GF Ontology Grammar Plugin for Eclipse

GF Ontology Grammar Plugin for Eclipse

This project is developed in Ontotext as a part of the Molto Project's WP2.
It is an Eclipse plugin that is part of the Grammatical Framework plugin for Eclipse(GFEP) but can be installed and used on its own.
The GF Ontology Grammar Plugin provides a GUI for calls to the Molto Repository Helper projects' code.
It is under the form of a wizard that leads a user to create abstract, concrete and SPARQL **gf grammar**, based on user-friendly templates that are filled in with repository entities by the user.
The wizard connects to a SPARQL endpoint and uses the ontology provided by it.
Then the user has to provide a specific templates file.
Next, pages for filling the templates are available.
Finally, the user is prompted to export the grammars created from the templates.
The exported grammars can then be edited by the GF Eclipse plugin.

GF Ontology Grammar Plugin - Developers notes

The code of the plugin is publicly available at Github:
The GF Plugin clone - Molto project
GF Ontology Grammar Plugin

Git notes

To see how to install `git`, check out this page.
To "checkout" the project as read-only, you just need to perform

```
git clone https://github.com/mmateva/gf-eclipse-plugin/
```

If you would like to contribute to the project at Github, create a Github account and follow the instructions here .
A basic Git explanation is introduced on this page. Generally, all you need is on the Github help pages.

Importing and using the project in Eclipse

We will refer to the project's home(Git's checkout directory) as `$GFEP_HOME`.
Basically, what you need to do is create an Eclipse workspace(can be the default one), and in the Project Explorer perform

```
Import >> General >> Existing Project into Workspace,
```

then via the file browser select the folder

```
$GFEP_HOME/gf-eclipse-plugin/workspace
```

The result is that you get(currently) six projects that you can import in your Eclipse workspace. The one that is described in this manual is `org.grammaticalframework.eclipse.ontology-grammar-wizard`.
You can optionally import the rest as well, they are part of the more general GF Eclipse plugin.

After you have imported the plugin project, you can directly run it in Eclipse.
Just select the `plugin.xml` file in the Project Explorer, right-click it, and `Run As >> Eclipse Application`.

Eclipse will run another Eclipse within itself, with its own classpath and Running Configuration.
The VM parameters are defined through `Run As >> Run Configuration`, all the rest is described via the plugin's `plugin.xml`, `build.properties`, `Manifest.MF`; details on these.
The "child" Eclipse instance has the GF Ontology Grammar Plugin installed.
You can start the plugin from it.
See below how to Run the plugin in Eclipse.

Eclipse plugins

Eclipse plugins are very well organized standalone modules that can be added to the Eclipse IDE and serve for various purposes.
Generally, they are any java code that is bound to the Eclipse IDE through extension points.
The two most important descriptor files are the `plugin.xml` and the `Manifest.MF`. Eclipse PDT provides a handy editor for them.
More about writing Eclipse plugins can be found at Eclipse Plugins Developer's guide and on this tutorial.

