A Catalan Verbalization Template
for ORM conceptual models and rules

Mustafa Jarrar, Vrije Universiteit Brussel, Belgium. (Contact Author)
Maria Keet, Free University of Bozen-Bolzano, Italy.
Núria Casellas, Universidad Autónoma de Barcelona, Spain.

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 Catalan verbalization template file. Given an ORM schema (or an ORM-ML file), and given the verbalization template, a Catalan 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 Catalan verbalization of the ORM models and rules.* The verbalization of several other languages (including, but not limited to: German, Italian, Arabic, Russian, Spanish, French, 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

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

can be translated into

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

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

[1] For Citation use: *Jarrar, M., Keet, C.M., Casellas, N.: A Catalan 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].*

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 Catalan 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.

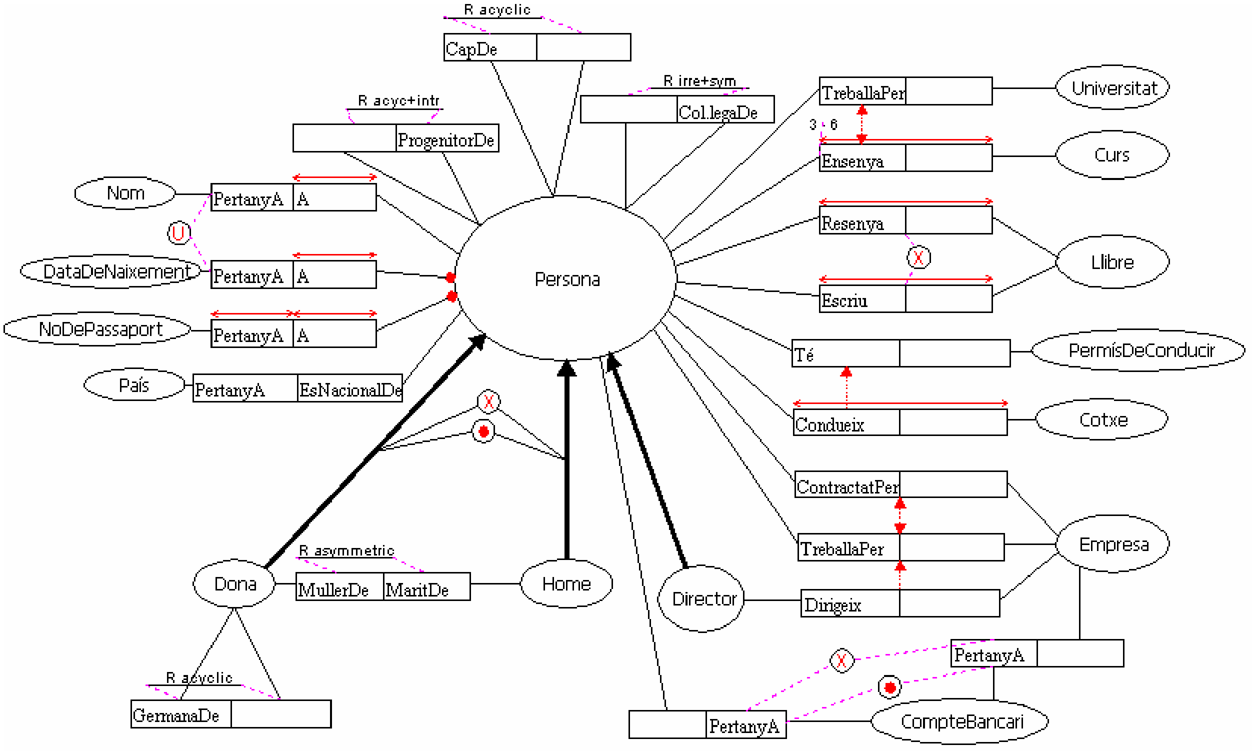

Fig. 1. Example of ORM rules, in Catalan.

