Semantic Annotations for BPMN models: Extending SeMFIS for supporting ontology reasoning and query functionalities

Dimitraki Katerina

Thesis submitted in partial fulfillment of the requirements for the
Master of Science degree in Computer Science
University of Crete
School of Sciences and Engineering
Computer Science Department

Thesis Supervisor: Prof. Dimitris Plexousakis

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH)

Contents

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH)

1) An abstract of my thesis

Business process models are a sequential representation of all functions associated with a specific business activity. In implementing business process modelling, there are many techniques that have been tried and tested throughout the years. Business Process Modelling Notation (BPMN) is the global standard for process modeling and provides a standard notation that is readily understandable by management personnel, analysts and developers. The original intent of BPMN was to help bridge communication gaps that often exist between the various departments within an organization or enterprise, by providing a notation that is intuitive to business users, yet able to represent complex process semantics.
Here we will discuss the approach of enriching the existing standards, with semantic information provided by OWL ontologies. Business Process annotation with semantic tags taken from an ontology is becoming a crucial activity for business designers. In fact, semantic annotations help business process comprehension, documentation, analysis and evolution. For this purpose there is an ongoing research and several software tools have been developed to support this approach.
SeMFIS tool is an open and extensible platform for engineering semantic annotations of conceptual models. We decided to extend SeMFIS to support consistency check on OWL ontology models, created for annotate BPMN process models, with the use of HermiT reasoner and a feature to allow user uploading his ontology with its instances and properties on Openlink Virtuoso engine and query them with SPARQL language. Creating a meat supply chain process model and a meat supply chain OWL ontology that enriches the process model with information necessary for traceability purposes, helps us understand the need for annotations and for features which process them.

2) What I wanted to achieve with SeMFIS

My contribution’s goal was to find a tool that allows semantic annotations and add new functionality which will add extra value to that tool and will assist users taking advantage of theirs enriched BPMN models in a more complete way.

The first part of my contribution is the addition of an ontology consistency check feature. It allows the user to perform a consistency check on the obtained ontology information related to one or several BPMN models. Consistency checking operations check whether instances of data are consistent with a given schema. If the ontology is inconsistent, the tool prompts the user to choose among the possible

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH)

correlated BPMN and semantic annotation models and checks if a BPMN activity is annotated to the class type of that inconsistent instance and if yes opens the BPMN model that contains that activity and places an error mark around it.
For the second part of my contribution due to the need for more complex queries on my ontology models I created the feature “Semantically enriched queries”. The functionality of this feature is to allow the user upload his ontology and his desirable ontology instances, after processing them and convert them in OWL format, on Openlink Virtuoso engine and query them. There are some predefined queries available and also the user can write his own queries.

3) Extensions on SeMFIS

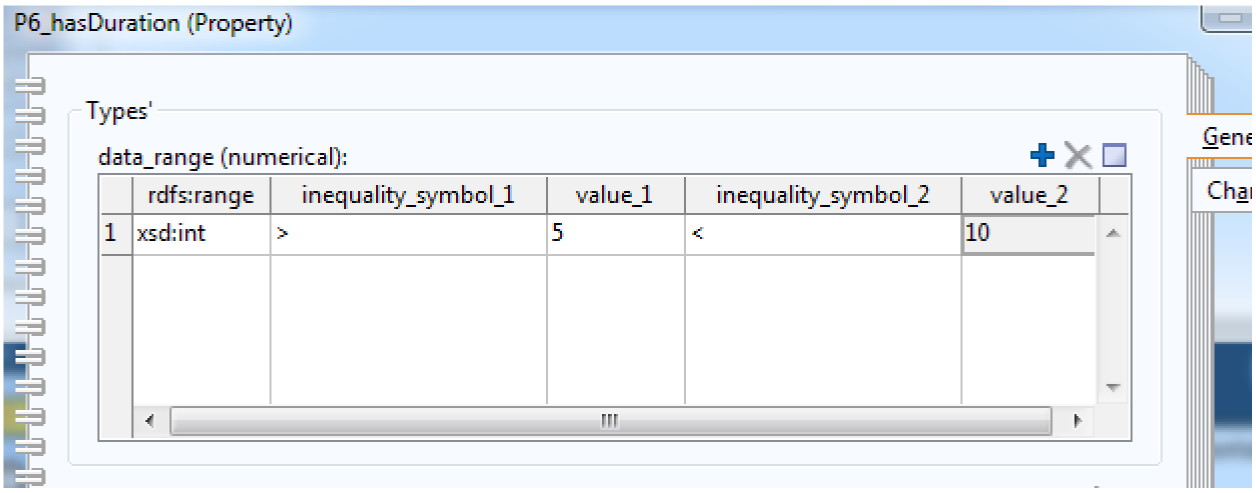
Extending SeMFIS modelling constructs for supporting concepts of OWL 2.0

I added the below extensions so as SeMFIS Ontology model could support the needs of my use case ontology “Meat Value Chain”. So, now SeMFIS can support schema constraints, value constraints and cardinality constraints.

3.1) Restrictions on Data Properties

A) Define data range restrictions on Datatype properties

I added a record that allows the user to define inequality and data range restrictions on Property class.