GF Ontology Grammar Plugin - User's guide

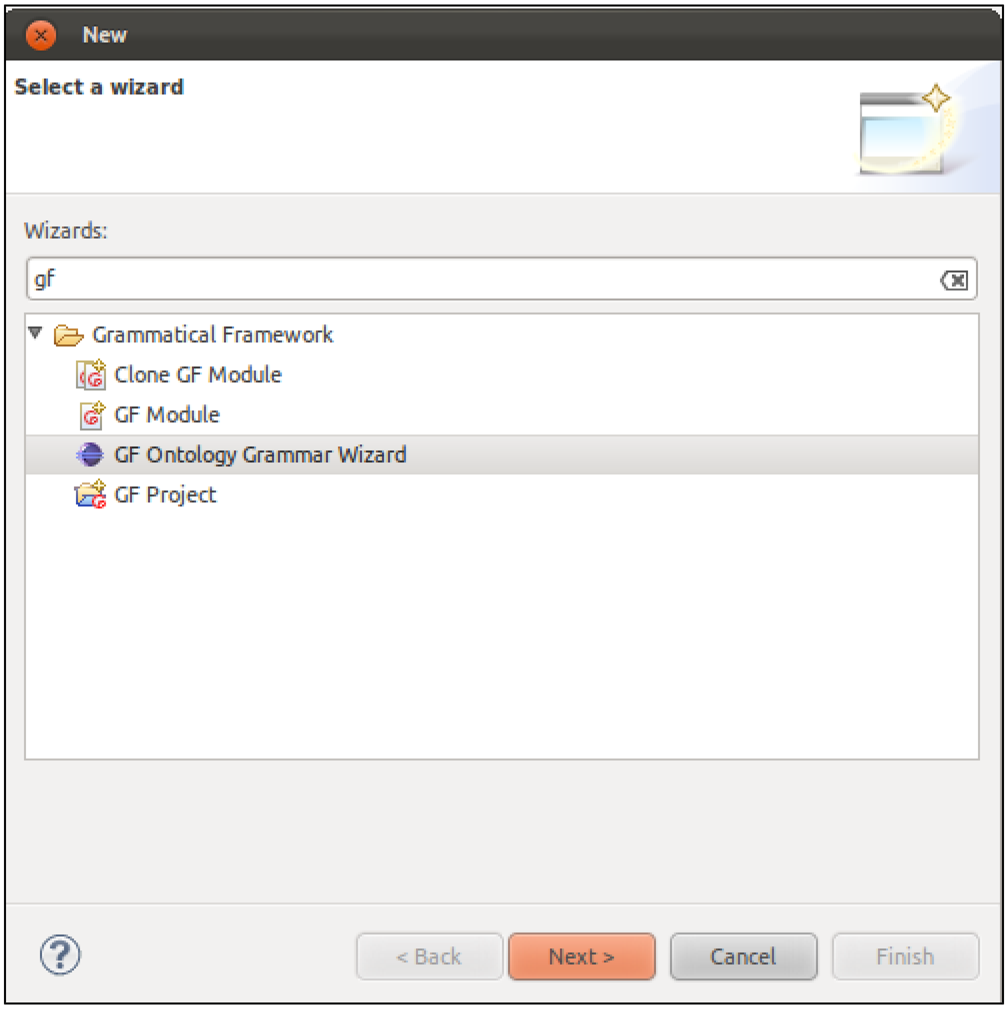
Import the plugin

⚠ Currently the plugin is not available for download from the Eclipse Plugin management pages. Hopefully, It will be soon 🙂

