

Using OwlExporter: various questions

Submitted by [massic80](#) [1] on Tue, 2010-09-07 14:48

- [Semantic Software Tools and Resources](#) [2]

Hi anyone!

I just finished reading OWLExporter documentation, I still have to begin trying it and I have a small doubt: in "Exporting Object Property Relationships across Domain and NLP Ontologies" chapter I see "We began by modifying the Domain ontology to import the NLP ontology". Should it also be done on the NLP Ontology for the inverse relation?

Another thing: "continuing with the Document example Jape grammar created previously, for the title and source information to be exported as datatype properties in the ontology we created the datatype properties title and source that have Document as the domain and xsd:string as the range"

Shall all Datatype properties be declared previously? I'd need to add some found "randomly", so having them pre-declared would be impossible to me.

If I create an annotation for OWLExporter containing some instances of a not-declared datatype, there would be a problem or the reasoner will simply infer the existence of that property reading the axiom?

I also have some minor correction and improvements to suggest for documentation, but I think it's better to write them directly to your email address... Waiting for an acknowledgment :)

Thanks again

Massi

[<OrganismTagger in GATE Embedded](#) [3] [Mulitpax not working with Supple Parser >](#) [4]

Ontology Population vs. Learning

Submitted by witte on Sat, 2010-09-11 11:55.

Hi Massy,

Regarding the question on adding new properties, I think you might be confused on the purpose of Ontology Learning vs. Ontology Population. OwlExporter is an *ontology population* tool, that is, it adds instances (detected in texts through GATE) to an already existing ontology. If you want to discover new ontology classes and relations from text, you need an *ontology learning* tool, like Text2Onto, which is also based on GATE.

Best, René

»

where i can found owl exporter

Submitted by Alternan on Mon, 2012-11-26 17:11.

Please, if you can tell me where to download the plugin owl exporter.

In several places I see how the people talk about this plugin, but cant download from the site.

Gate 7 target to <http://creole.semanticsoftware.info/OwlExporter> [5], but the site tell me that I don't have permission to download

the plugin.

I don't found in any other place that plugin.

Thanks in advance. My email is in the profile, but just in case you can write me here: acarrasco@udio.cujae.edu.cu [6]

»



[7]

OwlExporter web page

Submitted by rene on Mon, 2012-11-26 17:15.

You'll find it on the OwlExporter web page: <http://www.semanticsoftware.info/owlexporter> [8]

Please follow the installation instructions as explained under <http://www.semanticsoftware.info/owlexporter#Installation> [9]

Best, René

»

Hi René! Thanks for

Submitted by massic80 on Mon, 2010-09-13 03:11.

Hi René! Thanks for answering: I didn't really want to make ontology learning (and Text2Onto lacks lots of documentation :))... I thought that some could declare an axiom like "dog eats meat" and the reasoner would have inferred that the connection exists... like the inverse when I declare an Object Property with a Domain and Range, and if my axiom contains a different domain it infers and adds the new one.

»

[Hello Massi, @ Should it also](#) [10]

Submitted by ninus on Wed, 2010-09-08 15:06.

Hello Massi,

@ Should it also be done on the NLP Ontology for the inverse relation

When importing one ontology into another using "owl:import" it is bi-directional, meaning constructs (concepts, properties etc..) can span across both ontologies.

@ Shall all Datatype properties be declared previously

The owl exporter creates an "id" datatype property that corresponds to the id created by GATE for a given entity in the corpus.

All other properties (datatype or object) need to already exist in the ontology for relations to be created. And it is case sensitive.

@I'd need to add some found "randomly"

This is not possible without adding additional implementation to the owl exporter's source code which is made available under GNU licensing.

@If I create an annotation for OwlExporter containing some instances of a not-declared datatype

The triple will not be exported and there would be no inferencing on the expression.

@I also have some minor correction and improvements to suggest for documentation

We welcome any improvements to the documentation. You can reach me at ni kham encs.concordia.ca

»

Four more :)

Submitted by massic80 on Thu, 2010-09-09 04:33.

Hi again!

