# DBpedia in Fresnel Forms plugin.

Open Universiteit   
Faculteit Informatica

14-5-2015

versie 1.0

J. N. Theunissen

838573218

Inhoud

[DBpedia in Fresnel Forms plugin. 1](#_Toc419376467)

[Inleiding 3](#_Toc419376468)

[Analyse 3](#_Toc419376469)

[Situatie 3](#_Toc419376470)

[Knelpunten 4](#_Toc419376471)

[Oplossing 4](#_Toc419376472)

[User interface aanpassingen 4](#_Toc419376473)

[Conclusie 5](#_Toc419376474)

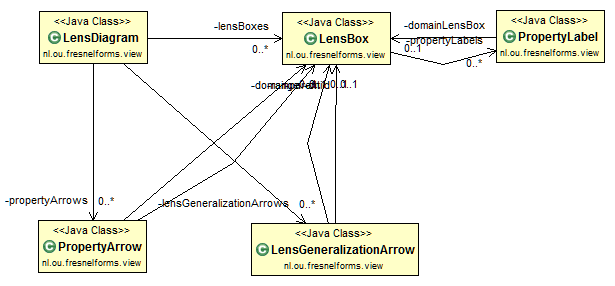
# Inleiding

Om de vraag van de opdrachtgever te kunnen beantwoorden of DBpedia ingelezen kon worden in de Fresnel Forms plug-in hebben we dit getest. Het bleek dat de DBpedia ontologie wel ingelezen kon worden maar ook dat de user interface onwerkbaar traag werd. In dit document wordt de analyse en de oplossing voor dit probleem beschreven. Nadat de oplossing de user interface weer werkbaar maakte zijn nog enkele kleine aanpassingen doorgevoerd die het werken met zeer grote ontologieën vergemakkelijken.

# Analyse

## Situatie

Om de oorzaak van de traagheid van de user interface te analyseren is de flow van de draw events van de verschillende klassen geanalyseerd. In fig. 1 staan de klasse die hierbij een rol speelde.



Figuur 1: Klassendiagram Fresnel Forms user interface.

We zien dat de klasse LensDiagram de klasse LensBox, PropertyArrow en LensGeneralizationArrow beheert. De beide klassen PropertyArrow en LensGeneralizationArrow zijn gerelateerd aan de twee lensboxes waar tussen de pijl getrokken moet gaan worden. De Lensbox beheert de property labels van de lens.

Een aanroep van het draw event van de klasse LensDiagram tekent het gehele Fresnel diagram bestaande uit de Lensen, de propertylabels, de pijlen naar de lenzen voor de object properties van de actieve lens en de hiërarchie pijlen tussen de lenzen.

## Knelpunten

Uit de analyse van de draw events zijn de volgende knelpunten naar voren gekomen.

1. De LensDiagram.draw methode initialiseert de PropertyArrow en LensGeneralisationArrow structuur elke keer als er een draw event plaats vind.
2. De lijnen worden bij elk draw event opnieuw berekend terwijl dit alleen maar hoeft als er een lensbox of propertlabel verplaatst is.
3. De propertylabels worden bij elk draw even van de Lensbox opnieuw gesorteerd door de Lensbox methode getPropertyLabels().

# Oplossing

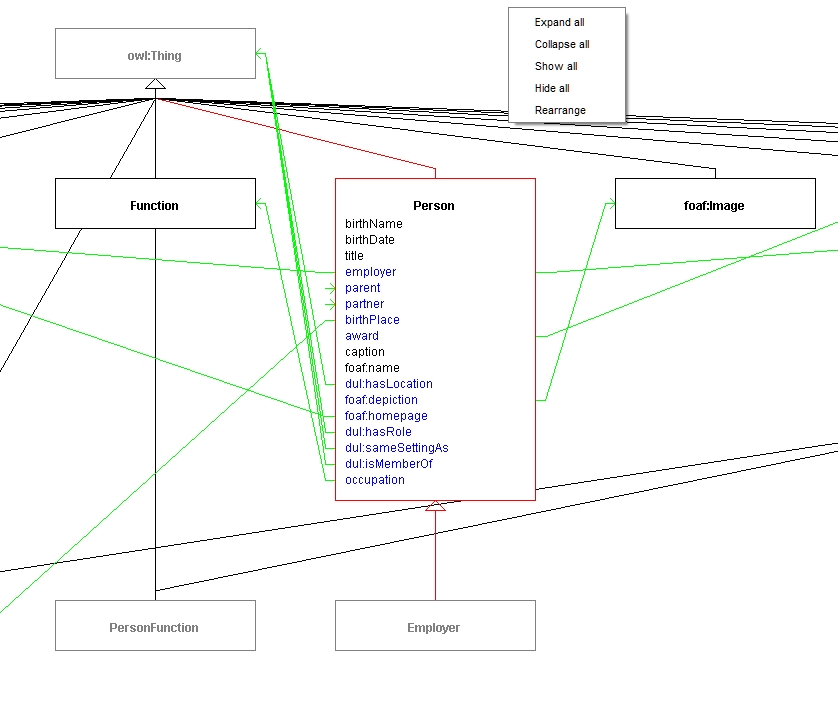
Voor bovengenoemde knelpunten zijn de volgende oplossingen mogelijk.

1. De PropertyArrow en LensGeneralisationArrow structuur worden één keer opgezet tijdens de initialisatiefase die aangegeven wordt door de boolean object variabele ‘init’ in het draw() event van de klasse LensDiagram.
2. De lensboxes worden voorzien van de boolean object variabele ‘lensdirty’. Deze variabele wordt geset als een lens verplaatst wordt of als de property labels opnieuw gesorteerd worden. De PropertyArrow en LensGeneralisationArrow klassen zijn voorzien van een refresh() methode die controleert of één van de twee lensboxes dirty is, alleen indien dat het geval is wordt de pijl opnieuw berekend. Het draw event van de LensDiagram reset de object variabele ‘lensdirty’ van de getekende lens.
3. De methode getPropertyLabels() retourneerd de propertylabel listarray zonder deze te sorteren. Er zijn 2 menuitems toegevoegd aan het rechtermuismenu van de lensbox waarmee de gebruiker de properties kan sorteren. De listarray wordt dan opnieuw opgebouwd met propertylabel objecten in de juiste volgorde.

# User interface aanpassingen

Het tekenen van alle propertylabels en de lijnen tussen de properties en de lensen vormen de grootste belasting. Daarom is besloten om alle lenzen gesloten te tekenen dat wil zeggen zonder de propertylabels. Een gebruiker kan met het rechtermuis menu een lens openen en sluiten.

Het algemene rechtermuis menu biedt de optie om alle lenzen te openen of te sluiten. Voor een uitgebreidere instructie van de user interface-wijzigingen zie de bijlage voor het document ‘User interface instructie.docx’ (Theunissen, 2015).



Figuur : Vernieuwde Fresnel Forms user interface.

# Conclusie

Bijna de gehele objecten structuur voor het LensDiagram werd bij elk draw event opnieuw opgebouwd waardoor het enorm traag werd bij grotere ontologieën. Door deze opbouw te beperken tot de situaties waarbij er iets veranderd is wordt de uitvoering van het draw event enorm versneld.   
De DBpedia ontologie kan nu in de FrenselForms plugin worden ingelezen.

Het blijkt dat een objectstructuur opbouwen een zware actie is terwijl het uitlezen van de object-structuur heel snel gaat.

# References

Theunissen, J. (2015). *Fresnel Forms User interface.*