An English Verbalization Template
for ORM conceptual models and rules

Mustafa Jarrar, Vrije Universiteit Brussel, Belgium. (Contact Author)
Maria Keet, Free University of Bozen-Bolzano, Italy.

A technical report of the article[1]: *Jarrar, M., Keet, C.M., Dongilli, P.* ***Multilingual verbalization of ORM conceptual models and axiomatized ontologies****. [Submitted].*
URL: http://www.starlab.vub.ac.be/staff/mustafa/orm/verbalization/

Abstract. In the above-mentioned article we describe a novel approach to support *multilingual* verbalization of logical theories, axiomatizations, and other specifications such as business rules. This engineering solution is demonstrated with the Object Role Modeling language, although its underlying principles can be reused with other conceptual models and formal languages, such as Description Logics, to improve its understandability and usability by the domain expert. The engineering solution for multilingual verbalization is characterized by its flexibility, extensibility and maintainability of the verbalization templates, which allow for easy augmentation with other languages than the 11 currently supported.

This report presents the English verbalization template file. Given an ORM schema (or an ORM-ML file), and given the verbalization template, an English verbalization of the rules and fact types (in the schema) is generated automatically. A comprehensive example of an ORM schema and its corresponding verbalization is generated and given in this report.

1 Introduction

In the above-mentioned article, we present a novel approach to support multilingual verbalization of logical theories, formal axiomatizations, and other specifications such as business rules, ontologies, etc. We demonstrate our approach by providing a flexible and extensible verbalization template for the Object Role Modeling language. This template can be easily customized and translated into other human languages. *This technical report provides the English verbalization of the ORM models and rules.* The verbalization of several other languages (including, but not limited to: German, Italian, Arabic, Russian, Dutch, Spanish, French, Chinese, Vietnamese, and Lithuanian) can be found at the above-mentioned URL.

The underlying principles of our approach can be reused for other conceptual models and formal languages, such as Description Logics. The objective was to define a template parameterized over a given set of rules, models, or axioms, with as output fixed-syntax pseudo natural language sentences. A simple example is the following: the formal rule

$$\forall x\ (Book(x) \rightarrow \exists y\ (ISBN(y) \wedge Has(x,y)))$$

can be translated into

It is mandatory that each **Book Has** an **ISBN**.

[1] For Citation use: *Jarrar, M., Keet, C.M.: An English Verbalization Template for ORM conceptual models and rules. A technical report of the article: Jarrar, M., Keet, C.M., Dongilli, P. Multilingual verbalization of ORM conceptual models and axiomatized ontologies. [Submitted].*

In this way, we enable domain experts themselves to build and/or validate the formal specifications of their domains, without having to know that these sentences are formal axioms; i.e. the underpinning logics and reasoning services are hidden from the user. Our approach with the provided templates can be reused in modeling business rules, ontologies, knowledge bases, etc. See [H04] for a similar approach to ORM business rules verbalization.

In the following section, we present an ORM example followed by verbalization of all rules in this ORM schema. These verbalizations are generated *automatically*, according to the English verbalization template presented in section 3. This approach is fully implemented and supported in the DogmaModeler ontology modeling tool [J05]. It is worth noting that DogmaModeler's automated verbalization has been used by tens of lawyers to build the Customer Complaint Ontology [J05][JVM03].

Remark on Modality: Our verbalization template can be adapted easily according to the application/reasoning scenario, whether it is used as integrity constraints, derivation/inference rules, business rules, etc. For example, the above mandatory constraint can be verbalized in different ways, such as: 1) Each **Book** must **Has** at least one **ISBN**. 2) Each **Book** **Has** some **ISBN** values. 3) If there is a **Book** then it **Has** an **ISBN** value. 4) A **Book** that does not **Has** an **ISBN** is not allowed. 5) If a **Book** does not **Has** an **ISBN** value then....