P6_hasDuration (Property)

Types'

data_range (numerical):

	rdfs:range	inequality_symbol_1	value_1	inequality_symbol_2	value_2
1	xsd:int	>	5	<	10

Figure 3.1: User can define the value of a datatype property

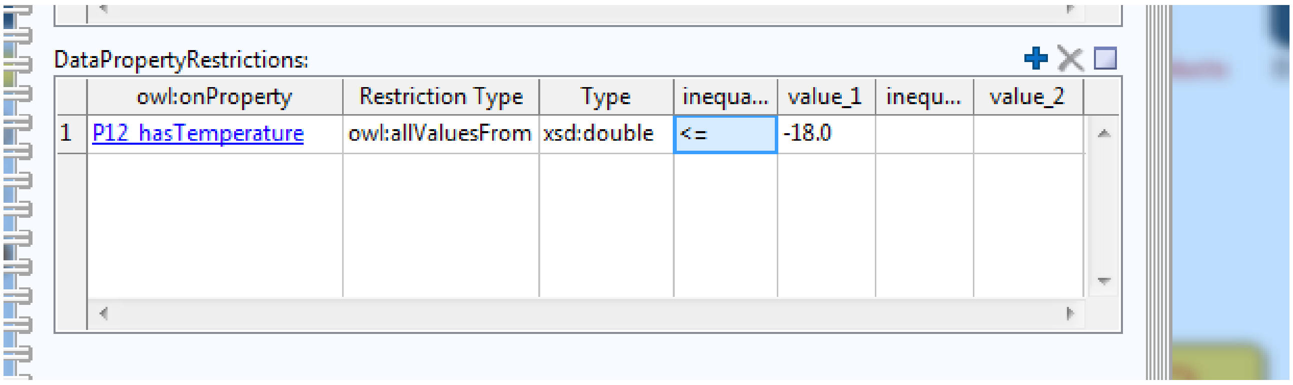
This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH)

Example:

P6_hasDuration
Domain: TransportationOfLivestock
Range: xsd: int [>5, <10] (hours)

B) Define data range restrictions on Datatype properties as subclass of a class

I added a record that allows the user to define inequality and data range property restrictions as subclasses of object Class.


DataPropertyRestrictions:

	owl:onProperty	Restriction Type	Type	inequa...	value_1	inequ...	value_2
1	P12_hasTemperature	owl:allValuesFrom	xsd:double	<=	-18.0		

Figure 3.2: User can set datatype property restrictions

Example:

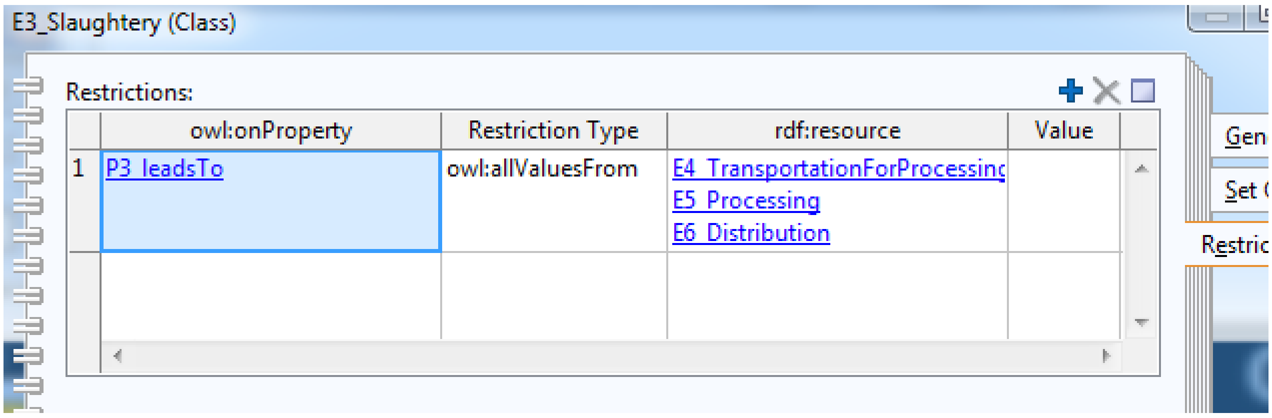
Class: E45_TransferFrozenMeat
<rdfs:subClassOf>
P12_hasTemperature owl: allValuesFrom xsd: double[<= -18] ^{0}C

This statement restricts the instances of type E45_TransferFrozenMeat when they are related to the property *P12_hasTemperature*, they are allowed to have values of type *double* and range <= -18 otherwise a consistency check will find our ontology instances inconsistent.

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH)

3.2) Sequence restrictions

*A) I extended the record "Restrictions" on notebook of object Class to be able to support concepts such as [**Class1** subclassOf (**Property1 Class2** (or **Class3**) ..)]*
This axiom says that if something is an instance of **Class1** and it's related to something by the property **Property1**, then that something is an instance of **Class2** (or an instance of **Class3**) etc.
This axiom is useful because I can restrict a process (represented with a class) to be followed by specific process (or processes).


E3_Slaughtery (Class)

Restrictions:

	owl:onProperty	Restriction Type	rdf:resource	Value
1	P3 leadsTo	owl:allValuesFrom	E4 TransportationForProcessinç E5 Processing E6 Distribution	

Figure 3.3: User can set datatype property restrictions

We can see that if an instance of class **E3_Slaughtery** is related to an instance by the property **P3_leadsTo**, that instance could belong only to the classes **E4_TransportationForProcessing**, **E5_Processing** or **E6_Distribution**.

4) Consistency check feature

The Consistency Check feature checks if an ontology is consistent. In case the reasoner returns that the ontology is inconsistent, the feature checks if there is any BPMN activity that is annotated to the class type of the detected inconsistent instance and then places an error mark around that activity. So, a non-expert user can conduct reasoning to an ontology that he knows that is annotated to BPMN processes of his interest and view the results in the BPMN model without get involved with ontology details.
As far as SeMFIS exports his models in an XML format we should transform the exported file(s) into RDF/XML syntax for processing them. The Java Architecture for

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH)

XML binding (JAXB) simplifies access to XML documents from a Java program. So, JAXB helped me to transform the OWL ontology concepts represented in XML to RDF/XML syntax. The transformation and processed results were then processed and displayed with the use of the scripting language of ADOxx AdoScript.