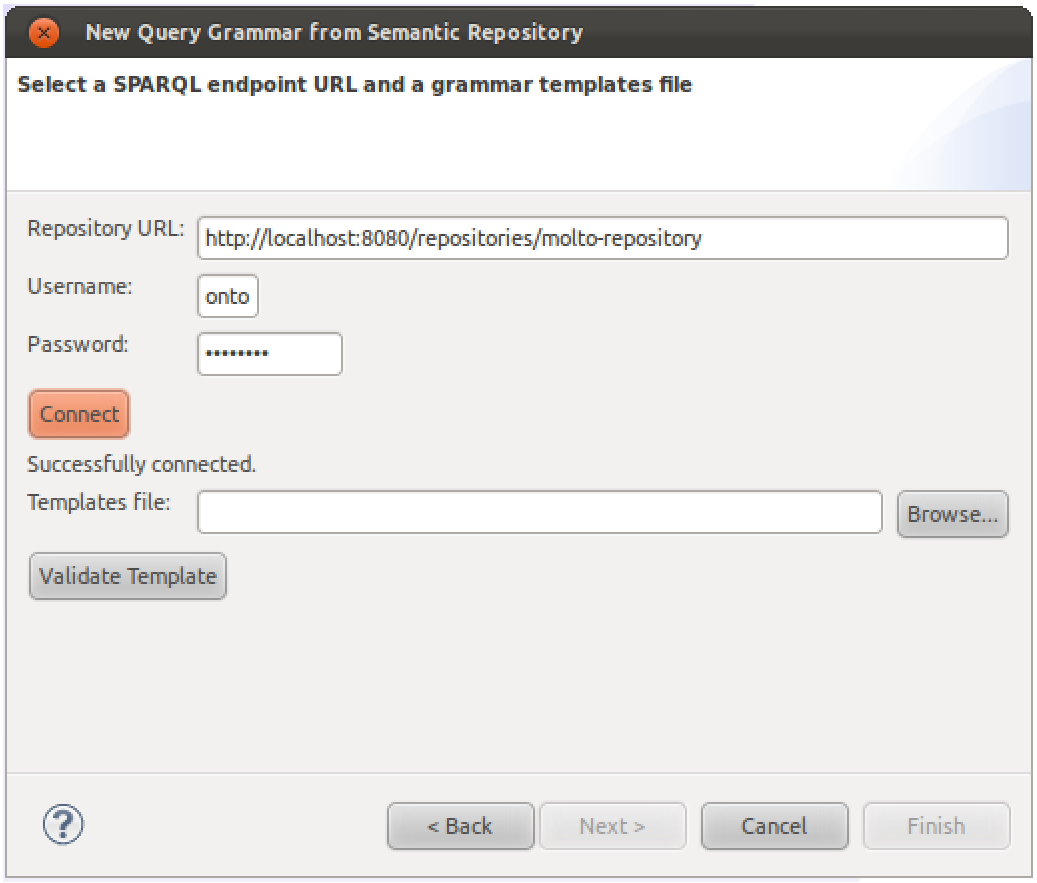
Run the plugin in Eclipse

⚠ Eclipse 3.7 or above is recommended

1. Navigate to File >> New >> Other... (or use Ctrl+N) from the Eclipse GUI.
Write "gf" in the search box, or scroll down to the "Grammatical Framework" category.
Select "GF Ontology Grammar Wizard".



2. Provide a SPARQL endpoint to be connected to(ontology repository).
Note: do not forget to explicitly add the port number in the address.
After the username and password are populated, use the "Connect" button to verify the connection is successful.