2 Example of an ORM Schema

We illustrate in one diagram most types of rules supported in ORM. Our article [JKD06] describes technical details on *how* our verbalization approach is implemented, see [H01] to know more about ORM, and [J05] to know about the DogmaModeler tool that we use to build and automatically verbalize ORM models.

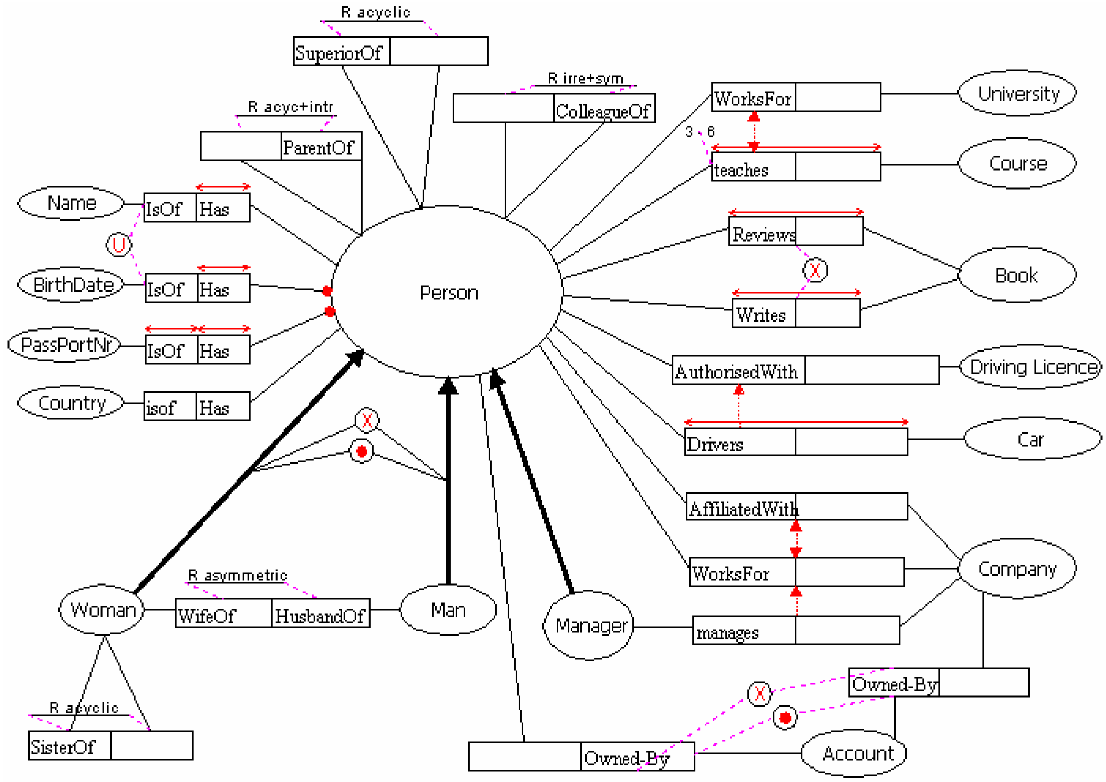

Fig. 1. Example of ORM rules, in English.

