

# Lietuviškas verbalizacijos šablonas

## skirtas ORM konceptiniams modeliams ir taisyklėms

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

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

Juozas Gordevičius, Free University of Bozen-Bolzano, Italy.

---

Techninė ataskaita straipsniui<sup>1</sup>: Jarrar, M., Keet, C.M., Dongilli, P. **Daugiakalbė verbalizacija ORM konceptiniams modeliams ir aksiomatizuotoms ontologijoms.** [Perduota spausdinti].

URL: <http://www.starlab.vub.ac.be/staff/mustafa/orm/verbalization/>

---

**Santrauka.** Aukščiau paminėtame straipsnyje mes aprašome naują būdą panaudoti *daugiakalbę* logikos teorijų, aksiomatizacijų ir kitų specifikacijų, pvz. verslo taisyklių, verbalizaciją. Ši sprendimą demonstruojame Objektų Rolių Modeliavimo kalbos pagalba, tačiau norint pagelbėti srities ekspertams, pagrindinius principus galima taikyti ir kitiems modeliams ar formalioms kalboms, pvz. Aprašomajai logikai. Taip pat mūsų sprendimas pasižymi lankstumu, plėtos galimybėmis, paprastai tvarkomais verbalizacijos šablonais. Tuo būdu modeliui paprasta pritaikyti daugiau nei šiuo metu siūlomų 11 kalbų.

Ši ataskaita aprašo *lietuvišką verbalizacijos šabloną*. Turint ORM schemą (arba ORM-ML bylą) ir verbalizacijos šabloną, lietuviška taisyklių ir faktų tipų verbalizacija generuojama automatiškai. Ataskaitoje pateikiamas išsamus ORM schemas ir atitinkamai sugeneruotos verbalizacijos pavyzdys.

## 1 Įvadas

Aukščiau paminėtame straipsnyje mes aprašome naują būdą panaudoti daugiakalbę logikos teorijų, formalių aksiomatizacijų ir kitų specifikacijų, pvz. verslo taisyklių, ontologijų ir kt., verbalizaciją. Siekdami pademonstruoti sprendimą, pateikiame lankstų ir plečiamą verbalizacijos šabloną, skirtą Objektų Rolių Modeliavimo kalbai. Šabloną lengva modifikuoti ir išversti į žmonių kalbas. Ši ataskaita aprašo lietuvišką ORM modelių ir taisyklių verbalizaciją. Kitų kalbų (vokiečių, italų, arabų, rusų, danų, ispanų, prancūzų ir kt.) verbalizacijų aprašymai prieinami aukščiau nurodytu Interneto adresu.

Pagrindiniai mūsų sprendimo principai gali būti panaudoti ir kitiems konceptiniams modeliams bei formalioms kalboms, pavyzdžiui Aprašomajai logikai. Mūsų tikslas aprašyti šabloną duotai taisyklių aibei, modeliams arba aksiomoms, kurio išeiiga būtų fiksuotos sintaksės pseudo-naturalios kalbos sakiniai. Pavyzdžiui, formali taisyklė

$$\forall x (Knyga(x) \rightarrow \exists y (ISBN(y) \wedge Turi(x, y)))$$

gali būti išversta į

**Kiekviena Knyga privalo turėti ISBN.**

Tokiu būdu mes suteikiame galimybę sričių ekspertams patiems aprašyti ir patikrinti formalias savo srities specifikacijas. Jie neturi žinoti, kad šie sakiniai yra formalios aksiomos, t.y. vartotojui nereikia žinoti apie giliau slypinčius loginius formalumus. Mūsų sprendimą kartu su pateikiamais šablonais galima panaudoti verslo taisyklių, ontologijų, žinių bazių ir kt. modeliavimui. Straipsnyje [H04] pateikiamas panašus ORM verslo taisyklių verbalizacijos sprendimas.

---

<sup>1</sup> Citavimui naudokite: Jarrar, M., Keet, C.M., Gordevičius, J.: *A Lithuanian 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].*

Sekančiame skyriuje pateikiame ORM pavyzdį kartu su visų modelyje esančių taisyklių verbalizacija. Ši verbalizacija generuota *automatiškai* pagal lietuvišką verbalizacijos šabloną, pateiktą 3 skyriuje. Tai pat, šis sprendimas pilnai realizuotas ir palaikomas DogmaModeler ontologijų modeliavimo įrankiui [J05]. Verta pastebėti, kad DogmaModeler automatiška verbalizacija buvo panaudota dešimčių teisininkų kuriant Vartotojų nusiskundimų ontologiją [J05][JVM03].