New Query Grammar from Semantic Repository

Select a SPARQL endpoint URL and a grammar templates file

Repository URL: http://localhost:8080/repositories/molto-repository

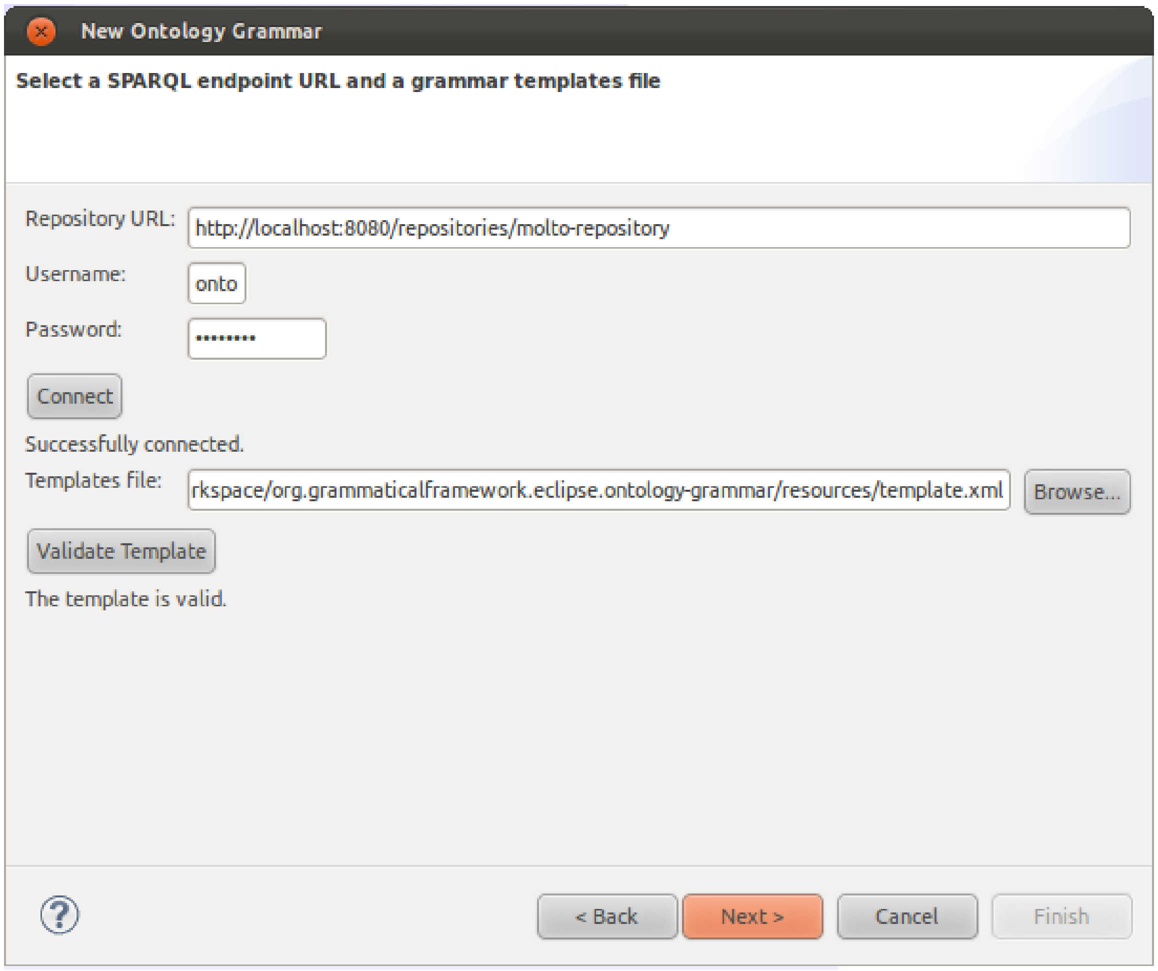
Username: onto

Password: •••••••••

Successfully connected.

Templates file:

3. Provide a templates file.
Select a templates from the file system.
An example for such a file is provided in the *resources* folder of the plugin.



Select a SPARQL endpoint URL and a grammar templates file

Repository URL: http://localhost:8080/repositories/molto-repository

Username: onto

Password: •••••••••

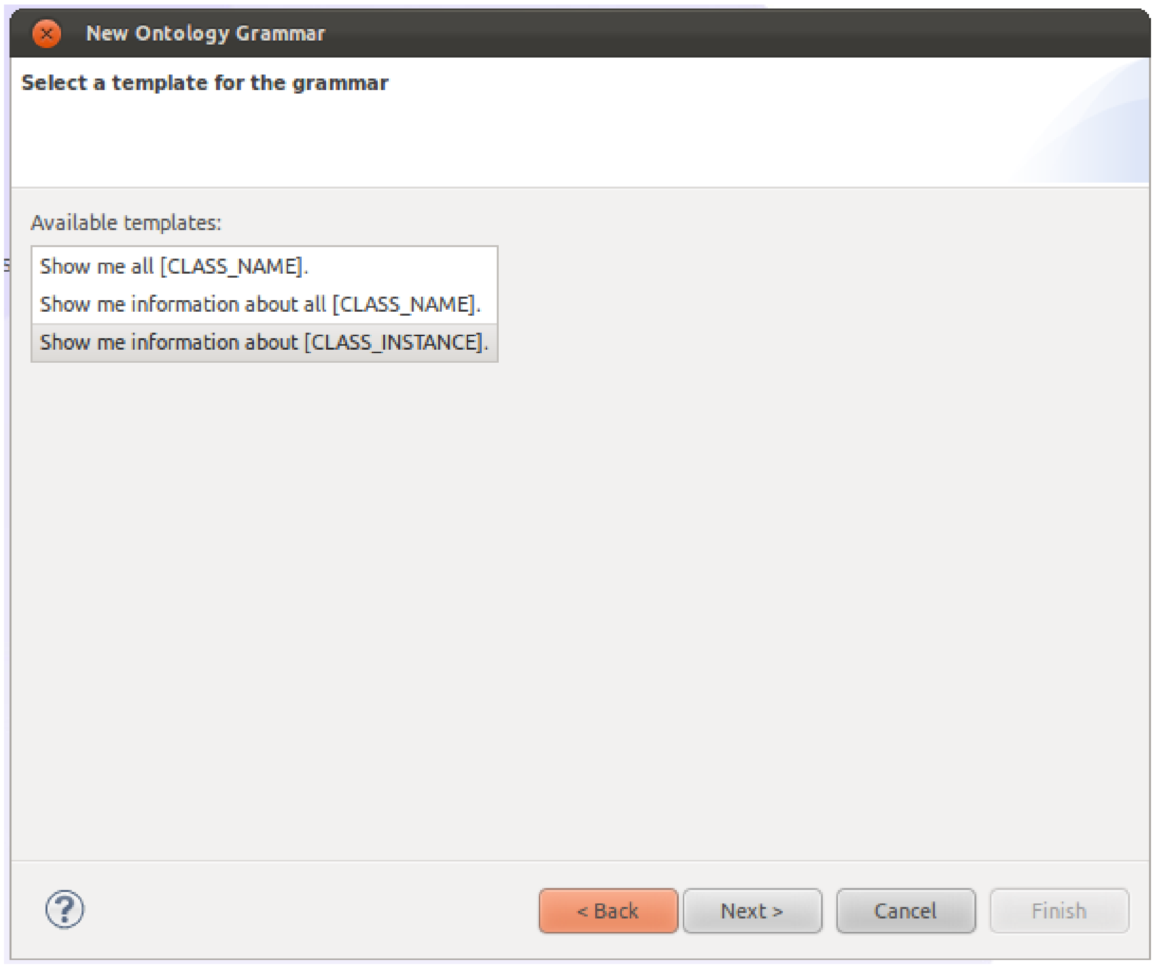
Successfully connected.

