

A Russian Verbalization Template for ORM conceptual models and rules

Mustafa Jarrar, Vrije Universiteit Brussel, Belgium. (Contact Author)

Andriy Lisovoy, Vrije Universiteit Brussel, Belgium.

Vladislav Ryzhikov, Free University of Bozen-Bolzano, Italy.

Maria Keet, Free University of Bozen-Bolzano, Italy.

A technical report of the article¹: *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 Russian verbalization template file. Given an ORM schema (or an ORM-ML file), and given the verbalization template, a Russian 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 Russian verbalization of the ORM models and rules.* The verbalization of several other languages (including, but not limited to: German, Italian, English, Arabic, 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

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

can be translated into

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

¹ For Citation use: *Jarrar, M., Lisovoy, A., Ryzhikov, V., Keet, C.M.: A Russian 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 Russian 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.

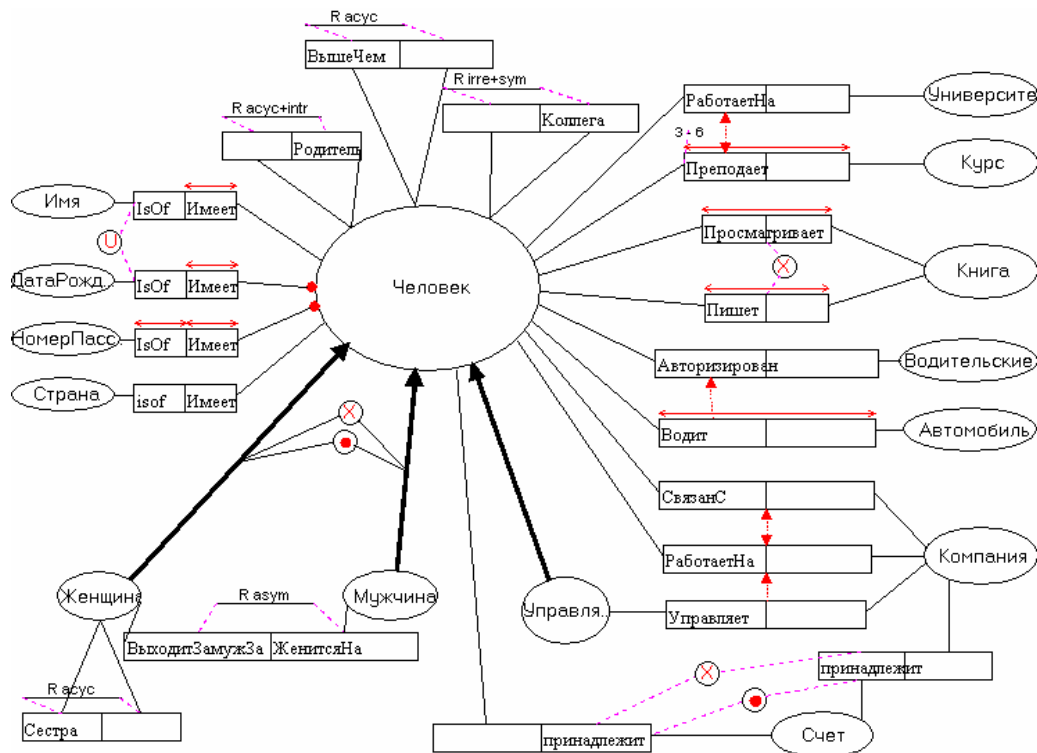


Fig. 1. Example of ORM rules, in Russian.