Describing my feature with more details:

1) We go to the Analysis interface and we see on the Menu bar the word "Reasoning", when we click on it, it is pointed out to click "Consistency check..".

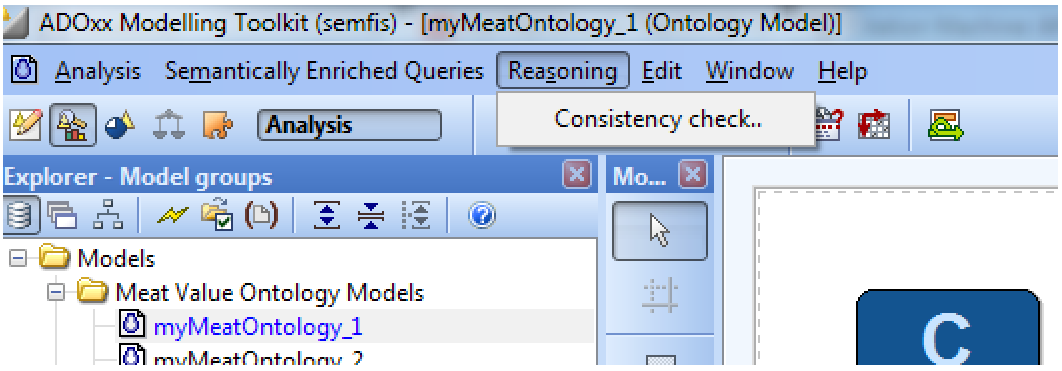
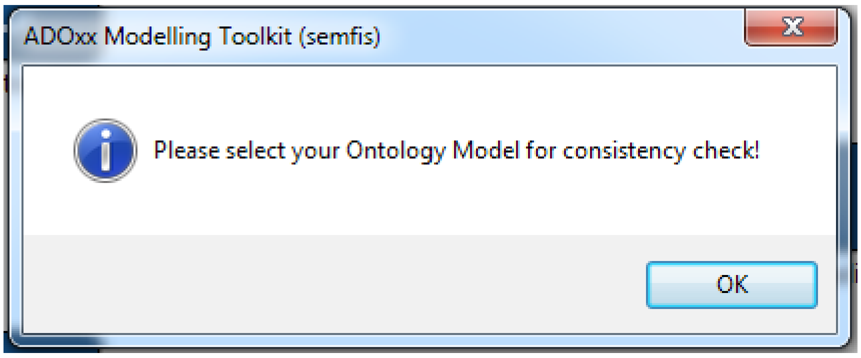

ADOxx Modelling Toolkit (semfis) - [myMeatOntology_1 (Ontology Model)]
Analysis
Semantically Enriched Queries
Reasoning
Edit
Window
Help
Consistency check..
Analysis
Explorer - Model groups
Mo...
Models
Meat Value Ontology Models
myMeatOntology_1
myMeatOntology_2
C

Figure 4.1: Reasoning feature on Menu

2) After clicking on "Consistency Check.." a message is displayed that calls the user to choose the preferred ontology for reasoning.

