecten uit de bronontologie waarop ze betrekking hebben.

Omdat in fragment identifiers van URI’s echter geen :-tekens mogen voorkomen, worden deze namen als volgt samengesteld:

1. Waar mogelijk wordt voor objecten waarop de Lens, Group of Format van toepassing is, de geprefixte versie van de URI ervan genomen. Dit is gedaan om, net als in de GUI, de namen waar mogelijk kort en bondig te houden.
2. In de bepaalde naam worden alle liggende streepjes (-) vervangen door twee liggende streepjes (--). Dit is gedaan om na de wijziging in de volgende stap toch altijd een unieke naam als resultaat te hebben.
3. In de naam met eventueel een even aantal liggende streepjes, wordt elk : element vervangen door een liggend streepje. Aangezien het :-teken een gereserveerd URI teken is kan het niet twee maal achter elkaar voorkomen, het wordt slechts als scheidingteken gebruikt na het scheme-part of als scheidingsteken na een prefix. (The Internet Engineering Task Force, 2015)

# Bibliografie

Antoniou, G., & van Harmelen, F. (2008). *A Semantic Web Primer.* Massachusetts Institute of Technology.

The Internet Engineering Task Force. (2015, 01 17). *RFC 3986 - Uniform Resource Identifier (URI): Generic Syntax*. Opgehaald van https://tools.ietf.org/html/rfc3986