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

Каждый Человек Имеет по крайней мере один НомерПаспорта.
Каждый Человек Имеет по крайней мере один ДатаРождения.
Каждая комбинация ДатаРождения и Имя относится только к одному Человек.
Каждый Человек может быть Женщина или Мужчина.
Каждый Человек является либо Женщина или Мужчина.
Если Человек Водит Автомобиль , то Человек Авторизирован ВодительскиеПрава.
Человек РаботаетНаУниверситет если и только если этот Человек ПреподаетКурс , и наоборот.
Человек СвязанС Компания если и только если этот Человек РаботаетНа Компания.
Если Управляющий Управляет Компания , то этот Человек РаботаетНа тот Компания.
Не существует Счет , который принадлежитЧеловек и принадлежитКомпания.
Не существует Человек , который Пишет Книга и Просматривает тот Книга..
Счет either принадлежит или Человек или принадлежит Компания.
Каждый Человек Имеет максимум один ДатаРождения.
Каждый Человек Имеет максимум один Имя.
Каждый Человек Имеет максимум один НомерПаспорта.
Каждый НомерПаспорта IsOf максимум один Человек.
возможно, что Человек Преподает больше, чем один Курс и, что Курс Преподает больше, чем один Человек.
возможно, что Человек Просматривает больше, чем один Книга и, что Книга Просматривает больше, чем один Человек.
возможно, что Человек Пишет больше, чем один Книга и, что Книга Пишет больше, чем один Человек.
возможно, что Человек Водит больше, чем один Автомобиль и, что Автомобиль Водит больше, чем один Человек.
Если Человек X Коллега Человек Y, то наоборот
Не существует Человек, который Коллега самого себя
Страна может быть представлен как: {Бельгия, Франция, Германия}
Человек не может быть прямо (или косвенно через связи) ВышеЧем самого себя
Женщина не может быть прямо (или косвенно через связи) Сестра самого себя
Если Человек X ВыходитЗамужЗа Человек Y, то не может быть наоборот
Если Человек X Родитель Человек Y и Y Родитель Z, то невозможно что X родитель Z
Если Человек Преподает Курс, то этот человек Преподает минимум 3 и максимум 6 Курс(ы)

3 The Russian Verbalization Template

The template is presented in an XML syntax, it is being implemented in the DogmaModeler tool to support the Russian 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'>

<ORMNLMeta>
<Meta name="DC.Title" content="Russian verbalization template"/>
<Meta name="DC.Version" content="0.1"/>
<Meta name="DC.Creator" content="Mustafa Jarrar"/>
  <Meta name="DC.Contributor" content="Andriy Lisovoy"/>
<Meta name="DC.Language" content="Russian"/>
</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="Lexical" >
<Text>Лексическими концепциями являются :{</Text>
  <Object index="0" />
  <Loop index="1">
    <Text>,</Text>
    <Object index="n" />
  </Loop>
  <Text> }</Text>
</Constraint>

<Constraint xsi:type="Value" >
<Object index="0" />
<Text> может быть представлен как :{</Text>
<Value index="0" />
<Loop index="1">
<Text>,</Text>
<Value index="n" />
</Loop>
<Text> }</Text>
</Constraint>

<Constraint xsi:type="Mandatory" >
<Text>Каждый</Text>
<Object index="0" />
<Role index="0" />
<Text> по крайней мере один</Text>
<Object index="1" />
</Constraint>

<Constraint xsi:type="Backward Mandatory" >
<Text>Для каждого</Text>
<Object index="0" />
<Text> существует по крайней мере один </Text>
<Object index="1" />
<Text> который </Text>
<Role index="1" />
<Text> этот</Text>
<Object index="0" />
</Constraint>

<Constraint xsi:type="Disjunctive Mandatory" >
<Object index="0" />
<Text>either</Text>
<Role index="0" />
<Text>или</Text>
<Object index="1" />
<Loop index="1" >
  <Text>или </Text>
  <Role index="n" />
</Loop>
<Object index="n" />
```

```
</Loop>
</Constraint>

<Constraint xsi:type="Uniqueness" >
<Text>Каждый</Text>
<Object index="0" />
<Role index="0" />
<Text> максимум один </Text>
<Object index="1" />
</Constraint>

<Constraint xsi:type="Backward Uniqueness" >
<Text>Для каждого </Text>
<Object index="0" />
<Text> существует по максимум один </Text>
<Object index="1" />
<Text> который </Text>
<Role index="1" />
<Text> этот</Text>
<Object index="0" />
</Constraint>

<Constraint xsi:type="External Uniqueness" >
<Text>Каждая комбинация </Text>
<Object index="1" />
<Loop index="1">
<Text>и</Text>
<Object index="n" />
</Loop>
<Text> относится только к одному </Text>
<Object index="0" />
</Constraint>

<Constraint xsi:type="Many Uniqueness" >
<Text>возможно, что</Text>
<Object index="0" />
<Role index="0"></Role>
<Text> больше, чем один </Text>
<Object index="1" />
<Text> и, что</Text>
<Object index="1" />
<Role index="1"></Role>
<Text> больше, чем один </Text>
<Object index="0" />
</Constraint>

<Constraint xsi:type="Subtype" >
<Text>Каждый</Text>
<Object index="child" />
<Text> также является </Text>
<Object index="parent" />
</Constraint>