@ "Should": U gave me a nice confirmation, thanks :) I didn't remember that particular.

@ "Shall": so, if I create an axiom which doesn't include a known pre-declared in the ontology, there will be a problem - like a Java exception? My particular case is: I find out documents' "metadata", like authors, from a table in the first page. This table isn't always identical, so it may contain, for example, "Author" or "Authors", and I use this for the properties' labels.

@ "If I create": ah you already answered my question; "will not be exported" is better than "will throw an exception" :)

@ "I also have": they will come! :) I'm a draft-corrector-born :D

I have more more questions: for the instanceName feature of OwlExportClassXXX annotations, you say that it's "the normalized form of the annotated text", and in the example code you don't make any normalization to exported strings... shall we use a String method to normalize it? What should be done to those strings? I figure out it's due to OWL limitations, but I don't remember them ;)

Can the import and export ontologies in runtime parameters be the same? This may also be tied to another parameter: MultiOwlExport. What's the difference in saying true or false?

In page 9 and 10 of OwlExporter guide you say that a sourceUrl datatype is used by OwlExporter to determine if a document has been already elaborated or not. In the code the feature is called sourceUrl, but in the description list, after "title" we can find just "source", and the same is in the generic NLP ontology you released for the tries. Since what you said before, creating a sourceUrl datatype property with OwlExporter for an ontology containing just "source" wouldn't add that axiom, right?

If I'm not mistaken, OwlExporter "ties" two different instances with an object relation by using the annotations IDs; what would it do in the not-frequent-but-not-impossible case where two exported annotations had the same ID?

Thanks for the n-th time!

Massi

»

[Hello Massi, Thank you for \[11\]](#)

Submitted by Anonymous on Wed, 2010-09-15 18:46.

Hello Massi,

Thank you for your patience,

@ for the instanceName feature of OWLExportClassXXX annotations

For the instanceName feature of a given annotation, I use GATE's doc.getContent(start,end) method that returns a string representation of the annotation given a span. The instanceName feature, gets exported as a datatype property instanceName(OWLExportClassXXX, xsd:string) for the created instance in the ontology.

Since the range is string there is no ASCII character limitation to what can be stored in that datatype property. Please remember that the "instanceName" datatype property needs to be created in the ontology for the expression to be exported.

@ Can the import and export ontologies in runtime parameters be the same

They cannot, the OwlExporter recreates the import ontologies from scratch and saves it to the export ontology.

@ In page 9 and 10 of OWLExporter guide you say

This is a mistake, the object property is sourceURL. Thank you for that Massi!

@ If I'm not mistaken, OWLExporter "ties"

You still need to create the OwlExporterRelationDomain or OwlExporteRelationNLP temp annotations, for relationships between entities to be exported. You could also export a reflexive relationship for example Ninus hasName Ninus, where Ninus has the same GATE Annotation ID. If you run into problems doing this and I can help you.

Thanks again for your patience Massi!

Regards,

Ninus.

»

[Hi Ninus! Back in the time \[12\]](#)

Submitted by Anonymous on Tue, 2010-11-02 11:50.

Hi Ninus!

Back in the time you wrote to me (on the forum) how to modify the build.properties file to make it correctly build and it worked. Later, it still asked me to find classes which it couldn't find in Windows classpath, and I solved by re-zipping 'em in OWLExporterV2.jar. You also suggested to modify some more files (I figure out instead of re-zipping) to solve this, but I didn't (look and) find out which ones, I need you to explain them better.

Now, the problem isn't exactly the classpath one:

OwlExporter, with a 10.2 MB ZIP containing all the classes, seems like starting, with

```
corefChainList = TempNPSW Chain (self-inserted, I'm not using them)
debugFlag= true
exportDomainOntology =
file:/C:/blahblahblah/OntologieOutput/SoftwareEngineering.owl
exportNLP = true (my first try just works on the NLP ontology)
exportNLPontology = file:/C:/blahblahblah/OntologieOutput/NLP.owl
importDomainOntology =
file:/C:/blahblahblah/htdocs/SoftwareEngineering.owl (in Apache
directory, I'll explain why, anyway another directory than the export
Ontology, because you said they must be different)
importNLPontology = file:/C:/blahblahblah/htdocs/NLP.owl
inputASName = (empty)
multiOwlExport = true
```

but says

```
WARNING: [Local Folder Repository] The specified file must be a
directory. (C:\Programmi\Gate\plugins\edu.stanford.smi.protegex.owl) --
LocalFolderRepository.update()
WARNING: Could not get ontology from URL: http://localhost/NLP.owl --
HTTPRepository.update()
```