Templates file: rkspace/org.grammaticalframework.eclipse.ontology-grammar/resources/template.xml

The template is valid.

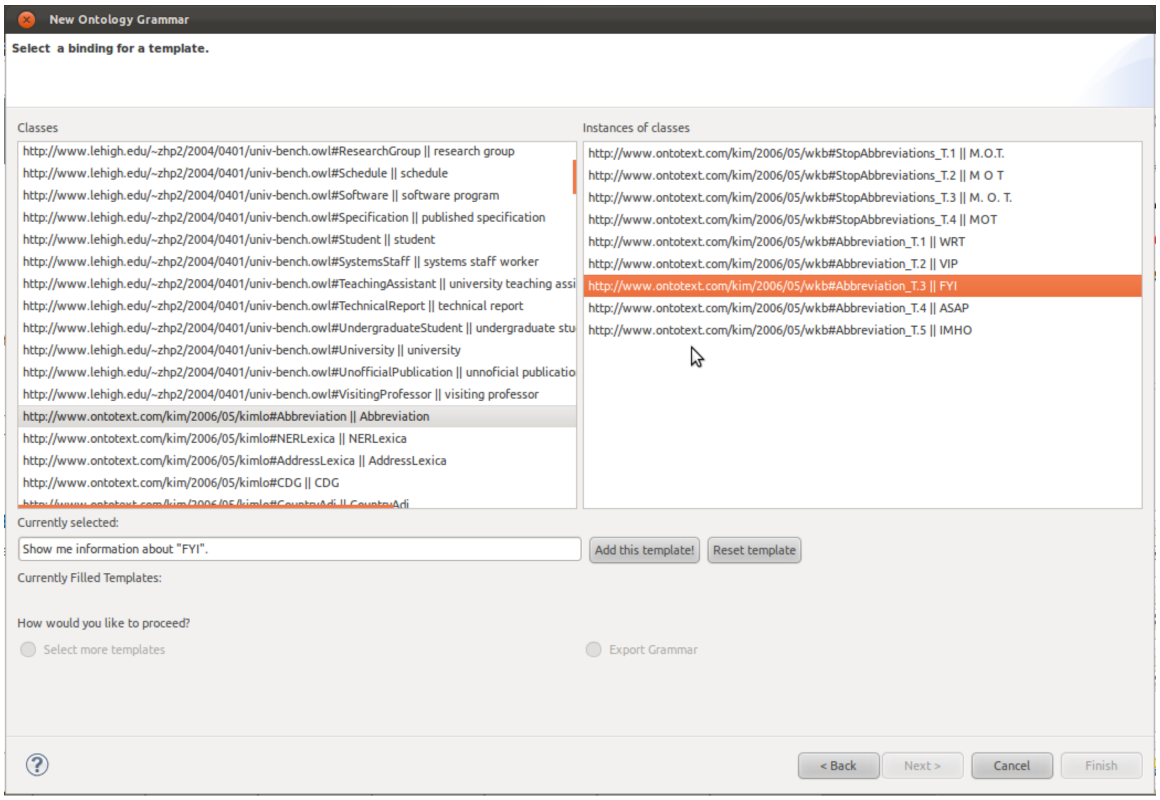
It is automatically validated, but you can still use the "Validate template" button to check if it is valid.
Then you can proceed to the next page via the "Next" button.

