Date: August 2016

Semantics of Business Vocabulary and Business Rules (SBVR), v1.4

Annex J - The ORM Notation for Verbalizing Facts and Business Rules

OMG Document Number: dtc/2016-08-29

Note: This SBVR Annex is published as a separate document solely for convenience and ease of use. The fact that it is published as a separate SBVR specification document makes no change to its status as part of the SBVR specification, or the way in which it can be updated under OMG Policies and Procedures.

Annex J - The ORM Notation for Verbalizing Facts and Business Rules

(informative)

J.1 General

Annexes C and F discussed how to verbalize facts and business rules in the SBVR Structured English and RuleSpeak notations. This annex briefly presents a third approach to verbalization that is based on *Object-Role Modeling* (ORM) [Halp1998, Halp2001], a conceptual modeling approach that has been used productively in industry for over 30 years. While this approach has been localized to other languages (including Japanese, German, and French), we restrict ourselves here to the English version.

Business rules may be specified in ORM using graphical and/or textual languages. We confine ourselves here to just part of ORM's textual language. We regard a *static business rule* to be a constraint or derivation rule that applies to each individual state of the business, taken one state at a time. This annex focuses on the *verbalization* of static business rules, ignoring dynamic rules relating to state transitions or workflows. In the interests of brevity, only a few of ORM's rule verbalization patterns are illustrated here, mainly using examples from the EU-Rent case study. A detailed discussion may be found in the references [Halp2003a, Halp2003b, Halp2003c, Halp2003d, Halp2004c, Halp2004d, Halp2004e, Halp2004f, Halp2004g, Halp2004b].

J.2 Criteria for Business Rule Verbalization in ORM

Static business rules are best applied to a fact model that identifies the fact types of interest to the business. Table J.1 shows some fact types with arities from 1 to 4. Each fact type role corresponds to an object placeholder (depicted here as an ellipsis "…") in the predicate. Here predicates are displayed in *mixfix* notation, allowing object terms to be placed in a sentence at any position. Higher arity predicates (quinary, etc.) are also possible.

Table J.1 Examples of fact types of different arity

Fact Type	Predicate	Arity
Person smokes	… smokes	1 (Unary)
Person was born in Country	… was born in …	2 (Binary)
Person played Sport for Country	… played … for …	3 (Ternary)
Person introduced Person to Person on Date	… introduced … to … on …	4 (Quaternary)

The ORM textual language for verbalizing fact instances, fact types, and business rules is based on the following criteria:

- *expressibility* - the language is able to express a wide range of business rules
- *clarity* - the rules are understandable by non-technical domain experts
- *flexibility* - the language directly supports predicates of any arity
- *localizability* - the language constructs are expressible in different native languages
- *formality* - the rules are unambiguous, and should ideally be executable

Apart from its graphical language, ORM uses a textual language that is both formal and conceptual, so that it can serve for communication and validation with domain experts, as well as being executable. Relevant dimensions used in ORM for rule verbalization are listed in Table J.2, along with the choices available. For detailed discussion of these criteria, see the references.

Table J.2 Classification schemes for rule verbalization

Dimension	Choices
Form	Positive Negative Default
Modality	Alethic Deontic
Style	Relational Attribute Mixed
Context	Local Global
Formality	Informal Semiformal Formal

ORM's verbalization language applies to mixfix predicates of any arity, with predefined patterns to cater for a very wide range of constraints found in business domains. Unlike some other approaches, ORM leaves the verbalization of the underlying fact model unchanged (e.g., no need to pluralize noun phrases and related verb phrases).

Every constraint has an associated *modality*, determined by the logical modal operator that functions explicitly or implicitly as its main operator. In practice, the modality is typically either *alethic* or *deontic* (see Table J.3). Logical negation may be used to obtain the usual equivalences (e.g., not necessary ≡ possible, not obligatory ≡ permitted, not permitted ≡ forbidden).

Table J.3 - Alethic and deontic modal operators

Alethic	Deontic
It is necessary that	It is obligatory that
It is possible that	It is permitted that
It is impossible that	It is forbidden that

The next two sub clauses present some simple examples. Far more complex examples may be found in the references.

J.3 Some Basic Rule Examples in ORM

Simple uniqueness constraint

Positive form:

Each rental car *is owned by* at most one branch.

In positive verbalizations, the modality is often assumed (as above), but may be explicitly prepended ("It is obligatory that" for deontic modality; "It is necessary that" for alethic modality).

Negative form, deontic modality:

> It is forbidden that the same rental car *is owned by* more than one branch.

Negative form, alethic modality:

> It is impossible that the same rental car *is owned by* more than one branch.

Composite uniqueness constraint:

Positive, deontic form of a uniqueness constraint over two fact type roles from the ternary fact type room *at* hour slot *is booked for* course.

> It is obligatory that
>
> given any room and hour slot
>
> that room *at* that hour slot *is booked for* at most one course

Composite Exclusion constraint*:*

Relational style: No person *directed* and *reviewed* the same movie.

Attribute style: For each movie:

no director is a reviewer.

Join Subset constraint:

> Each advisor who *serves in* a country
>
> also *speaks* a language that *is used by* that country.

Derivation Rule:

Relational style: Define person$_1$ *is an uncle of* person$_2$ as

person$_1$ *is a brother of* person$_3$ who *is a parent of* person$_2$

Attribute style: For each person: uncle = brother of parent.

J.4 Some EU-Rent Rule Examples

It is obligatory that each rental *has* a car group.

It is obligatory that each rental car that *has* a service reading greater than 5000 miles *is scheduled for service.*

It is obligatory that
if a rental car *is in* an international return that *is to* a receiving branch
that *is in* a local area that *is in* a country
then that rental car *is registered in* that country.

It is permitted that each renter *books* more than one rental.

J.5 EU-Rent Examples in ORM

This sub clause provides restatements in ORM of the EU-Rent examples presented in SBVR Structured English (Annex G.1.4) and in RuleSpeak (Annex H). The ORM rewording is displayed after the SBVR Structured English formulation, assuming that the fact types used in the ORM verbalization are defined in the model.

Conventions used

- Object types are bold and underlined.
- Verb phrases are bold.
- Components of constraints are in italics.
- Articles and referents are unadorned.
- The terms "may" and "must" indicate deontic modalities permission and obligation, respectively.
- The term "might" (as in #9) indicates alethic possibility; lack of any modal term (as in #1) indicates alethic necessity.
- The term "which" is used to provide proper English syntax to avoid ending with a preposition; the preposition immediately preceding "which" actually terminates a verb phrase in the model.

1	It is necessary that each rental *has exactly one* requested car group.
	Each **rental** **requests** *at least one* **car group**. *Each* **rental** **requests** *at most one* **car group**. - or, combined: *Each* **rental** **requests** *exactly one* **car group**.

Guidance Type: structural business rule

2	It is obligatory that the rental duration of *each* rental *is at most* *90 rental days*.
	It must be that each **rental** **lasts** *at most 90* **rental days**.

Guidance Type: operative business rule

3	It is obligatory that each driver of a rental *is qualified.*
	It must be that each **driver that drives** a **rental is qualified at** the **date/time at** which that **rental actually started**.

Guidance Type: operative business rule

4	It is obligatory that the rental *incurs* a location penalty charge if the drop-off location *of* a rental *is not* the EU-Rent site that *is base for* the return branch *of* the rental.
	It must be that a **rental incurs** a **location penalty charge** *if* the **rented car of that rental is dropped off at a location that is *different* from the EU-Rent site where the return branch of that rental is based.**

Guidance Type: operative business rule

Note not expressible using standard ORM constraint notation

5	It is obligatory that the rental charge *of* a rental *is calculated in* the business currency *of* the rental.
	It must be that a **rental charge** that **is incurred by** a **rental is calculated in a business currency that is used by that rental.**

Guidance Type: operative business rule

6	It is permitted that a rental *is open* only if an estimated rental charge *is provisionally charged to* a credit card *of* the renter that *is responsible for* the rental.
	It may be that a **rental is open** *only if* **an estimated rental charge** that **is incurred by** that **rental is provisionally charged to** a **credit card** that **is held by the customer** that **acquires** that **rental.**

Guidance Type: operative business rule

Note: not expressible using standard ORM constraint notation.

7	It is obligatory that the local area that *includes* the return branch *of* an in-country rental or international inward rental *owns* the rented car *of* the rental *at* the actual return date/time *of* the rental.
	It must be that the **local area** that **includes** the **return branch** that **is** the **destination of an in-country rental** *or* an **international inward rental owns** the **rental car** that **is assigned to** that **rental at** the **date/time at** which that **rental is returned**.

Guidance Type: operative business rule

Note: not expressible using standard ORM constraint notation.

8	It is obligatory that *at* the actual pick-up date/time *of* each rental the fuel level *of* the rented car *of* the rental *is* full.
	It must be that the **rental car** that **is assigned** to a **rental** **has** a **fuel level** *equal* **to** '*full*' **at** the **date/time** **at** which that **rental** **actually started.**

Guidance Type: operative business rule

Note: not expressible using standard ORM constraint notation.

9	It is possible that the notification date/time *of* a bad experience that *occurs during* a rental *is after* the actual return date/time *of* the rental.
	It might be that the **notification of** a **bad experience** that **occurs during** a **rental** **is received at** a **date/time** that *is greater than* the **date/time** **at which** that **rental** **is actually returned**.

Guidance Type: advice of possibility

Note: not expressible using standard ORM constraint notation; however, possibilities are implied by the absence of other constraints - especially necessities - that preclude them.

10	It is permitted that the drop-off branch *of* a rental *is* not the return branch *of* the rental.
	It may be that a **rental** **is dropped off at** a **different** **branch** than the **branch** **to which** that **rental** **is to be returned**.

Guidance Type: advice of permission

Note: implied by the model, as is (no equality constraint is specified, therefore it is permitted).