**Pastaba apie modališkumą:** Mūsų verbalizacijos šablonas gali būti pritaikytas pagal aplikacijos/pagrindimo scenarijų, t.y. priklausomai nuo to, ar modeliuojami integralumo apribojimai, išvedimo taisyklės, verslo taisyklės ar kita. Pavyzdžiui, aukščiau pateikta taisyklė gali būti verbalizuota keliais skirtingais būdais: 1) Kiekviena **Knygaprivalo Turi** bent vieną **ISBN**. 2) Kiekviena **Knyga Turi** kokį nors **ISBN**. 3) Jeigu yra **Knyga**, tai ji **Turi ISBN**. 4) **Knyga**, kuri **Neturi ISBN**, neleidžiama. 5) Jei **Knyga Neturi ISBN**, tai...

## 2 ORM schemas pavyzdys

Vienoje diagramoje pavaizduojame daugumą ORM palaikomų taisyklių tipų. Mūsų straipsnis [JKD06] aprašo technines detales, kaip realizuotas verbalizacijos sprendimas. Žūrėkite [H01] norėdami išsiaiškinti daugiau apie ORM ir [J05] apie DogmaModeler įrankį, kurį mes naudojame automatiniam ORM modelių ir verbalizacijų konstravimui.

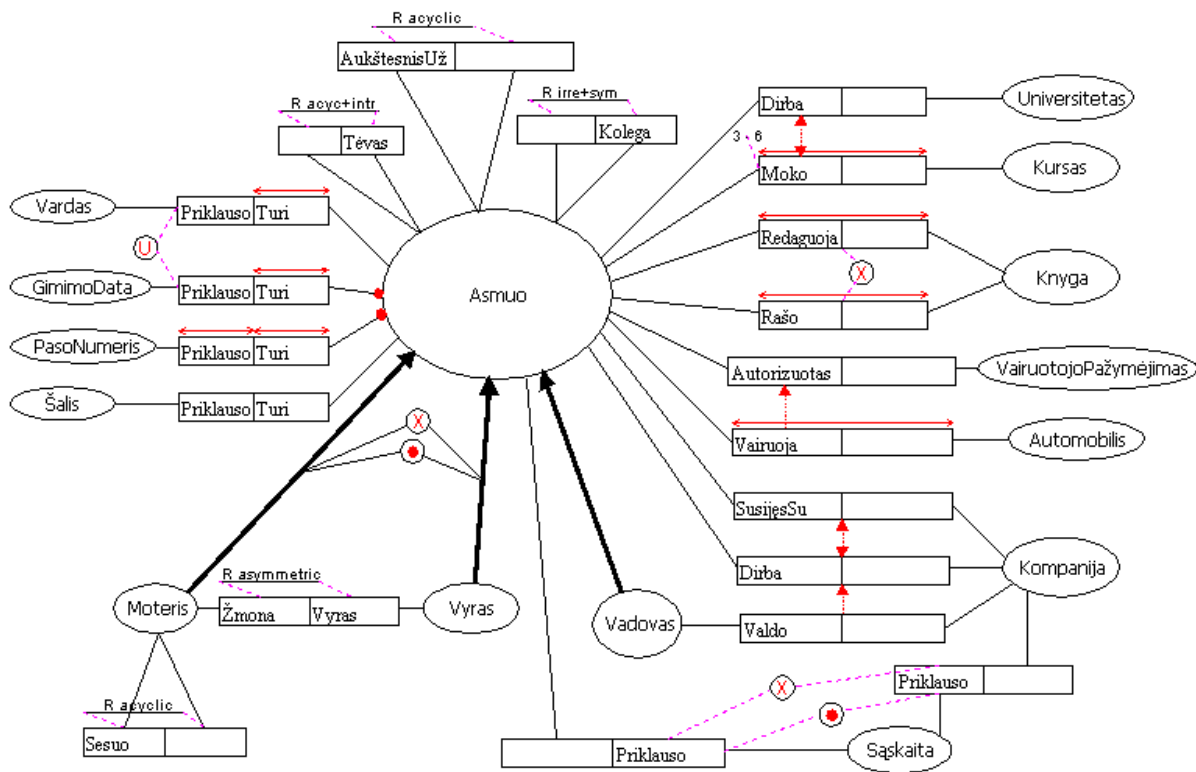


Fig. 1. Example of ORM rules, in Lithuanian.