4. Select a template from the list.
Select a single template from the list - as on the screenshot.
The binding in square brackets, e.g. *[CLASS_INSTANCE], [CLASS_NAME]*, will be populated on the next page.



Show me all [CLASS_NAME].
Show me information about all [CLASS_NAME].
Show me information about [CLASS_INSTANCE].

5. Fill the templates.
On the page for entities selection, currently one has "Classes" and "Instances".
The instances on the left are the corresponding ones to the classes on the right(they are the instances of the selected class).
If a class has no instances, then an *[EMPTY LIST]* note is shown.
The template is automatically populated with a matching binding when one is selected(see the screenshot below).


New Ontology Grammar

Select a binding for a template.

Classes
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#ResearchGroup || research group
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Schedule || schedule
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Software || software program
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Specification || published specification
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Student || student
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#SystemsStaff || systems staff worker
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#TeachingAssistant || university teaching assi
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#TechnicalReport || technical report
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#UndergraduateStudent || undergraduate stu
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#University || university
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#UnofficialPublication || unofficial publicatio
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#VisitingProfessor || visiting professor
- http://www.ontotext.com/kim/2006/05/kimlo#Abbreviation || Abbreviation
- http://www.ontotext.com/kim/2006/05/kimlo#NERLexica || NERLexica
- http://www.ontotext.com/kim/2006/05/kimlo#AddressLexica || AddressLexica
- http://www.ontotext.com/kim/2006/05/kimlo#CDG || CDG
- http://www.ontotext.com/kim/2006/05/kimlo#CountryAdj || CountryAdj

Instances of classes
- http://www.ontotext.com/kim/2006/05/wkb#StopAbbreviations_T.1 || M.O.T.
- http://www.ontotext.com/kim/2006/05/wkb#StopAbbreviations_T.2 || M O T
- http://www.ontotext.com/kim/2006/05/wkb#StopAbbreviations_T.3 || M. O. T.
- http://www.ontotext.com/kim/2006/05/wkb#StopAbbreviations_T.4 || MOT
- http://www.ontotext.com/kim/2006/05/wkb#Abbreviation_T.1 || WRT
- http://www.ontotext.com/kim/2006/05/wkb#Abbreviation_T.2 || VIP
- http://www.ontotext.com/kim/2006/05/wkb#Abbreviation_T.3 || FYI
- http://www.ontotext.com/kim/2006/05/wkb#Abbreviation_T.4 || ASAP
- http://www.ontotext.com/kim/2006/05/wkb#Abbreviation_T.5 || IMHO

Currently selected:

Show me information about "FYI".

Add this template! | Reset template

Currently Filled Templates:

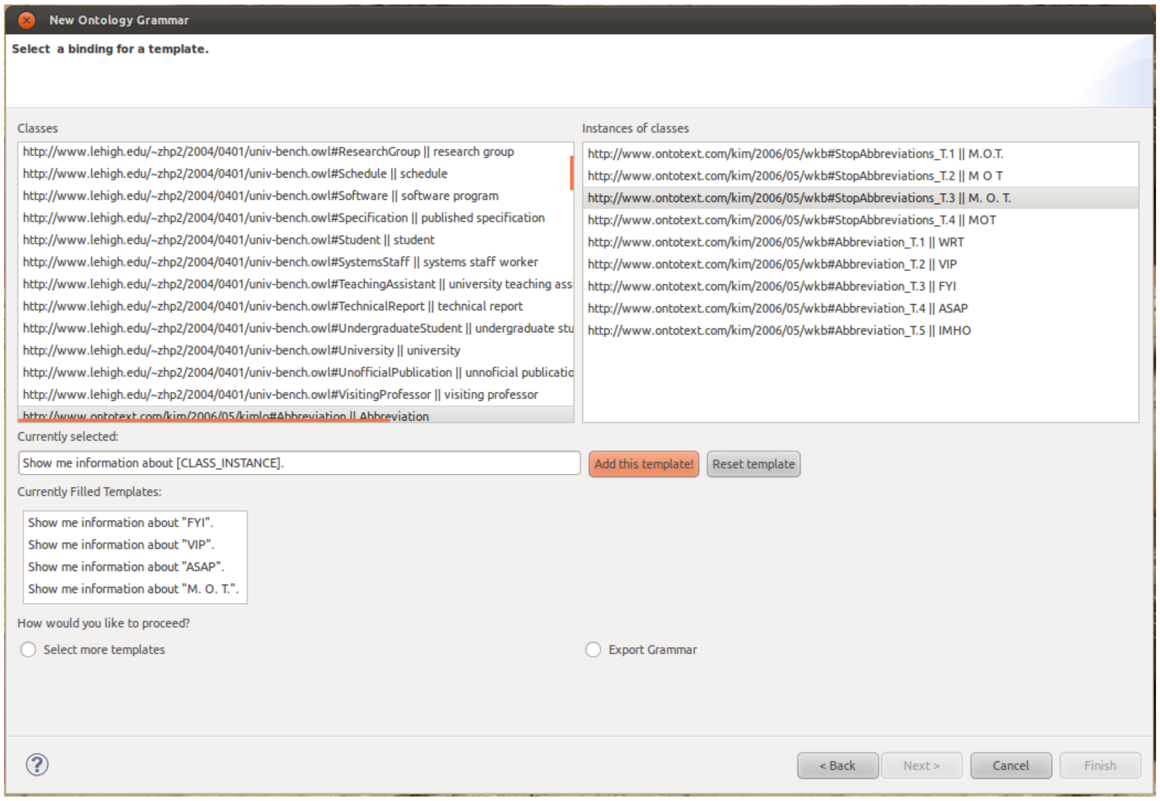
How would you like to proceed?