The constraints/rules in the above ORM example are verbalized automatically (according to the verbalization template presented in section 2):

```
-[Mandatory]  Each Person must Has at least one PassPortNr.
-[Mandatory]  Each Person must Has at least one BirthDate.
-[Mandatory]  Each Account should be  Owned-By Company or Owned-By Person.
-[Uniqueness]  Each Person must Has at most one BirthDate.
-[Uniqueness]  Each Person must Has at most one Name.
-[Uniqueness]  Each Person must Has at most one PassPortNr.
-[Uniqueness]  Each PassPortNr must IsOf at most one Person.
-[Uniqueness]  It is possible that  Person teaches more than one Course , and vice versa.
-[Uniqueness]  It is possible that  Person Reviews more than one Book , and vice versa.
-[Uniqueness]  It is possible that  Person Writes more than one Book , and vice versa.
-[Uniqueness]  It is possible that  Person Drivers more than one Car , and vice versa.
-[Uniqueness]  The combination of { BirthDate and Name } must refer to at most one Person.
-[Exclusive]  Each Person should be either Woman or Man.
-[Totality]  Each Person must be, at least,  Man or Woman.
-[Subset]  If Person  Drivers Car then this Person  AuthorisedWith Driving Licence.
-[Subset]  If Manager  manages Company then this Person  WorksFor that Company.
-[Equality]  Person  WorksFor University if and only if this Person  teaches Course.
-[Equality]   Person  AffiliatedWith Company if and only if this Person  WorksFor that Company.
-[Exclusion]  No Account  Owned-By Company and also  Owned-By Person.
-[Exclusion]  No Person  Writes Book and also  Reviews that Book.
-[Value] The possible instances of Country are :{Belgium, France, Germany}
-[Irreflexive] No Person ColleagueOf it/him self.
-[Symmetric] If Person X ColleagueOf Person Y, it must be vice versa.
-[Acyclic]   Person cannot be directly (or indirectly through a chain)  SuperiorOf it/him self .
-[Acyclic]   Woman cannot be directly (or indirectly through a chain)  SisterOf it/him self .
-[Asymmetric]  If Person X  WifeOf Person Y, it cannot be vice versa .
-[Intransitive] If Person X ParentOf Person Y, and Y ParentOf Z, then it cannot be that X
ParentOf Z.
-[Frequency] If Person Teaches Course, then this Person Teaches at least 3 and at most 6
Course(s).
```