Apribojimai ir taisyklės aukščiau pateiktame ORM pavyzdyje verbalizuojami automatiškai pagal verbalizacijos šabloną, pateiktą 3 skyriuje.

- [Mandatory] Kiekvienas Asmuo privalo Turi bent vieną PasoNumeris.
- [Mandatory] Kiekvienas Asmuo privalo Turi bent vieną GimimoData.
- [Mandatory] Kiekvienas Sąskaita turėtų būti Asmuo arba Priklauso Kompanija Priklauso.
- [Uniqueness] Kiekvienas Asmuo privalo Turi daugiausia vieną GimimoData.
- [Uniqueness] Kiekvienas Asmuo privalo Turi daugiausia vieną Vardas.
- [Uniqueness] Kiekvienas Asmuo privalo Turi daugiausia vieną PasoNumeris.
- [Uniqueness] Kiekvienas PasoNumeris privalo Priklauso daugiausia vieną Asmuo.
- [Uniqueness] įmanoma, kad Asmuo Moko daugiau nei vieną Kursas ir atvirkščiai.
- [Uniqueness] įmanoma, kad Asmuo Redaguoja daugiau nei vieną Knyga ir atvirkščiai.
- [Uniqueness] įmanoma, kad Asmuo Rašo daugiau nei vieną Knyga ir atvirkščiai.
- [Uniqueness] įmanoma, kad Asmuo Vairuoja daugiau nei vieną Automobilis ir atvirkščiai.
- [Uniqueness] { GimimoData ir Vardas } turi atitikti daugiausia vieną Asmuo.
- [Exclusive] Kiekvienas Asmuo turėtų būti arba Vyras , arba Moteris.
- [Totality] Kiekvienas Asmuo privalo būti mažiausiai Vyras , arba Moteris.
- [Subset] Jei Asmuo Vairuoja Automobilis , tai Asmuo Autorizuotas VairuotojoPažymėjimas.
- [Subset] Jei Vadovas Valdo Kompanija , tai šis Asmuo Dirba tam/tai Kompanija.
- [Equality] Asmuo Dirba Universitetas tada ir tik tada, kai Asmuo Moko Kursas.
- [Equality] Asmuo SusijęsSu Kompanija tada ir tik tada, kai šis Asmuo Dirba tam/tai Kompanija.
- [Exclusion] Nei vienas Sąskaita ne- Priklauso Asmuo , kai Priklauso Kompanija.
- [Exclusion] Niekada Asmuo , kuris Redaguoja Knyga , tos Knyga ne- Rašo.
- [Value] Galimos Šalis reikšmės yra: { Belgija , Prancūzija, Vokietija }.
- [Irreflexive] Joks Asmuo nėra Kolega pats sau.
- [Symmetric] Jei Asmuo X Kolega Asmuo Y, tai ir atvirkščiai.
- [Acyclic] Asmuo negali būti tiesiogiai (arba netiesiogiai) AukštesnisUž pats sau .
- [Acyclic] Moteris negali būti tiesiogiai (arba netiesiogiai) Sesuo pats sau .
- [Asymmetric] Jei Moteris X Žmona Moteris Y, tai ne atvirkščiai .
- [Intransitive] Jei Asmuo X Tėvas Asmuo Y, ir Y Tėvas Z, tai negali bti, kad X Tėvas Z.
- [Frequency] Jei Asmuo Moko Kursas, tai šis Asmuo Moko mažiausiai 3 ir daugiausiai 6 Kursas.