○ Select more templates ○ Export Grammar

< Back | Next > | Cancel | Finish

If you constantly get the "LOADING" sign in the Instances box, you might need to consider a better connection speed to your SPARQL endpoint.

6. Add templates
Use the "Add Template" button to add new templates.


New Ontology Grammar

Select a binding for a template.

Classes
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#ResearchGroup || research group
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Schedule || schedule
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Software || software program
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Specification || published specification
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Student || student
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#SystemsStaff || systems staff worker
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#TeachingAssistant || university teaching ass
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#TechnicalReport || technical report
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#UndergraduateStudent || undergraduate stu
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#University || university
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#UnofficialPublication || unofficial publicatic
- http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#VisitingProfessor || visiting professor
- http://www.ontotext.com/kim/2006/05/kimlo#Abbreviation || Abbreviation

Instances of classes
- http://www.ontotext.com/kim/2006/05/wkb#StopAbbreviations_T.1 || M.O.T.
- http://www.ontotext.com/kim/2006/05/wkb#StopAbbreviations_T.2 || M O T
- http://www.ontotext.com/kim/2006/05/wkb#StopAbbreviations_T.3 || M. O. T.
- http://www.ontotext.com/kim/2006/05/wkb#StopAbbreviations_T.4 || MOT
- http://www.ontotext.com/kim/2006/05/wkb#Abbreviation_T.1 || WRT
- http://www.ontotext.com/kim/2006/05/wkb#Abbreviation_T.2 || VIP
- http://www.ontotext.com/kim/2006/05/wkb#Abbreviation_T.3 || FYI
- http://www.ontotext.com/kim/2006/05/wkb#Abbreviation_T.4 || ASAP
- http://www.ontotext.com/kim/2006/05/wkb#Abbreviation_T.5 || IMHO

Currently selected:

Show me information about [CLASS_INSTANCE].

Add this template! | Reset template

Currently Filled Templates:

Show me information about "FYI".
Show me information about "VIP".
Show me information about "ASAP".
Show me information about "M. O. T.".

How would you like to proceed?