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

```
-[Mandatory] Cada Persona ha de A almenys un NoDePassaport.
-[Mandatory] Cada Persona ha de A almenys un DataDeNaixement.
-[Mandatory]  Cada CompteBancari hauria de   PertanyA Empresa o PertanyA Persona.
-[Uniqueness]  Cada Persona ha de A com a màxim un(a) DataDeNaixement.
-[Uniqueness]  Cada Persona ha de A com a màxim un(a) Nom.
-[Uniqueness]  Cada Persona ha de A com a màxim un(a) NoDePassaport.
-[Uniqueness]  Cada NoDePassaport ha de PertanyA com a màxim un(a) Persona.
-[Uniqueness]  És possible que Persona Ensenya més d'un(a) Curs , y a l'inrevés.
-[Uniqueness]  És possible que Persona Resenya més d'un(a) Llibre , y a l'inrevés.
-[Uniqueness]  És possible que Persona Escriu més d'un(a) Llibre , y a l'inrevés.
-[Uniqueness]  És possible que Persona Condueix més d'un(a) Cotxe , y a l'inrevés.
-[Uniqueness]  La combinació de { DataDeNaixement i Nom } ha de referir-se només a un Persona.
-[Exclusive]  Cada Persona hauria de ser o bé Dona o Home.
-[Totality]  Cada Persona ha de ser, almenys, Home o Dona.
-[Subset]  Si Persona  Condueix Cotxe llavors aquest(a) Persona  Té PermísDeConducir.
-[Subset]  Si Director  Dirigeix Empresa llavors aquest(a) Persona  TreballaPer aquell(a) Empresa.
-[Equality]  Persona  TreballaPer  Universitat si i només si aquest(a) Persona  Ensenya  Curs.
-[Equality]   Persona  ContractatPer Empresa si i només si aquest(a) Persona  TreballaPer aquell(a)
Empresa.
-[Exclusion]  Cap CompteBancari  PertanyA  Persona i també  PertanyA  Empresa.
-[Exclusion]  Cap  Persona  Escriu Llibre i també  Resenya aquest(a) Llibre.
-[Irreflexive] Cap Persona Col.legaDe ell(a) mateix(a).
-[Symmetric] Si Persona X Col.legaDe Persona Y, ha de ser també a la inversa.
-[Acyclic] Persona no pot ser directament (o indirectament a través d'una cadena) CapDe ell(a) mateix(a).
-[Acyclic] Dona no pot ser directament (o indirectament a través d'una cadena)  GermanaDe ell(a)
mateix(a).
-[Asymmetric]  Si Persona X  MullerDe  Persona Y, no pot ser a la inversa .
-[Intransitive] Si Persona X ProgenitorDe Persona Y, e Y ProgenitorDe Z, llavors no pot ser que X
ProgenitorDe Z.
-[Frequency] Si Persona Ensenya Curs, llavors aquest(a) Persona Ensenya almenys 3 i com a màxim 6
Curs(os).
```

3 The Catalan Verbalization Template

The template is presented in an XML syntax, it is being implemented in the DogmaModeler tool to support the Catalan 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="Catalan verbalization template (Ver0.1)"/>
 <Meta name="DC.Version" content="0.1"/>
 <Meta name="DC.Creator" content="Mustafa Jarrar"/>
 <Meta name="DC.Contributor" content="Maria Keet"/>
 <Meta name="DC.Contributor" content="Núria Casellas"/>
 <Meta name="DC.Language" content="Catalan"/>
 </ORMNLMeta>

<ORMNLBody>

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

<Constraint xsi:type="Mandatory">
<Text> -[Mandatory] Cada</Text>
<Object index="0"/>
<Text>ha de</Text>
<Role index="0"/>
<Text>almenys un</Text>
<Object index="1"/>
</Constraint>

<Constraint xsi:type="Backward Mandatory">
<Text> -[M]  Per cada</Text>
<Object index="0"/>
<Text>hi ha almenys un</Text>
<Object index="1"/>
<Text>que</Text>
<Role index="1"/>
<Text>aquest(a)</Text>
<Object index="0"/>
</Constraint>

<Constraint xsi:type="Disjunctive Mandatory">
<Text> -[Mandatory]  Cada</Text>
<Object index="0"/>
<Text>hauria de </Text>
<Role index="0"/>
<Object index="1"/>
<Loop index="1">
<Text>o</Text>
<Role index="n"/>
<Object index="n"/>
</Loop>
</Constraint>

<Constraint xsi:type="Uniqueness">
<Text> -[Uniqueness]  Cada</Text>
<Object index="0"/>
<Text>ha de</Text>
<Role index="0"/>
<Text>com a màxim un(a)</Text>
<Object index="1"/>
</Constraint>
```

```
<Constraint xsi:type="Backward Uniqueness">
<Text> -[Uniqueness]  Per cada</Text>
<Object index="0"/>
<Text>hi ha d’haver només un(a)</Text>
<Object index="1"/>
<Text>aquell(a)</Text>
<Role index="1"/>
<Text>aquest(a)</Text>
<Object index="0"/>
</Constraint>

<Constraint xsi:type="Many Uniqueness">
<Text> -[Uniqueness]  És possible que</Text>
<Object index="0"/>
<Role index="0"/>
<Text>més d’un(a)</Text>
<Object index="1"/>
<Text>, y a l’inrevés</Text>
</Constraint>

<Constraint xsi:type="External Uniqueness">
<Text> -[Uniqueness]  La combinació de {</Text>
<Object index="1"/>
<Loop index="1">
<Text>i</Text>
<Object index="n"/>
</Loop>
<Text>} ha de referir-se només a un</Text>
<Object index="0"/>
</Constraint>

<Constraint xsi:type="Subtype">
<Text> -[Subtype]  Cada instància</Text>
<Object index="child"/>
<Text>és també una instància de</Text>
<Object index="parent"/>
</Constraint>

<Constraint xsi:type="Value">
<Text> -[Value] Les possibles instàncies de</Text>
<Object index="0"/>
<Text> són :{</Text>
<Value index="0"/>
<Loop index="1">
<Text>,</Text>
<Value index="n"/>
</Loop>
<Text> }</Text>
</Constraint>