3 The English Verbalization Template

The template is presented in an XML syntax, it is being implemented in the DogmaModeler tool to support the English verbalization of ORM models. More details about this approach can be found [JKD06], and refer to [J05] about DogmaModeler.

```
<?xml version='1.0' encoding='UTF-8'?>
<ORMSchema xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:noNamespaceSchemaLocation='http://www.starlab.vub.ac.be/staff/mustafa/orm
/verbalization/'>
```

```
<ORMNLMeta>
 <Meta name="DC.Title" content="English verbalization template (Ver0.3)"/>
 <Meta name="DC.Version" content="0.3"/>
 <Meta name="DC.Creator" content="Mustafa Jarrar"/>
 <Meta name="DC.Language" content="English"/>
 </ORMNLMeta>

<ORMNLBody>

<FactType xsi:type="FactType" >
<Text></Text>
<Object index="0" />
<Role index="0" />
<Text>/</Text>
<Role index="1" />
<Text></Text>
<Object index="1" />
</FactType>

<Constraint xsi:type="Mandatory">
 <Text> -[Mandatory]  Each</Text>
 <Object index="0"/>
 <Text>must</Text>
 <Role index="0"/>
 <Text>at least one</Text>
 <Object index="1"/>
</Constraint>

<Constraint xsi:type="Backward Mandatory">
 <Text> -[M]  For each</Text>
 <Object index="0"/>
 <Text>there is at least one</Text>
 <Object index="1"/>
 <Text>that</Text>
 <Role index="1"/>
 <Text>this</Text>
 <Object index="0"/>
</Constraint>

<Constraint xsi:type="Disjunctive Mandatory">
 <Text> -[Mandatory]  Each</Text>
 <Object index="0"/>
 <Text>should be</Text>
 <Role index="0"/>
 <Object index="1"/>
 <Loop index="1" >
  <Text>or</Text>
  <Role index="n"/>
  <Object index="n"/>
 </Loop>
</Constraint>

<Constraint xsi:type="Uniqueness">
 <Text> -[Uniqueness]  Each</Text>
 <Object index="0"/>
 <Text>must</Text>
 <Role index="0"/>
 <Text>at most one</Text>
 <Object index="1"/>
</Constraint>

<Constraint xsi:type="Backward Uniqueness">
 <Text> -[Uniqueness]  For each</Text>
 <Object index="0"/>
 <Text>there must be at most one</Text>
 <Object index="1"/>
 <Text>that</Text>
 <Role index="1"/>
 <Text>this</Text>
 <Object index="0"/>
</Constraint>

<Constraint xsi:type="Many Uniqueness">
 <Text> -[Uniqueness]  It is possible that </Text>
 <Object index="0"/>
<Role index="0"></Role>
 <Text>more than one</Text>
 <Object index="1"/>
 <Text>, and vice versa</Text>
</Constraint>

<Constraint xsi:type="External Uniqueness">
 <Text> -[Uniqueness]  The combination of {</Text>
 <Object index="1"/>
 <Loop index="1">
  <Text>and</Text>
  <Object index="n"/>
 </Loop>
 <Text>} must refer to at most one</Text>
 <Object index="0"/>
</Constraint>

<Constraint xsi:type="Subtype">
 <Text> -[Subtype]  Each instance</Text>
 <Object index="child"/>
 <Text>is also an instance of</Text>
 <Object index="parent"/>
</Constraint>

<Constraint xsi:type="Value">
 <Text> -[Value]  The possible instances of </Text>
 <Object index="0"/>
 <Text> are :{</Text>
 <Value index="0"/>
 <Loop index="1">
  <Text>,</Text>
  <Value index="n"/>
 </Loop>
 <Text> }</Text>
 </Constraint>

<Constraint xsi:type="Exclusive">
<Text> -[Exclusive]  Each</Text>
<Object index="0"/>
```

```
<Text>should be either</Text>
<Object index="1"/>
<Loop index="1">
  <Text>or</Text>
  <Object index="n"/>
</Loop>
</Constraint>

<Constraint xsi:type="Total">
 <Text> -[Totality]  Each</Text>
 <Object index="0"/>
 <Text>must be, at least, </Text>
 <Object index="1"/>
 <Loop index="1">
  <Text>or</Text>
  <Object index="n"/>
 </Loop>
</Constraint>

<Constraint xsi:type="Partition">
 <Text> -[Partition]  Each</Text>
 <Object index="0"/>
 <Text>is at least one of</Text>
 <Object index="1"/>
 <Loop index="1">
  <Text>or</Text>
  <Object index="n"/>
 </Loop>
 <Text>but not all</Text>
</Constraint>

<Constraint xsi:type="Subset">
 <Text> -[Subset]  If</Text>
 <Object index="0"/>
 <Role index="child"/>
 <Object index="child"/>
 <Text>then this</Text>
 <Object index="0"/>
 <Role index="parent"/>
 <Object index="parent"/>
</Constraint>

<Constraint xsi:type="Subset FactType">
 <Text> -[Subset]  If</Text>
 <Object index="0"/>
 <Role index="child"/>
 <Object index="child"/>
 <Text>then this</Text>
 <Object index="1"  />
 <Role index="parent"/>
 <Text>that</Text>
 <Object index="parent"/>
</Constraint>

<Constraint xsi:type="Equality">
 <Text> -[Equality]  </Text>
 <Object index="0"/>
 <Role index="first"/>
 <Object index="first"/>
 <Text>if and only if</Text>
 <Text>this </Text>
 <Object index="0"/>
 <Role index="second"/>
 <Object index="second"/>
</Constraint>

<Constraint xsi:type="Equality FactType">
 <Text> -[Equality]  </Text>
 <Object index="0"/>
 <Role index="First"/>
 <Object index="First"/>
 <Text>if and only if</Text>
 <Text>this</Text>
 <Object index="1"/>
 <Role index="Second"/>
 <Text>that</Text>
 <Object index="Second"/>
</Constraint>

<Constraint xsi:type="Exclusion">
 <Text> -[Exclusion]  No</Text>
 <Object index="0"/>
 <Role index="first"/>
 <Object index="first"/>
 <Text>and also</Text>
 <Role index="second"/>
 <Object index="second"/>
</Constraint>

<Constraint xsi:type="Exclusion FactType">
 <Text> -[Exclusion]  No</Text>
 <Object index="0"/>
 <Role index="first"/>
 <Object index="first"/>
 <Text>and also</Text>
 <Role index="second"/>
 <Text>that</Text>
 <Object index="second"/>
</Constraint>

<Constraint xsi:type="Frequency">
 <Text> -[Frequency] If </Text>
 <Object index="0"/>
 <Role index="0"/>
 <Object index="1"/>
 <Role index="0"/>
 <Text>, then this </Text>
 <Object index="0"/>
 <Role index="0"/>
 <Text>at least </Text>
 <Minimum/>
 <Text> and most most </Text>
 <Maximum/>
 <Role index="0"/>
 <Text>(s)</Text>
</Constraint>

<Constraint xsi:type="Irreflexive">
 <Text> -[Irreflexive]  No</Text>
 <Object index="0"/>
 <Role index="0"/>
 <Text> it/him self</Text>
</Constraint>

<Constraint xsi:type="Symmetric">
 <Text> -[Symmetric]  If</Text>
 <Object index="0"/>
 <Text>X</Text>
 <Role index="0"/>
 <Object index="0"/>
 <Text>Y</Text>
 <Text>, it must be vice versa</Text>
</Constraint>

<Constraint xsi:type="Asymmetric">
 <Text> -[Asymmetric]  If</Text>
 <Object index="0"/>
 <Text>X</Text>
 <Role index="0"/>
 <Object index="0"/>
<Text> Y, it cannot be vice versa</Text>
</Constraint>

<Constraint xsi:type="Acyclic">
 <Text> -[Acyclic]  </Text>
 <Object index="0"/>
 <Text> cannot be directly (or indirectly through a chain)</Text>
 <Role index="0"/>
 <Text> it/him self</Text>
</Constraint>

<Constraint xsi:type="Transitve">
 <Text> -[Intransitve]  If</Text>
 <Object index="0"/>
 <Text>X</Text>
 <Role index="0"/>
 <Object index="0"/>
 <Text>Y, and Y</Text>
 <Role index="0"/>
 <Text> Z, then it cannot be that X</Text>
 <Role index="0"/>
 <Text>Z</Text>
</Constraint>

</ORMNLBody>
</ORMSchema>
```