My ontologies have a <http://localhost/> [13] namespace, because I tried to "aid" OwlExporter/Protege in finding the NLP ontology when imported in the domain ontology (Protege uses that repository-method...), as indicated in your documentation. In order to allow them to find the NLP ontology I installed Apache, and put the ontologies in its base folder, to be "findable" by their whole URI (which should be the first location in which Protege looks for the imports).

The same happens with latest version of OwlExporter, which you released in september and just needs a file to be modified with the right path and builds successfully (thank you). As the previous version, I needed to zip the classes in the jar, and it began to "try working".

Maybe it's still a path problem? Why doesn't it find the imported ontology? Is there a runtime parameters problem? I also tried with or without an existing output ontology: nothing changed.

I built the ontologies with Protégé 3.4.4: I could also show them to you, if you need that.

In three cases, it gave this message:

```
WARNING: [Local Folder Repository] The specified file must be a
directory. (C:\Programmi\Gate\plugins\edu.stanford.smi.protegex.owl) --
LocalFolderRepository.update()
Exception in thread "ApplicationViewer1" java.lang.NoClassDefFoundError:
org/protege/editor/owl/model/hierarchy/roots/Relation
at
edu.stanford.smi.protegex.owl.jena.parser.TripleProcessor.(TripleProcessor.java:61)
```

```
at
edu.stanford.smi.protege.owl.jena.parser.GlobalParserCache.getTripleProcessor(GlobalParserCache.java:74)

at
edu.stanford.smi.protege.owl.jena.parser.ProtegeOWLParser.loadTriples(ProtegeOWLParser.java:290)

at
edu.stanford.smi.protege.owl.jena.parser.ProtegeOWLParser.run(ProtegeOWLParser.java:213)

at
edu.stanford.smi.protege.owl.jena.JenaKnowledgeBaseFactory.loadKnowledgeBase(JenaKnowledgeBaseFactory.java:227)

at edu.stanford.smi.protege.model.Project.loadDomainKB(Unknown Source)
at
edu.stanford.smi.protege.model.Project.createDomainKnowledgeBase(Unknown
Source)
at
edu.stanford.smi.protege.owl.jena.creator.OwlProjectFromUriCreator.create(OwlProjectFromUriCreator.java:85)

at
edu.stanford.smi.protege.owl.ProtegeOWL.createJenaOWLModelFromURI(ProtegeOWL.java:107)

at
info.semanticsoftware.owl exporter.OwlExporterOntology.createOntology(OwlExporterOntology.java:158)

at
info.semanticsoftware.owl exporter.OwlExporter.execute(OwlExporter.java:247)
at gate.util.Benchmark.executeWithBenchmarking(Benchmark.java:299)
at gate.creole.SerialController.runComponent(SerialController.java:221)
at gate.creole.SerialController.executeImpl(SerialController.java:153)
at
gate.creole.SerialAnalyserController.executeImpl(SerialAnalyserController.java:115)
at gate.creole.AbstractController.execute(AbstractController.java:62)
at gate.util.Benchmark.executeWithBenchmarking(Benchmark.java:299)
at
gate.gui.SerialControllerEditor$RunAction$1.run(SerialControllerEditor.java:1526)
at java.lang.Thread.run(Thread.java:662)
Caused by: java.lang.ClassNotFoundException:
org.protege.editor.owl.model.hierarchy.roots.Relation
at java.net.URLClassLoader$1.run(URLClassLoader.java:202)
at java.security.AccessController.doPrivileged(Native Method)
at java.net.URLClassLoader.findClass(URLClassLoader.java:190)
at java.lang.ClassLoader.loadClass(ClassLoader.java:307)
at gate.util.GateClassLoader.loadClass(GateClassLoader.java:63)
at java.lang.ClassLoader.loadClass(ClassLoader.java:248)
... 19 more
```