<Constraint xsi:type="Exclusive">
<Text> -[Exclusive]  Cada</Text>
<Object index="0"/>
<Text>hauria de ser o bé</Text>
<Object index="1"/>
<Loop index="1">
```

```
<Text>o</Text>
<Object index="n"/>
</Loop>
</Constraint>

<Constraint xsi:type="Total">
<Text> -[Totality]  Cada</Text>
<Object index="0"/>
<Text>ha de ser, almenys,</Text>
<Object index="1"/>
<Loop index="1">
<Text>o</Text>
<Object index="n"/>
</Loop>
</Constraint>

<Constraint xsi:type="Partition">
<Text> -[Partition]  Cada</Text>
<Object index="0"/>
<Text>és com a mínim un de</Text>
<Object index="1"/>
<Loop index="1">
<Text>o</Text>
<Object index="n"/>
</Loop>
<Text>però no tots</Text>
</Constraint>

<Constraint xsi:type="Subset">
<Text> -[Subset]  Si</Text>
<Object index="0"/>
<Role index="child"/>
<Object index="child"/>
<Text>llavors aquest(a)</Text>
<Object index="0"/>
<Role index="parent"/>
<Object index="parent"/>
</Constraint>

<Constraint xsi:type="Subset FactType">
<Text> -[Subset]  Si</Text>
<Object index="0"/>
<Role index="child"/>
<Object index="child"/>
<Text>llavors aquest(a)</Text>
<Object index="1"/>
<Role index="parent"/>
<Text>aquell(a)</Text>
<Object index="parent"/>
</Constraint>

<Constraint xsi:type="Equality">
<Text> -[Equality]  </Text>
<Object index="0"/>
<Role index="first"/>
<Text> </Text>
<Object index="first"/>
```

```
<Text>si i només si</Text>
<Text>aquest(a) </Text>
<Object index="0"/>
<Role index="second"/>
<Text> </Text>
<Object index="second"/>
</Constraint>

<Constraint xsi:type="Equality FactType">
<Text> -[Equality]  </Text>
<Object index="0"/>
<Role index="First"/>
<Object index="First"/>
<Text>si i només si</Text>
<Text>aquest(a)</Text>
<Object index="1"/>
<Role index="Second"/>
<Text>aquell(a)</Text>
<Object index="Second"/>
</Constraint>

<Constraint xsi:type="Exclusion">
<Text> -[Exclusion]  Cap </Text>
<Object index="0"/>
<Role index="first"/>
<Text> </Text>
<Object index="first"/>
<Text>i també</Text>
<Role index="second"/>
<Text> </Text>
<Object index="second"/>
</Constraint>

<Constraint xsi:type="Exclusion FactType">
<Text> -[Exclusion]  Cap </Text>
<Object index="0"/>
<Role index="first"/>
<Object index="first"/>
<Text>i també</Text>
<Role index="second"/>
<Text>aquest(a)</Text>
<Object index="second"/>
</Constraint>

<Constraint xsi:type="Frequency">
<Text> -[Frequency] Si </Text>
<Object index="0"/>
<Role index="0"/>
<Object index="1"/>
<Role index="0"/>
<Text>, llavors aquest(a)</Text>
<Object index="0"/>
<Role index="0"/>
<Text>almenys</Text>
<Minimum/>
<Text>i com a màxim</Text>
<Maximum/>
```

```
<Role index="0"/>
<Text>(os)</Text>
</Constraint>

<Constraint xsi:type="Irreflexive">
<Text> -[Irreflexive]  Cap</Text>
<Object index="0"/>
<Role index="0"/>
<Text>ell(a) mateix(a)</Text>
</Constraint>

<Constraint xsi:type="Symmetric">
<Text> -[Symmetric]  Si</Text>
<Object index="0"/>
<Text>X</Text>
<Role index="0"/>
<Object index="0"/>
<Text>Y</Text>
<Text>, ha de ser també a la inversa</Text>
</Constraint>

<Constraint xsi:type="Asymmetric">
<Text> -[Asymmetric]  Si</Text>
<Object index="0"/>
<Text> X </Text>
<Role index="0"/>
<Text> </Text>
<Object index="0"/>
<Text> Y, no pot ser a la inversa</Text>
</Constraint>

<Constraint xsi:type="Acyclic">
<Text> -[Acyclic]  </Text>
<Object index="0"/>
<Text> no pot ser directament (o indirectament a través d’una cadena)</Text>
<Role index="0"/>
<Text> ell(a) mateix(a)</Text>
</Constraint>

<Constraint xsi:type="Transitve">
<Text> -[Intransitve]  Si</Text>
<Object index="0"/>
<Text>X</Text>
<Role index="0"/>
<Object index="0"/>
<Text>Y, e Y</Text>
<Role index="0"/>
<Text> Z, llavors no pot ser que 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.