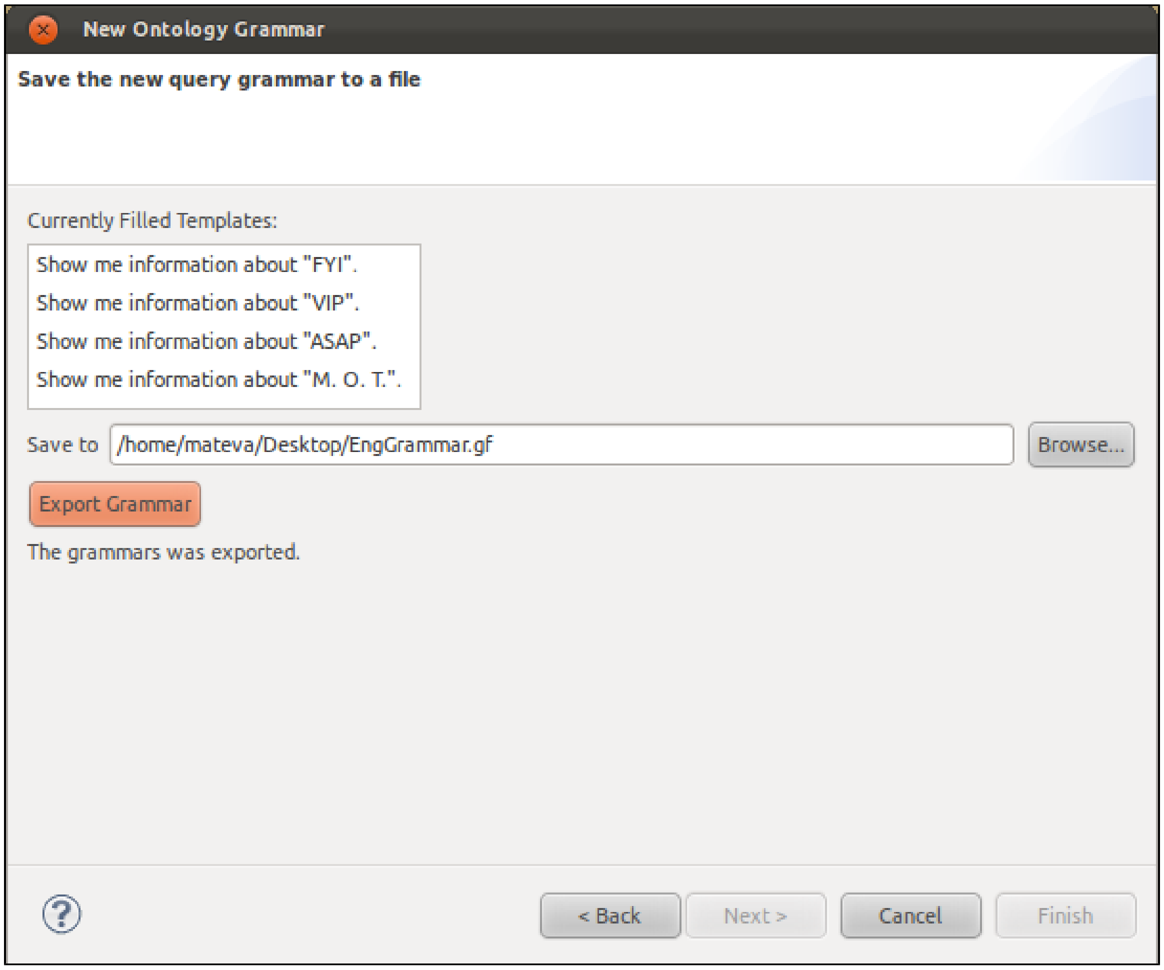
○ Select more templates ○ Export Grammar

< Back | Next > | Cancel | Finish

Also, you can "Reset" the template, to see the binding type that is populated.

When enough templates, you can either continue with adding new template patterns("Select more templates"), or save the grammar created to a file("Export Grammar").
The first option actually brings to 4.

7. Export Grammars
The grammar exportpage simply gives you the opportunity to choose where to store the ontology grammar files.
After selecting the destination, you need to press the "Export Grammar" button.


New Ontology Grammar

Save the new query grammar to a file

Currently Filled Templates:

Show me information about "FYI".
Show me information about "VIP".
Show me information about "ASAP".
Show me information about "M. O. T.".

Save to /home/mateva/Desktop/EngGrammar.gf

The grammars was exported.

This will create the following three grammar files in the destination folder:

Abstract<GrammarName>.gf - abstract ontology grammar that was created
Concrete<GrammarName>.gf - concrete ontology grammar that was created
<GrammarName>SPARQL.gf - SPARQL query grammar that was created

Molto Repository Helper project

The Molto Repository Helper is developed in Ontotext, also as part of the Molto project.
It provides an API for creation of abstract, concrete and SPARQL GF grammars.
They are based on user-friendly templates that are filled in with repository entities.