Acknowledgments

We are in debt to Andriy Lisovoy who helped in the implementation of DogmaModeler, and to Hai Nguyen Hoang who helped in the first implementation of the verbalization component during his Master thesis. This work is partially supported by the EU Knowledge Web NoE project (IST-2004-507482).

References

[JKD06] Jarrar, M., Keet, C.M., Dongilli, P. Multilingual verbalization of ORM conceptual models and axiomatized ontologies. [Submitted].

[J05] Jarrar, M.: Towards Methodological Principles for Ontology Engineering. PhD thesis, Vrije Universiteit Brussel, 2005.

[JVM03] Jarrar, M., Verlinden, R., Meersman, R.: Ontology-based Customer Complaint Management. In: Jarrar M., Salaun A., (eds.): Proceedings of the workshop on regulatory ontologies and the modeling of complaint regulations, Catania, Sicily, Italy. Springer Verlag LNCS. Vol. 2889. November (2003) pp. 594–606

[H01] Halpin, T.: Information Modeling and Relational Databases. 3rd ed. Morgan-Kaufmann. (2001)

[H04] Halpin, T.: Business Rule Verbalization. In Doroshenko, A., Halpin, T., Liddle, S., Mayr H. (eds): Information Systems Technology and its Applications, 3rd International Conference (ISTA'2004), LNI 48 GI ISBN 3-88579-377-6, (2004) pp:39-52.