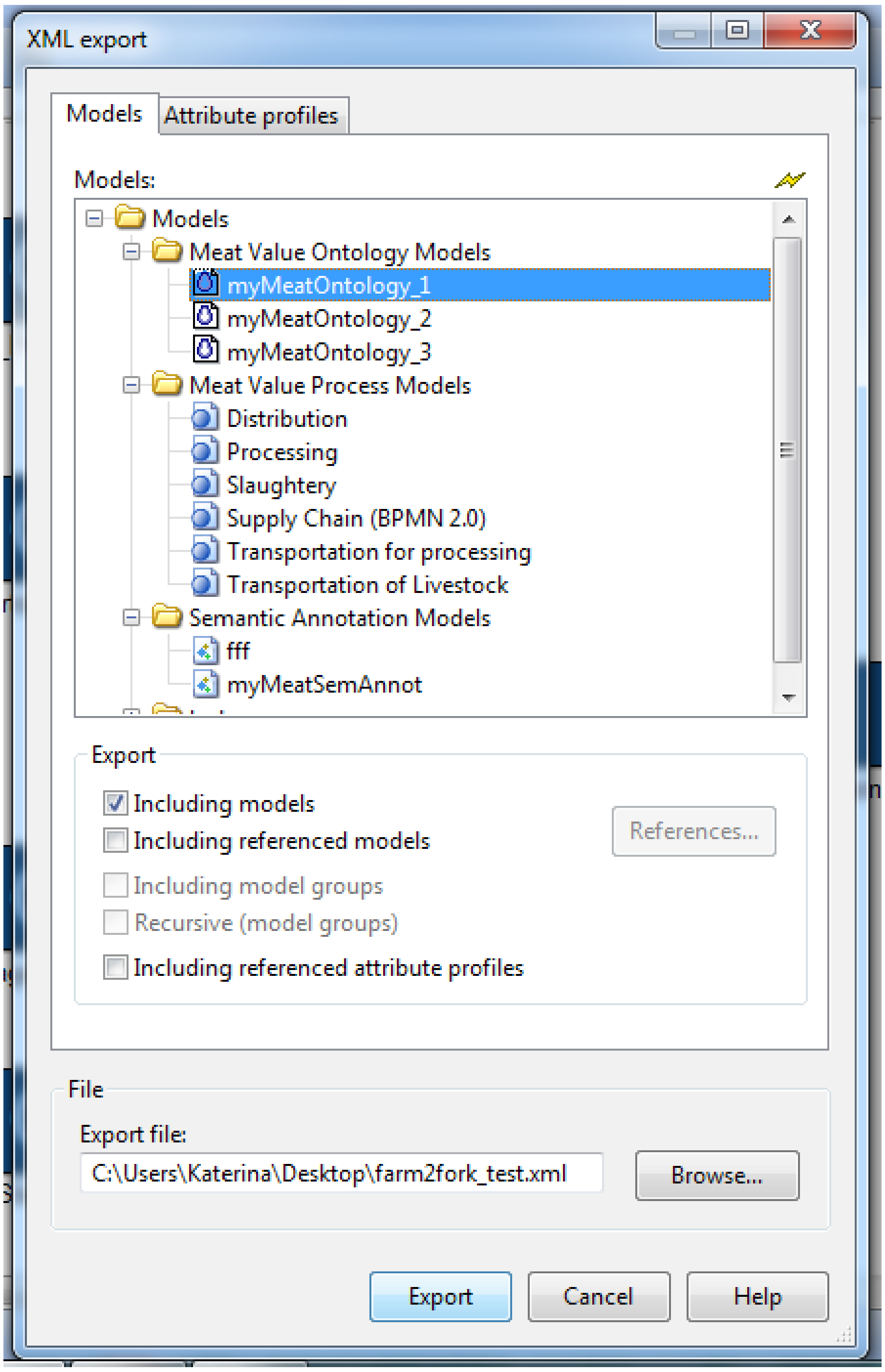


Please select your Ontology Model for consistency check!

Figure 4.2: Prompts user to choose his ontology models

3) The user chooses the ontology model for reasoning, then he chooses a file for exporting it and at last he clicks "Export. We suppose that the reasoning will be conducted over one ontology schema though ontology instances can be into multiple ontology models.

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH)


XML export

Models | Attribute profiles

Models:

- Models
  - Meat Value Ontology Models
    - myMeatOntology_1
    - myMeatOntology_2
    - myMeatOntology_3
  - Meat Value Process Models
    - Distribution
    - Processing
    - Slaughtery
    - Supply Chain (BPMN 2.0)
    - Transportation for processing
    - Transportation of Livestock
  - Semantic Annotation Models
    - fff
    - myMeatSemAnnot

Export

[✓] Including models
[ ] Including referenced models
[ ] Including model groups
[ ] Recursive (model groups)
[ ] Including referenced attribute profiles

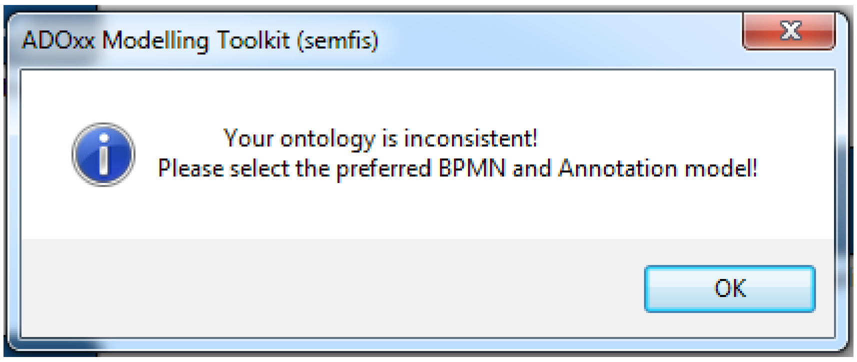
File

Export file: C:\Users\Katerina\Desktop\farm2fork_test.xml

Figure 4.3:The user chose the ontology model “myMeatOntology_1”

4) Then in case that the chosen ontology is inconsistent, we get the following message:

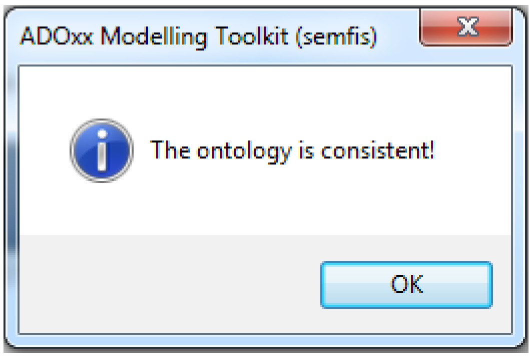
This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH)


ADOxx Modelling Toolkit (semfis)

Your ontology is inconsistent!
Please select the preferred BPMN and Annotation model!

Figure 4.4: Message in case of inconsistency

It prompts us to choose one or more BPMN and Semantic Annotation models that we know (or suppose) that we would find an annotation with our inconsistent ontology. The feature finds just one inconsistent instance for the sake of simplicity. If we want to find out more inconsistencies we have to correct the ones that has been found previously.
Otherwise we get the message that our ontology is consistent:

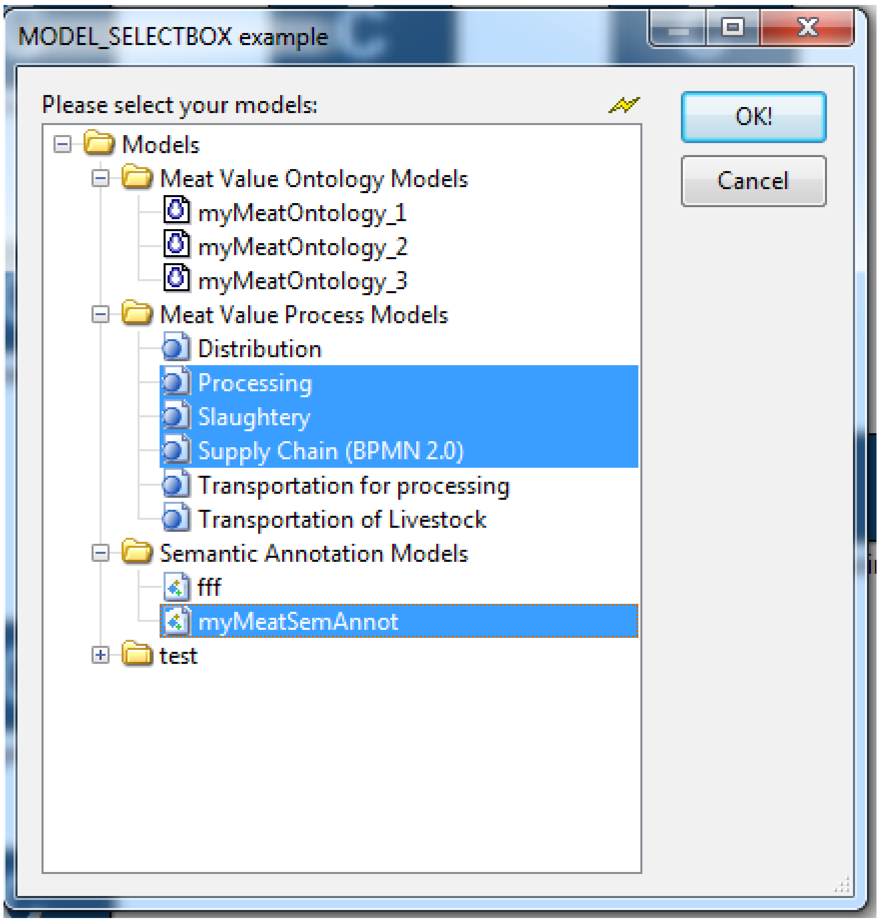

ADOxx Modelling Toolkit (semfis)

The ontology is consistent!

Figure 4.5: The ontology is consistent

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH)

5) The user chooses the BPMN and Semantic Annotation model


MODEL_SELECTBOX example

Please select your models:

- Models
  - Meat Value Ontology Models
    - myMeatOntology_1
    - myMeatOntology_2
    - myMeatOntology_3
  - Meat Value Process Models
    - Distribution
    - Processing
    - Slaughtery
    - Supply Chain (BPMN 2.0)
    - Transportation for processing
    - Transportation of Livestock
  - Semantic Annotation Models
    - fff
    - myMeatSemAnnot
  - test

Figure 4.6: User should choose the preferred BPMN and Semantic Annotation models

6) If there is a semantic annotated BPMN activity with our inconsistent class, the BPMN model that contains it, opens automatically and we can see an error mark around that activity.

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH)

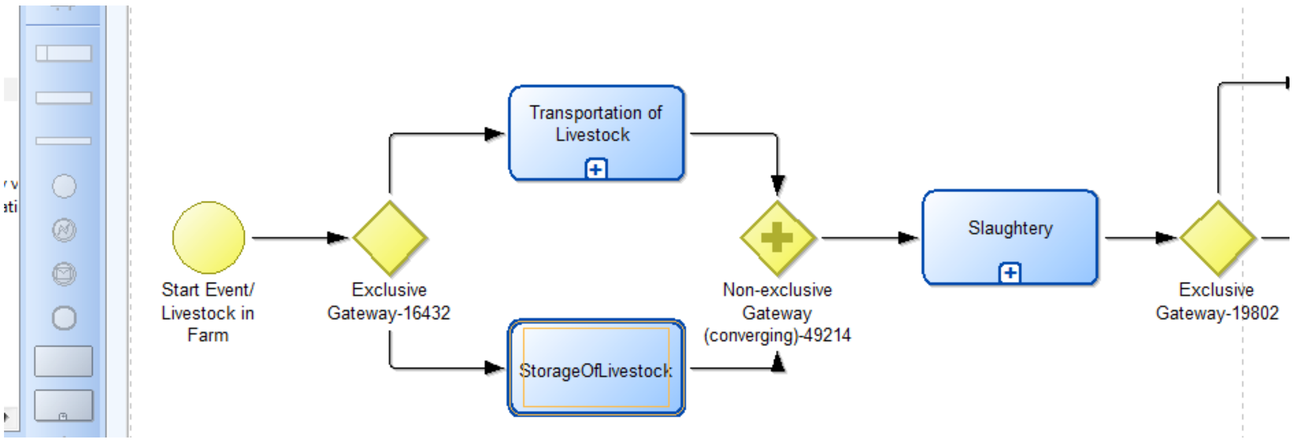
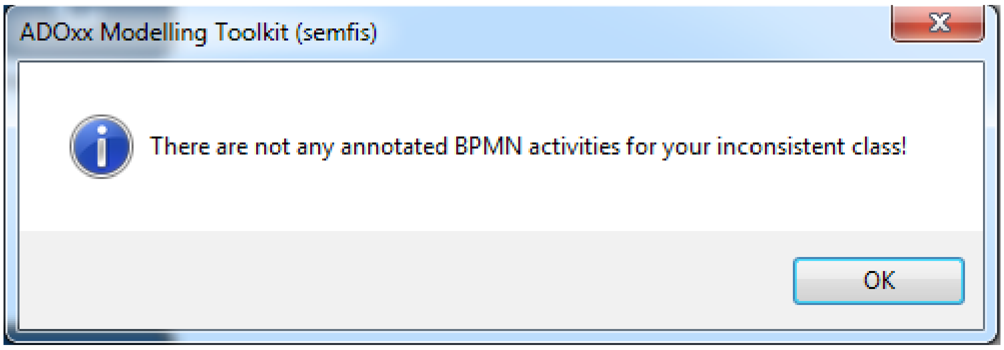

Transportation of Livestock
Start Event/ Livestock in Farm
Exclusive Gateway-16432
StorageOfLivestock
Non-exclusive Gateway (converging)-49214
Slaughtery
Exclusive Gateway-19802

Figure 4.7: Annotated BPMN Activity

If we cannot address an annotated activity we get the following message:


ADOxx Modelling Toolkit (semfis)

There are not any annotated BPMN activities for your inconsistent class!