### 3 Lietuviškas verbalizacijos šablonas

Šablonas pateikiamas XML formatu, kuris realizuotas DogmaModeler įrankyje siekiant pritaikyti lietuvišką verbalizaciją ORM modeliams. Daugiau detalių apie šį sprendimą pateikta [JKD06]. Taip pat skaitykite [J05] apie 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/verbaliz
ation/'>

<ORMNLMeta>
  <Meta name="DC.Title" content="Lithuanian verbalization template (Ver0.1)"/>
  <Meta name="DC.Version" content="0.1"/>
  <Meta name="DC.Creator" content="Mustafa Jarrar"/>
  <Meta name="DC.Contributor" content="Juozas Gordevicius"/>
  <Meta name="DC.Language" content="Lithuanian"/>
</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] Kiekvienas</Text>
  <Object index="0"/>
  <Text>privalo</Text>
  <Role index="0"/>
  <Text>bent viena</Text>
  <Object index="1"/>
</Constraint>

<Constraint xsi:type="Backward Mandatory">
  <Text> -[Mandatory] Kiekvienam</Text>
  <Object index="0"/>
  <Text>yra bent vienas</Text>
  <Object index="1"/>
  <Text>, kuris</Text>
  <Role index="1"/>
  <Text>tam</Text>
  <Object index="0"/>
</Constraint>

<Constraint xsi:type="Disjunctive Mandatory">
  <Text> -[Mandatory] Kiekvienas</Text>
  <Object index="0"/>
  <Text>turėtų būti</Text>
  <Object index="1"/>
  <Loop index="1" >
    <Text>arba</Text>
    <Role index="n"/>
    <Object index="n"/>
  </Loop>
  <Role index="0"/>
</Constraint>

<Constraint xsi:type="Uniqueness">
  <Text> -[Uniqueness] Kiekvienas</Text>
  <Object index="0"/>
  <Text>privalo</Text>
  <Role index="0"/>
  <Text>daugiausia viena</Text>
  <Object index="1"/>
</Constraint>

<Constraint xsi:type="Backward Uniqueness">
  <Text> -[Uniqueness] Kiekvienas</Text>
  <Object index="0"/>
  <Text>turi turėti daugiausia viena</Text>
  <Object index="1"/>
  <Text>kuris</Text>
  <Role index="1"/>
  <Text>tam</Text>
  <Object index="0"/>
</Constraint>

<Constraint xsi:type="Many Uniqueness">
  <Text> -[Uniqueness] įmanoma, kad </Text>
  <Object index="0"/>
<Role index="0"></Role>
  <Text>daugiau nei viena</Text>
  <Object index="1"/>
  <Text> ir atvirkščiai</Text>

```

```

</Constraint>

<Constraint xsi:type="External Uniqueness">
  <Text> -[Uniqueness] {</Text>
  <Object index="1"/>
  <Loop index="1">
    <Text>ir</Text>
    <Object index="n"/>
  </Loop>
  <Text>} turi atitikti daugiausia vieną</Text>
  <Object index="0"/>
</Constraint>

<Constraint xsi:type="Subtype">
  <Text> -[Subtype] Kiekvienas</Text>
  <Object index="child"/>
  <Text>taip pat yra</Text>
  <Object index="parent"/>
</Constraint>

<Constraint xsi:type="Value">
  <Text> -[Value] Galimos </Text>
  <Object index="0"/>
  <Text> reikšmės yra: {</Text>
  <Value index="0"/>
  <Loop index="1">
    <Text>,</Text>
    <Value index="n"/>
  </Loop>
  <Text> }</Text>
</Constraint>

<Constraint xsi:type="Exclusive">
  <Text> -[Exclusive] Kiekvienas</Text>
  <Object index="0"/>
  <Text>turėtų būti arba</Text>
  <Object index="1"/>
  <Loop index="1">
    <Text>,</Text> arba</Text>
    <Object index="n"/>
  </Loop>
</Constraint>

<Constraint xsi:type="Total">
  <Text> -[Total] Kiekvienas</Text>
  <Object index="0"/>
  <Text>privalo būti mažiausiai</Text>
  <Object index="1"/>
  <Loop index="1">
    <Text>,</Text> arba</Text>
    <Object index="n"/>
  </Loop>
</Constraint>

<Constraint xsi:type="Partition">
  <Text> -[Partition] Kiekvienas</Text>
  <Object index="0"/>
  <Text>yra bent vienas iš</Text>
  <Object index="1"/>
  <Loop index="1">
    <Text>,</Text> arba</Text>
    <Object index="n"/>
  </Loop>

```