I saw this after having "upgraded" OWLExporter and inserted the URI instead of dir path, so <http://localhost/SoftwareEngineering.owl> [14] and <http://localhost/NLP.owl> [15] for the input ontologies. It happened both for the latest and butlast version of OWLEXporter (re-switching the folders and restarting GATE), then, after having re-modified the input ontologies to a "file:" URI and back again to http it gives the same problem described first. It gave the second problem also trying with your demo ontologies and gazetteers and default ANNIE. I can't find that

not-found class in Protégé 3.4.4 (you say you used 3.4.2), but only in 4.x, is that normal?

I figure out the first is an ontology importing problem, but how did you solve it? I know that Protégé asks for where to find the files (if not available at the indicated URL), but this doesn't work with the folders nor with the URLs as namespace. How could I solve? Merging the two ontologies?

I can also let you use my computer with Teamviewer or any remote desktop
U'd prefer to make the tries :)

Thanks in advance for you (hopefully fast) answer! I know that this mail is a complete chaos, but pardon me, it's 3 a.m. and I fought OWLExporter all day! :)

Massi

P.S.: if you want I may re-post this in the forum, but I still don't have plenty of time :)

P.P.S.: I used part of the classes found in Protegé package, part from Protégé-OWL... I could find them all in just one of them. Did I make it right?

P.P.P.S.: what are those nlp.dot and nlp.dot-input files in your demo ZIP?

»

Hi again, Ninus, and thank

Submitted by massic80 on Thu, 2010-09-16 07:12.

Hi again, Ninus, and thank you for your answers, even if I'm still waiting for some details-answers to my previous post :)

For the rest:

@the instanceName feature, gets exported as a datatype property instanceName(OWLExportClassXXX, xsd:string) for the created instance in the ontology.

I thought that the instanceName was the name/URI-fragment-identifier of the ontology individual... if it becomes a datatype property, where does the name come from then?

@ the OwlExporter recreates the import ontologies from scratch and saves it to the export ontology

Explain to me: the import is copied in the new one (comprehending instances) and the new axioms are added?

And what's the difference between creating multiple OWLs and not?

@ This is a mistake, the object property is sourceURL. Thank you

U're welcome... modify the documentation PDF! ;)

@ You still need to create the OwlExporterRelationDomain or OwlExporteRelationNLP temp annotations

Yes, but what I mean is: when they are exported to the ontology, the IDs "vanish", keeping them tied by their URIs and avoiding multiple equal IDs?

»



Except where otherwise noted, all original content on this site is copyright by its author and licensed under a [Creative Commons Attribution-Share Alike 2.5 Canada License](https://creativecommons.org/licenses/by-sa/2.5/ca/).

Source URL (retrieved on 2025-12-21 19:52):

<https://www.semanticsoftware.info/forum/using-owl exporter-particular-documentation-example-importing-ontology>

Links:

- [1] <https://www.semanticsoftware.info/users/massic80>
- [2] <https://www.semanticsoftware.info/forums/tools-resources-forum/durm-corpus-wiki-tools>
- [3] <https://www.semanticsoftware.info/forum/organismtagger-gate-embedded>
- [4] <https://www.semanticsoftware.info/forum/mulitpax-not-working-supple-parser>
- [5] <http://creole.semanticsoftware.info/OwlExporter>
- [6] <mailto:acarrasco@udio.cujae.edu.cu>
- [7] <https://www.semanticsoftware.info/users/rene>
- [8] <http://www.semanticsoftware.info/owl exporter>
- [9] <http://www.semanticsoftware.info/owl exporter#Installation>
- [10] [#comment-33](#)
- [11] [#comment-61](#)
- [12] [#comment-75](#)
- [13] <http://localhost/>
- [14] <http://localhost/SoftwareEngineering.owl>
- [15] <http://localhost/NLP.owl>