Figure 4.8: The feature could not find any annotated activities

4.1) Supported owl concepts and limitations

Supported owl concepts

The transformation from XML syntax representation of SeMFIS to RDF/XML syntax can support the following concepts:

Class class

1) Inheritance
2) owl:intersectionOf
3) owl:unionOf
4) owl:complementOf
5) owl:disjointWith

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH)

6) Restrictions on Object and on Datatype properties

Property class

1) Distinguish a property as Object or DatatypeProperty
2) Define Domain and Range for both types of Property class
3) Numerical Datatype property restrictions
4) owl:inverseOf, owl:functional, owl:inverseFunctional, owl:symmetric, owl:transitive

Instance class

1) Type of instances
2) Datatype Properties
3) Object Properties

Limitations

- I do not support Namespaces class, I use one common namespace for the sake of simplicity

- I restricted my code to read the explanations after the consistency check fails and find out only datatype property inconsistencies, restrictions on object properties and functional property inconsistencies

- A consistency check finds one inconsistency per time, if somebody wants to find more, he can correct the one found and run another check

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH)

5) Semantically enriched queries

Due to the need for more complex queries on my ontology models, I created the feature “Semantically enriched queries”. With this feature, my purpose is to take advantage of the SPARQL query language. The functionality of this feature is to allow the user upload his ontology and his desirable ontology instances on Openlink Virtuoso engine and query them.

Describing my feature **“Upload to Virtuoso”** with more details:

1) We go to the Analysis interface and we see on the Menu bar the item “Semantically enriched queries”, when we click on it, it is pointed out to click “Upload to Virtuoso” or “Query Ontology Models”.

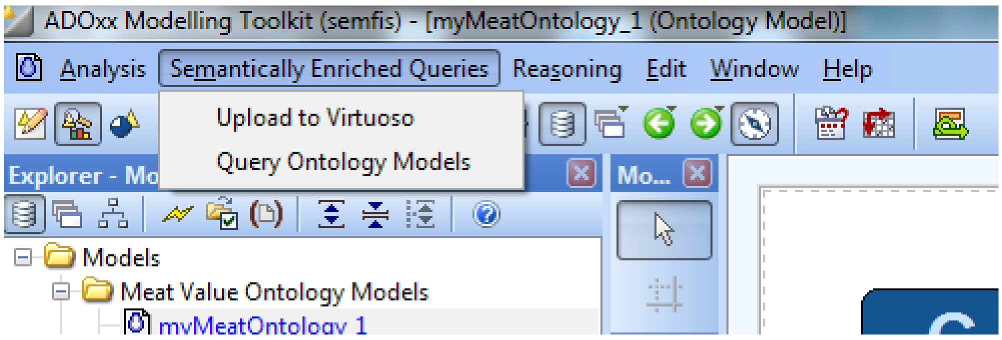
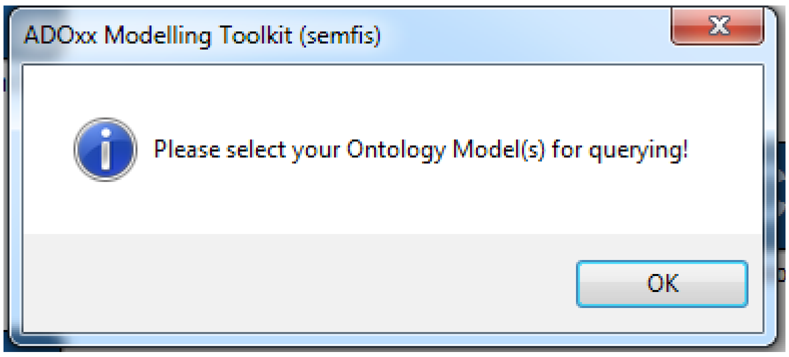

ADOxx Modelling Toolkit (semfis) - [myMeatOntology_1 (Ontology Model)]
Analysis
Semantically Enriched Queries
Reasoning
Edit
Window
Help
Upload to Virtuoso
Query Ontology Models
Explorer - Mo
Mo...
Models
Meat Value Ontology Models
myMeatOntology 1

Figure 5.1: Semantically enriched queries

2) When we click on “Upload to Virtuoso” we get a message:

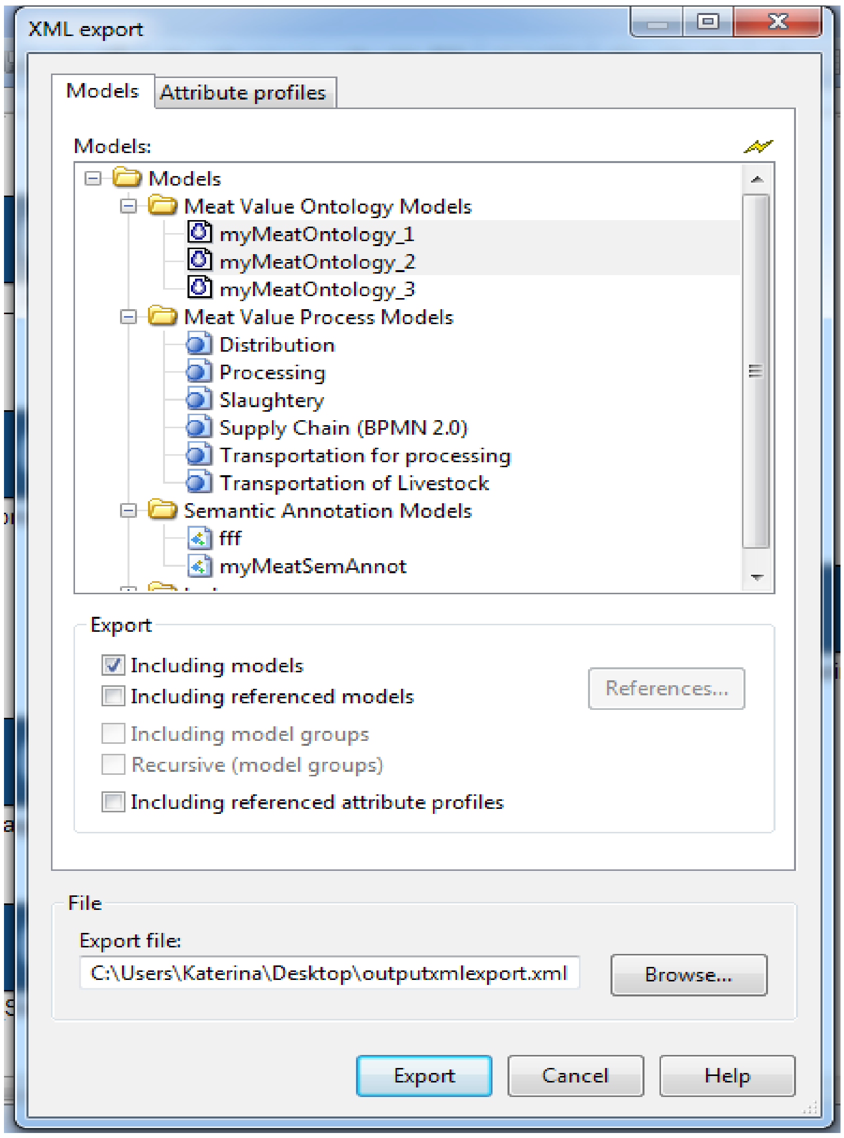

ADOxx Modelling Toolkit (semfis)

Please select your Ontology Model(s) for querying!

Figure 5.2: User selects ontology models for uploading on Virtuoso

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH)

3) Then we choose the ontology model(s) that we want to import on Virtuoso for querying. The tool exports the models as an XML file into our file system and then it transforms them into an OWL file and uploads it to Virtuoso. Of course prerequisite for this functionality is to have installed Virtuoso on our system.



Models:

- Models
  - Meat Value Ontology Models
    - myMeatOntology_1
    - myMeatOntology_2
    - myMeatOntology_3
  - Meat Value Process Models
    - Distribution
    - Processing
    - Slaughtery
    - Supply Chain (BPMN 2.0)
    - Transportation for processing
    - Transportation of Livestock
  - Semantic Annotation Models
    - fff
    - myMeatSemAnnot

Export

- [✓] Including models
- [ ] Including referenced models
- [ ] Including model groups
- [ ] Recursive (model groups)
- [ ] Including referenced attribute profiles

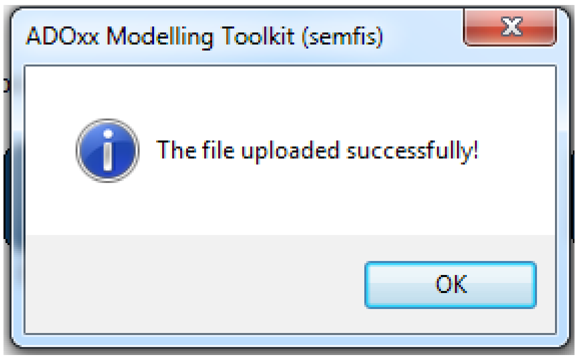
File

Export file: C:\Users\Katerina\Desktop\outputxmlexport.xml

Figure 5.3: Choose the ontology models for export

4) If the file uploaded successfully we get the following message otherwise we get an error message.

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH)


ADOxx Modelling Toolkit (semfis)

The file uploaded successfully!

OK

Figure 5.4: Successful uploading

After uploading the preferred ontology model(s) on Virtuoso, we would like to ask useful queries. So when we are clicking on the **“Query Ontology Models”** we get the following window that includes some predefined queries to choose so as to avoid writing the query from scratch. Also the user can choose the 7th entry that allows him to write a query on his own.

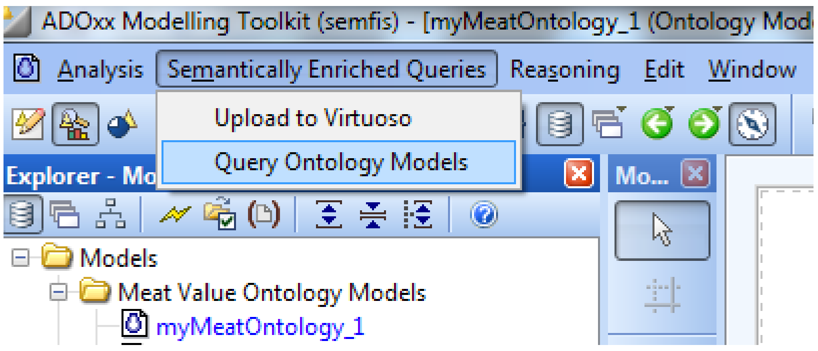
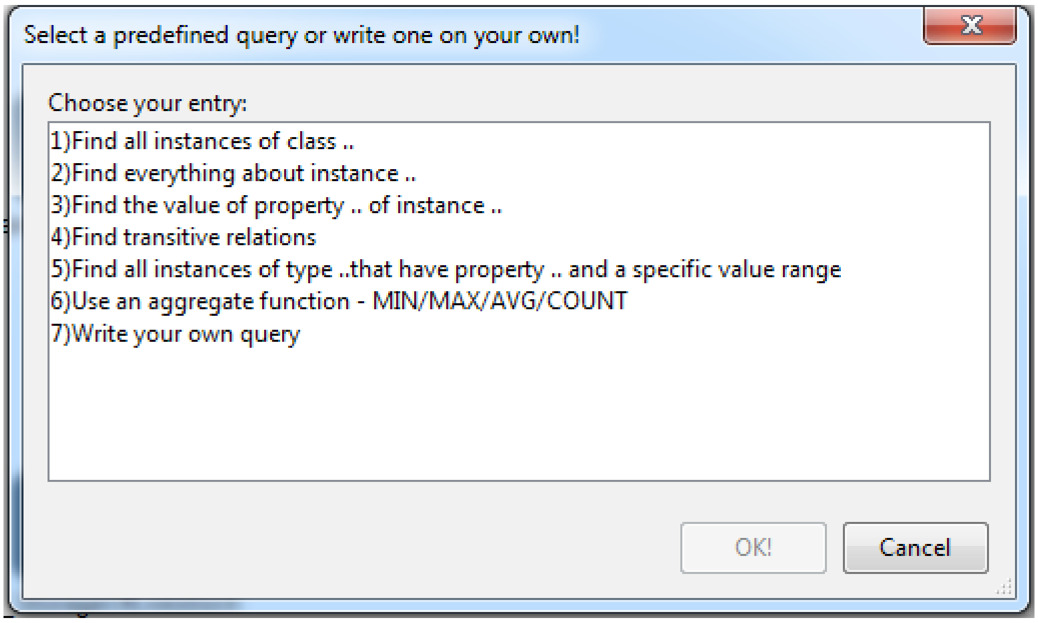