<Constraint xsi:type="Exclusive" >
<Text>Каждый</Text>
<Object index="0"/>
<Text> может быть </Text>
<Object index="1"/>
```

```
<Loop index="1">
  <Text>или</Text>
  <Object index="n"/>
</Loop>
</Constraint>

<Constraint xsi:type="Total" >
  <Text>Каждый</Text>
  <Object index="0" />
  <Text> является либо </Text>
  <Object index="1" />
  <Loop index="1" >
    <Text> или </Text>
    <Object index="n" />
  </Loop>
</Constraint>

<Constraint xsi:type="Partition" >
  <Text>Each </Text>
  <Object index="0" />
  <Text> по крайней мере является одним из </Text>
  <Object index="1" />
  <Loop index="1" >
    <Text> или </Text>
    <Object index="n" />
  </Loop>
  <Text>но не всеми сразу</Text>
</Constraint>

<Constraint xsi:type="Subset" >
  <Text>Если </Text>
  <Object index="0" />
  <Role index="child" />
  <Object index="child" />
  <Text>, то </Text>
  <Object index="0" />
  <Role index="parent" />
  <Object index="parent" />
</Constraint>

<Constraint xsi:type="Equality" >
  <Object index="0" />
  <Role index="first" />
  <Object index="first" />
  <Text>если и только если</Text>
  <Text>этот </Text>
  <Object index="0" />
  <Role index="second" />
  <Object index="second" />
  <Text>, и наоборот</Text>
</Constraint>

<Constraint xsi:type="Equality FactType" >
  <Object index="0" />
  <Role index="First" />
  <Object index="First" />
  <Text>если и только если</Text>
  <Text>этот </Text>
```

```
<Object index="1" />
<Role index="Second" />
<Object index="Second" />
</Constraint>

<Constraint xsi:type="Subset FactType" >
<Text>Если </Text>
<Object index="0" />
<Role index="child" />
<Object index="child" />
<Text>, то этот </Text>
<Object index="1" />
<Role index="parent" />
<Text> тот </Text>
<Object index="parent" />
</Constraint>

<Constraint xsi:type="Exclusion" >
<Text>Не существует </Text>
<Object index="0" />
<Text>, который </Text>
<Role index="first" />
<Object index="first" />
<Text> и </Text>
<Role index="second" />
<Object index="second" />
</Constraint>

<Constraint xsi:type="Exclusion FactType" >
<Text>Не существует </Text>
<Object index="0" />
<Text>, который </Text>
<Role index="first" />
<Object index="first" />
<Text>и</Text>
<Role index="second" />
<Text>тот </Text>
<Object index="second" />
</Constraint>

<Constraint xsi:type="Reflexive" >
<Text>Каждый</Text>
<Object index="0"/>
<Role index="0"/>
<Text> </Text>
</Constraint>

<Constraint xsi:type="Irreflexive" >
<Text>No</Text>
<Object index="0"/>
<Role index="0"/>
<Text> самого себя</Text>
</Constraint>

<Constraint xsi:type="Symmetric" >
<Text>Если</Text>
<Object index="0"/>
<Text> x</Text>
```

```

<Role index="0"/>
<Object index="0"/>
<Text> у</Text>
<Text> то наоборот</Text>
</Constraint>

<Constraint xsi:type="Frequency">
<Text> -[Frequency] Если</Text>
<Object index="0"/>
<Role index="0"/>
<Object index="1"/>
<Role index="0"/>
<Text> , тогда этот </Text>
<Object index="0"/>
<Role index="0"/>
<Text> минимум </Text>
<Minimum/>
<Text> и максимум </Text>
<Maximum/>
<Role index="0"/>
<Text>(ы)</Text>
</Constraint>

<Constraint xsi:type="Asymmetric">
<Text> -[Asymmetric] Если</Text>
<Object index="0"/>
<Text>X</Text>
<Role index="0"/>
<Object index="0"/>
<Text> Y, то не может быть наоборот</Text>
</Constraint>

<Constraint xsi:type="Acyclic">
<Text> -[Acyclic] </Text>
<Object index="0"/>
<Text> не может быть прямо (или косвенно через связи)</Text>
<Role index="0"/>
<Text> самого себя</Text>
</Constraint>

<Constraint xsi:type="Transitive">
<Text> -[Intransitive] Если</Text>
<Object index="0"/>
<Text>X</Text>
<Role index="0"/>
<Object index="0"/>
<Text>Y, и Y</Text>
<Role index="0"/>
<Text> Z, то невозможно что 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 edn. 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.