```

    <Text>, bet ne visi kartu</Text>
  </Constraint>

  <Constraint xsi:type="Subset">
    <Text> -[Subset] Jei</Text>
    <Object index="0"/>
    <Role index="child"/>
    <Object index="child"/>
    <Text>, tai</Text>
    <Object index="0"/>
    <Role index="parent"/>
    <Object index="parent"/>
  </Constraint>

  <Constraint xsi:type="Subset FactType">
    <Text> -[Subset] Jei</Text>
    <Object index="0"/>
    <Role index="child"/>
    <Object index="child"/>
    <Text>, tai šis</Text>
    <Object index="1" />
    <Role index="parent"/>
    <Text>tam/tai</Text>
    <Object index="parent"/>
  </Constraint>

  <Constraint xsi:type="Equality">
    <Text> -[Equality] </Text>
    <Object index="0"/>
    <Role index="first"/>
    <Text> </Text>
    <Object index="first"/>
    <Text>tada ir tik tada, kai </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>tada ir tik tada, kai</Text>
    <Text>šis</Text>
    <Object index="1"/>
    <Role index="Second"/>
    <Text>tam/tai</Text>
    <Object index="Second"/>
  </Constraint>

  <Constraint xsi:type="Exclusion">
    <Text> -[Exclusion] Nei vienas </Text>
    <Object index="0"/>
    <Text>ne-</Text>
    <Role index="first"/>
    <Text> </Text>
    <Object index="first"/>
    <Text>, kai</Text>
    <Role index="second"/>
    <Text> </Text>
  </Constraint>

```

```

    <Object index="second"/>
  </Constraint>

<Constraint xsi:type="Exclusion FactType">
  <Text> -[Exclusion] Niekada</Text>
  <Object index="0"/>
  <Text>, kuris</text>
  <Role index="first"/>
  <Object index="first"/>
  <Text>, tos</Text>
  <Object index="second"/>
  <Text>ne-</Text>
  <Role index="second"/>
</Constraint>

<Constraint xsi:type="Frequency">
  <Text> -[Frequency] Jei </Text>
  <Object index="0"/>
  <Role index="0"/>
  <Object index="1"/>
  <Text>, tai šis </Text>
  <Object index="0"/>
  <Role index="0"/>
  <Text>mažiausiai </Text>
  <Minimum/>
  <Object index="1"/>
  <Text> ir daugiausiai </Text>
  <Maximum/>
  <Object index="1"/>
</Constraint>

<Constraint xsi:type="Irreflexive">
  <Text> -[Irreflexive] Joks</Text>
  <Object index="0"/>
  <Text>nėra</Text>
  <Role index="0"/>
  <Text> pats sau</Text>
</Constraint>

<Constraint xsi:type="Symmetric">
  <Text> -[Symmetric] Jei</Text>
  <Object index="0"/>
  <Text>X</Text>
  <Role index="0"/>
  <Object index="0"/>
  <Text>Y</Text>
  <Text>, tai ir atvirkščiai</Text>
</Constraint>

<Constraint xsi:type="Asymmetric">
  <Text> -[Asymmetric] Jei</Text>
  <Object index="0"/>
  <Text> X</Text>
  <Role index="0"/>
  <Text> </Text>
  <Object index="0"/>
  <Text> Y, tai ne atvirkščiai</Text>
</Constraint>

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

```

```

<Text> negali būti tiesiogiai (arba netiesiogiai)</Text>
<Role index="0"/>
<Text> pats sau</Text>
</Constraint>

<Constraint xsi:type="Transitive">
<Text> -[Intransitive] Jei</Text>
<Object index="0"/>
<Text>X</Text>
<Role index="0"/>
<Object index="0"/>
<Text>Y, ir Y</Text>
<Role index="0"/>
<Text> Z, tai negali būti, kad X</Text>
<Role index="0"/>
<Text>Z</Text>
</Constraint>

</ORMNLBody>
</ORMSchema>

```

## Padėka

Mes skolingi Andriy Lisovoy, kuris padėjo realizuoti DogmaModeler, ir Hai Nguyen Hoang, savo magistro studijų metu padėjusiam dirbant prie pirmosios verbalizacijos komponento verbalizacijos. Šis darbas dalinai finansuotas EU Knowledge Web NoE projekto (IST-2004-507482).

## Nuorodos

- [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.