ADOxx Modelling Toolkit (semfis) - [myMeatOntology_1 (Ontology Mod
Analysis
Semantically Enriched Queries
Reasoning
Edit
Window
Upload to Virtuoso
Query Ontology Models
Explorer - Mo
Mo...
Models
Meat Value Ontology Models
myMeatOntology_1

Figure 5.5: Query Ontology Models

I chose the below predefined queries having in mind what a business designer would need more often while searching useful information, annotated on his BPMN process models.

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH)


Select a predefined query or write one on your own!

Choose your entry:

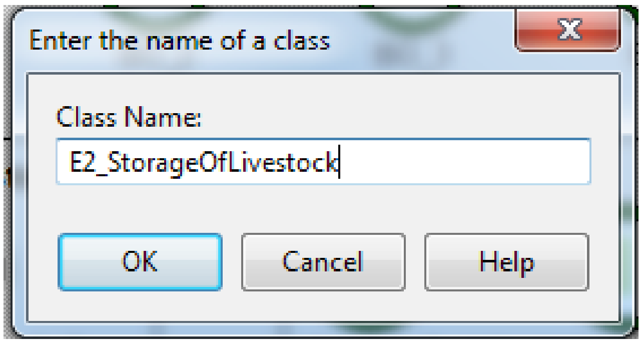
1)Find all instances of class ..
2)Find everything about instance ..
3)Find the value of property .. of instance ..
4)Find transitive relations
5)Find all instances of type ..that have property .. and a specific value range
6)Use an aggregate function - MIN/MAX/AVG/COUNT
7)Write your own query

Figure 5.6: List of predefined queries

e.g. Choosing option “1)Find all instances of class ..”

User is asked to enter the name of the class that he wants to find out instances of it.

An editbox is appeared and you can enter the name of your class

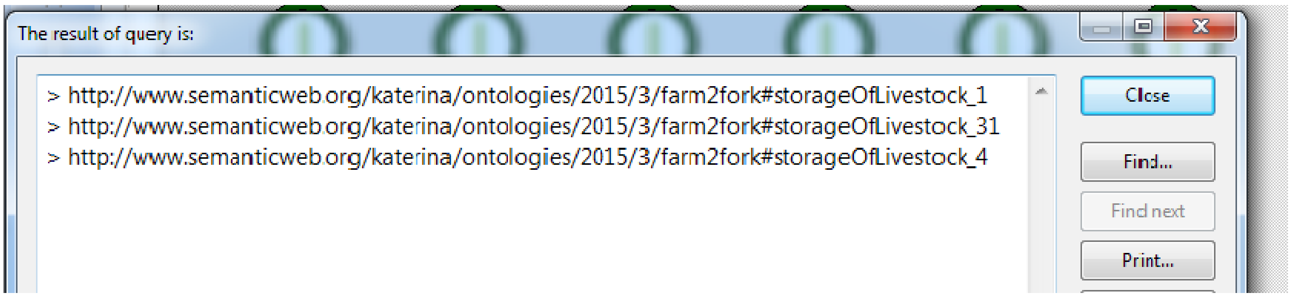

Enter the name of a class

Class Name: E2_StorageOfLivestock

Figure 5.7: Enter the name of your class

Then you get the results from the SPARQL query:

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH)



```
> http://www.semanticweb.org/katerina/ontologies/2015/3/farm2fork#storageOfLivestock_1
> http://www.semanticweb.org/katerina/ontologies/2015/3/farm2fork#storageOfLivestock_31
> http://www.semanticweb.org/katerina/ontologies/2015/3/farm2fork#storageOfLivestock_4
```

Figure 5.8: Results of predefined query 1

6) Attached files

The folder "Thesis_kdimitraki_final" contains the following subfolders:

- *ADOxx extension all files*

It contains the external coupling adds on and the exported library with all the extensions

- *ADOxx models exported*

Here, we concentrated in one file the use case models

- *Functionalities in java code*

Here, there are three java projects, "ConvertAndReason", it converts the ADOxx XML file into RDF/XML format and then conducts reasoning with Hermit reasoner and returns the explanations in case of inconsistency. "UploadToVirtuoso" uploads the converted ontology models on Virtuoso engine and "QueryVirtuoso" allows the user to run some predefined queries or write new ones on his own.

- *Queries*

The folder contains the .jar files with the described functionality as well as the .asc files with ADOscript code that are called from external coupling for the feature "**Semantically enriched queries**"

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH)

- *Reasoning*

The folder contains the .jar file with the described functionality as well as the .asc files with ADOscript code that are called from external coupling for the feature "**Reasoning**"

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH)