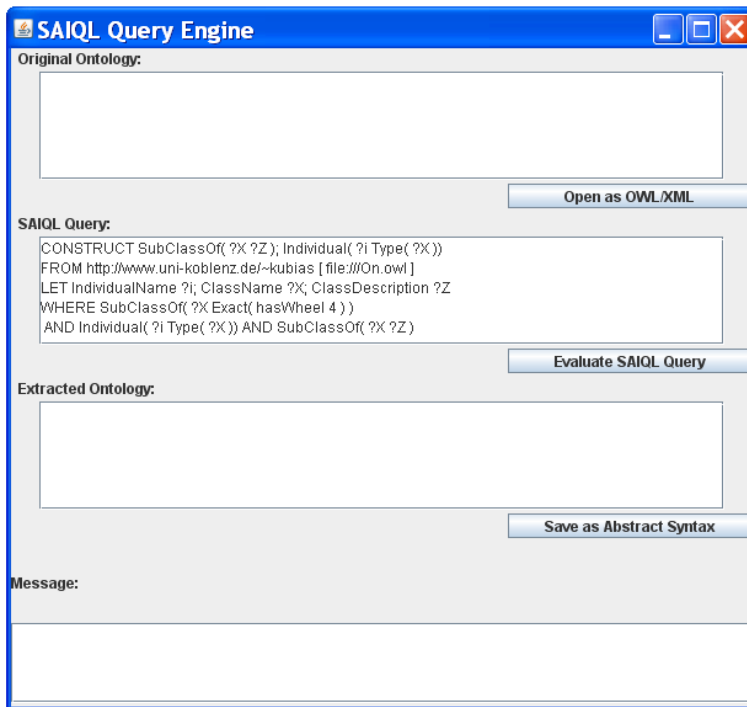


Manual for SAIQL Query Engine

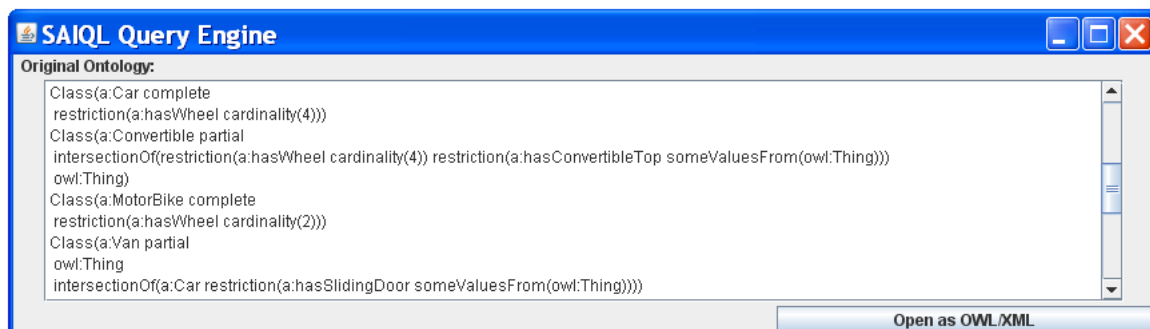
Start of the Application

After starting, the application should look like shown in the following figure.



Loading of the Original Ontology

The first step is to open or to load the Original Ontology. SAIQL extract elements from this ontology. Use the button „Open as OWL/XML“ in the section „Original Ontology“. Afterwards the ontology is loaded and shown as OWL Abstract Syntax in the memo-textfield, below the label „Original Ontology“. This situation is demonstrated in the next figure.



This action changed the SAIQL-expression in the next memo-textfield.
The line: FROM <http://www.uni-koblenz.de/~kubias>
[file:/C:/ISWEB/Software/workspaceSAIQL/SAIQLQueryEngine/MotorOntology.owl].

Entering the SAIQL Query

The next step is the input of the SAIQL Query in the second Memo-Textfield in the section „SAIQL Query“.

Note: The SAIQL-Syntax was changed in the meantime and is now as described in the SAIQL-Techreport-2007 and shortened demonstrated below:

```
OWL-SAIQL-query ::= 'CONSTRUCT' constructClause
                  'FROM' fromClause
                  'LET' letClause
                  'WHERE' whereClause

constructClause ::= axiomPattern {';' axiomPattern}
fromClause     ::= ontologyID
letClause      ::= variableBinding {';' variableBinding}
whereClause    ::= axiomPattern {'AND' axiomPattern}

axiomPattern   ::= classAxiomPattern | individualAxiomPattern

className     ::= URIreference
individualName ::= URIreference
ontologyID    ::= URIreference
indProperty   ::= URIreference

variableBinding ::= classNameBinding
                | individualNameBinding
                | classDescriptionBinding
                | indPropertyBinding
classNameBinding ::= 'ClassName' classNameVar {',' classNameVar}
individualNameBinding ::= 'IndividualName' individualNameVar {',' individualNameVar}
classDescriptionBinding ::= 'ClassDescription' classDescriptionVar
                          {',' classDescriptionVar}
individualPropertyBinding ::= 'IndividualProperty' indPropertyVar
                             {',' indPropertyVar}

lexicalForm ::= a unicode string in normal form C
classNameVar ::= '?'lexicalForm

individualNameVar ::= '?'lexicalForm
classDescriptionVar ::= '?'lexicalForm
indPropertyVar ::= '?'lexicalForm

classNameOrVar ::= classNameVar | className
indNameOrVar ::= individualNameVar | individualName
classDescOrVar ::= classDescVar | classDesc

classAxiomPattern ::= 'SubClassOf(' classDescOrVar classDescOrVar ')'
                  | 'DisjointClasses(' classDescOrVar classDescOrVar ')'
                  | 'EquivalentClasses(' classDescOrVar classDescOrVar ')'
```

```

classDesc ::= classNameOrVar
           | restriction
           | 'UnionOf(' {classDescOrVar } ')''
           | 'IntersectionOf(' { classDescOrVar } ')''
           | 'ComplementOf(' classDescOrVar ')''

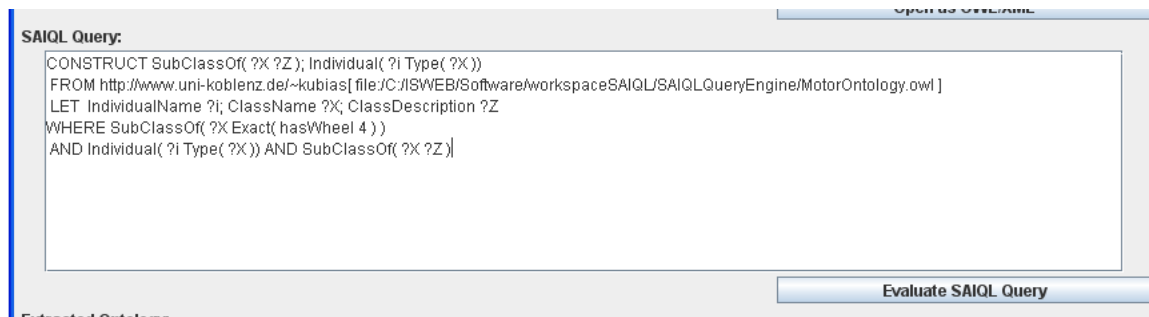
restriction ::= 'All(' indProperty classDescOrVar ')''
            | 'Some(' indProperty classDescOrVar ')''
            | 'Value(' indProperty indNameOrVar ')''
            | 'Min(' indProperty non-negative-integer ')''
            | 'Max(' indProperty non-negative-integer ')''
            | 'Exact(' indProperty non-negative-integer ')''

individualAxiomPattern ::= 'Individual(' indNameOrVar 'type(' classDescOrVar ')')''
                        | 'SameIndividual(' indNameOrVar indNameOrVar ')''
                        | 'DifferentIndividuals(' indNameOrVar indNameOrVar ')''

```

Evaluation of the SAIQL Query

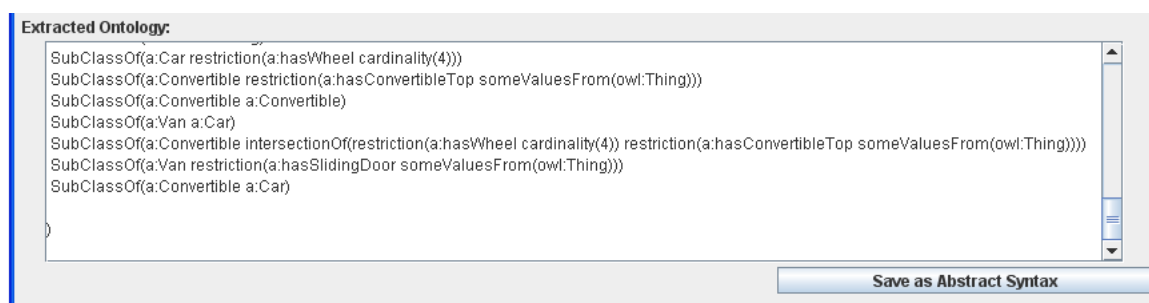
The next step is the evaluation of the SAIQL Query. Use the button “Evaluate SAIQL Query”. This section of the application is shown in the figure below.



The evaluation process will last for some time and could even pause the GUI during this action.

Display of the extracted ontology

The extracted ontology“ is shown in the Memo-Textfield „Extracted Ontology“ in OWL Abstract Syntax.



Save the extracted Ontology

The extracted ontology could be saved as OWL Abstract Syntax, using the „Save as Abstract Syntax“ button.