

Verbalizing Business Rules: Part 10

Terry Halpin
Northface University

Business rules should be validated by business domain experts, and hence specified using concepts and languages easily understood by business people. This is the tenth in a series of articles on expressing business rules formally in a high-level, textual language. The first article [3] discussed criteria for a business rules language, and verbalization of simple uniqueness and mandatory constraints on binary associations. Article two [4] examined hyphen-binding, and verbalization of internal uniqueness constraints that span a whole association, or that apply to n -ary associations. Article three [5] covered verbalization of basic external uniqueness constraints. Article four [6] considered relational-style verbalization of external uniqueness constraints involving nesting or long join paths, as well as attribute-style verbalization of uniqueness constraints and simple mandatory constraints. Article five [7] discussed verbalization of mandatory constraints on roles of n -ary associations, and disjunctive mandatory constraints (also known as inclusive-or constraints) over sets of roles. Article six [8] considered verbalization of value constraints. Article seven [9] examined verbalization of subset constraints. Article eight [10] discussed verbalization of equality constraints. Article nine [11] covered verbalization of exclusion constraints. This tenth article deals with verbalization of internal frequency constraints on single roles.

Verbalization of internal frequency constraints on single roles

Consider the output report displayed in Table 1, which lists details about musical duets (i.e. pairs of musicians who perform together). The Everly Brothers is an actual duet, but the other duet examples are invented for discussion purposes. In this business domain, musicians are identified simply by their name, so the Don Everly appearing in two duets is the same person. The third column indicates whether or not the duet is currently popular. Assume closed world semantics for popularity (i.e. for each duet we do know its popularity). The “?” mark indicates missing information—we know the name and popularity of the Northface Duo, but its members are not yet decided. For each duet in this business domain, our knowledge about its membership is either complete (we know both the members) or totally incomplete (we know neither member). In other words, we cannot know that one person is a member of a duet without knowing who the other duet member is.

Table 1 Details about some musical duets.

Duet	Musicians	Popular?
Everly Brothers	Don Everly, Phil Everly	Yes
Two Dons	Don Everly, Donovan Leitch	Yes
Northface Duo	?	No

Figure 1 shows one way to schematize this report in Object-Role Modeling (ORM). A duet’s popularity is modeled using the unary fact type Duet is popular. A duet’s membership is modeled using the binary fact type Duet includes Musician (inverse reading: Musician is a member of Duet). The arrow-tipped bar on the binary fact type denotes a spanning uniqueness constraint, indicating that the relationship is many-to-many (the same duet may include more than one musician, and the same musician may be a member of more than one duet), and that each membership fact is recorded just once (for each state of the business). The “2” besides Duet’s role in the membership fact type indicates the frequency (number of occurrences) for that role, for each duet that plays that role. In ORM this is called a *frequency constraint*. Verbalizing this kind of constraint is the focus of this article.

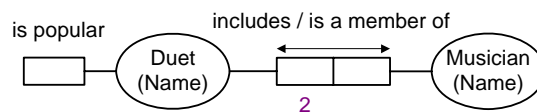


Figure 1 An ORM schema for Table 1.

Adding a frequency constraint of 2 to Duet’s role in the membership predicate means that, for each state of the business domain, each duet that plays that role does so 2 times. This constraint is perhaps best understood by populating the schema with the fact instances from the original report, as shown in Figure 2. In ORM, an object type’s instances may be displayed in a table, and an elementary fact type’s instances may be displayed in a table, with a column for each role.

The table beneath the Duet entity type lists three duets, which is equivalent to stating three existential facts: there is a Duet that has DuetName ‘Everly Brothers’; there is a Duet that has DuetName ‘Two Dons’; there is a Duet that has DuetName ‘Northface Duo’. The table below the unary fact type lists two elementary facts: Duet ‘Everly Brothers’ is popular; Duet ‘Two Dons’ is popular. Given the closed world assumption, the absence of Northface Duo in this table indicates the negated fact that Northface Duo is not popular. The table below the binary fact type lists four elementary facts: Duet ‘Everly Brothers’ includes Musician ‘Don Everly’; Duet ‘Everly Brothers’ includes Musician ‘Phil Everly’; Duet ‘Two Dons’ includes Musician ‘Don Everly’; Duet ‘Two Dons’ includes Musician ‘Donovan Leitch’.

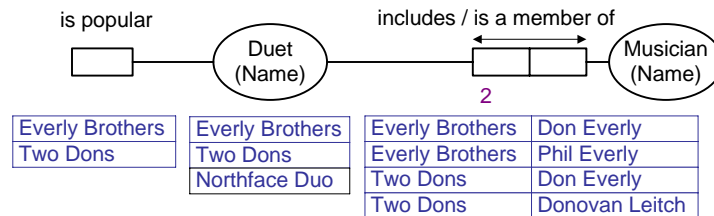


Figure 2 Populating the previous ORM schema with fact instances.

Notice that each entry in the fact column for Duet’s role in the membership fact type occurs there exactly twice, in accordance with the frequency constraint of 2 on that role. This frequency constraint may be formally *verbalized* in relational style in either of the following ways:

- Each Duet includes zero or two instances of Musician.
- Each Duet that includes a Musician includes two instances of Musician.

If automated support for pluralization is available, more natural readings are obtained by replacing “instances of Musician” by “Musicians” in the above verbalizations. In this series of articles, we do not assume that such support is available.

The first verbalization above is especially appropriate if the schema is presented in UML notation, as shown in Figure 3(b). In UML, the frequency constraint appears as a compound multiplicity constraint “0, 2” (zero or two) next to the role played by Musician. The “{P}” annotation is a non-standard extension to UML to express the preferred identifier (and hence its uniqueness constraint). UML does not directly support unary fact types, so popularity is modeled as the Boolean attribute “isPopular”. For binary associations, the relevant UML multiplicity constraint applies to the role opposite the role with the ORM frequency constraint.

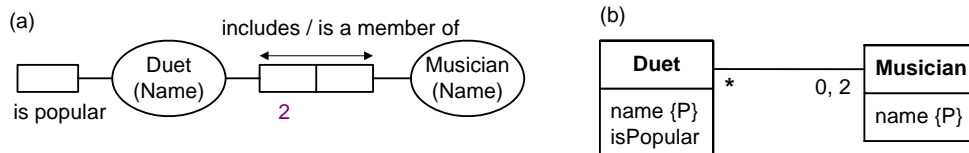


Figure 3 (a) The previous ORM schema, and (b) an equivalent UML class diagram.

As an extension to UML, the class diagram could be populated by a fact table for the binary association, but UML’s use of attributes makes it less convenient to populate the rest of the diagram (instead of single fact tables, UML uses multiple object diagrams for such purposes). So we do not bother displaying the fact population in UML.

A frequency constraint depicted in ORM by a positive integer corresponds to an “all-or-nothing” case of knowledge completeness. More commonly, frequency constraints simply set just an upper or lower bound on knowledge completeness. For example, suppose we modify our example so that for each duet we may know zero, one, or both of the musicians in the duet. An illustrative population is shown in Table 2.

Table 2 A duet’s membership may be totally unknown, partially known, or fully known.

<i>Duet</i>	<i>Musicians</i>	<i>Popular?</i>
Everly Brothers	Don Everly, Phil Everly	Yes
Two Dons	Don Everly, Donovan Leitch	Yes
Northface Duo	?	No
Minnesota Twins	Pat Barden	No

ORM and UML schemas for this situation are shown in Figure 4. The ORM frequency constraint is now depicted by placing “≤ 2” (less than or equal to 2) next to the relevant role. If you populate this role with the sample membership data, you’ll notice that each duet in the role’s fact column appears there at most twice. In UML, the corresponding multiplicity constraint becomes “0..2”. This frequency constraint may be verbalized thus:

Each Duet includes at most two instances of Musician.

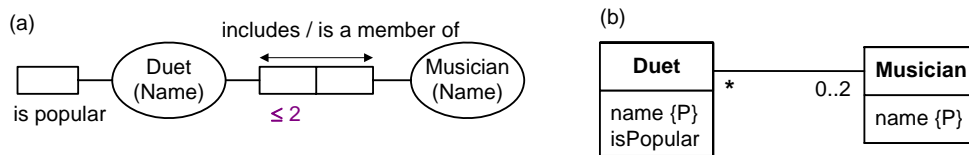


Figure 4 Modeling the structure of Table 2 in (a) ORM and (b) UML.

Now let’s modify our example, requiring that we know who all the members are of each duet. Table 2 illustrates this situation, listing two Northface Duo musicians.

Table 3 The membership of each duet must be fully known.

<i>Duet</i>	<i>Musicians</i>	<i>Popular?</i>
Everly Brothers	Don Everly, Phil Everly	Yes
Two Dons	Don Everly, Donovan Leitch	Yes
Northface Duo	Shania Triad, Suzi Quinquo	No

Figure 5 models the structure of Table 3 in ORM and UML. From an ORM perspective, the frequency constraint of 2 on Duet’s role in the membership fact type is unchanged from the original case for Table 1. However, a mandatory constraint has been added to that role (depicted by a solid dot on its role connector) to indicate that each instance of Duet’s population must play that role. For each state of the business, the term “population” denotes the set of known instances for that state (typically a subset of the type, which includes the instances from all possible states). In UML, the multiplicity constraint becomes “2” (or 2..2).

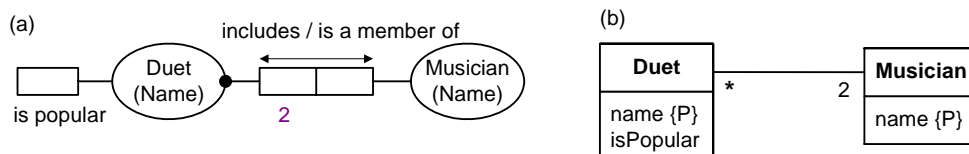


Figure 5 Modeling the structure of Table 3 in (a) ORM and (b) UML.

The frequency and mandatory constraints may be verbalized separately, for example:

Each Duet includes zero or two instances of Musician. -- frequency constraint

Each Duet includes at least one Musician. -- mandatory constraint

Alternatively, the combination of both constraints may be verbalized more compactly thus:

Each Duet includes exactly two instances of Musician. -- frequency and mandatory constraints combined

An ORM frequency constraint may be considered to be a generalization of an ORM uniqueness constraint. Basically, a uniqueness constraint is a frequency constraint where the frequency equals one, but uniqueness is such an important case that it deserves a special treatment and notation of its own, as discussed in earlier articles.

Like uniqueness constraints, frequency constraints in ORM are local to their predicate, in contrast to mandatory role constraints, which have global impact by implying subset constraints involving other roles played by the same object type. This separation of local and global aspects enhances ORM's orthogonality, minimizing the impact of changes to existing constraints. In contrast, UML collapses the two constraints into a single multiplicity constraint, thereby reducing orthogonality. As discussed in the next article, UML's multiplicity constraint is also incapable of specifying certain kinds of ORM frequency constraint when the fact type or association is n -ary (3 roles or more).

Frequency constraints may also specify lower bounds on frequencies, using the notation " $\geq n$ " for "at least n ", or frequency ranges using the notation " $n..m$ " for "at least n and at most m ". Figure 6 shows some examples in ORM, based on an example from [1, p. 281]. The exclamation mark in "Panel !" indicates the independence of this object type (we may know a given instance independently of knowing any other facts about it). The " $4..7$ ", " ≤ 5 ", and " ≥ 2 " respectively denote a frequency range constraint, a maximum frequency constraint, and a minimum frequency constraint.

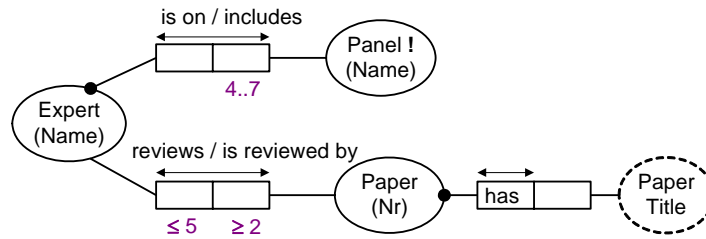


Figure 6 Examples of constraints on frequency minima, maxima, and ranges.

A UML class diagram for the same example is shown in Figure 7. Role names have been added to clarify the semantics. As usual, UML uses multiplicity constraints to capture both mandatory and frequency constraints. This works fine for binary associations, but leads to some problems with n -ary associations, as discussed in the next article.

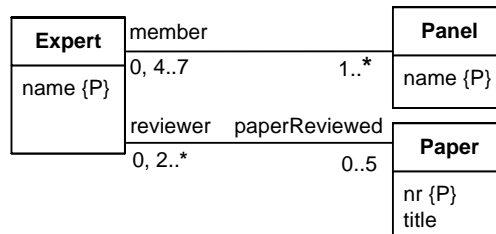


Figure 7 A UML class diagram corresponding to the ORM schema in Figure 6.

The three frequency constraints may be verbalized as follows:

Each Panel that includes an Expert includes at least 4 and at most 7 instances of Expert.

Each Expert reviews at most 5 instances of Paper.

Each Paper that is reviewed by an Expert is reviewed by at least two instances of Expert.

Alternative verbalizations exist (e.g. the third constraint may be recast as: Each Paper is reviewed by either zero instances, or at least two instances, of Expert.). Again, a pluralization engine would enable more natural verbalizations. Though very rare in practice, disjunctions of frequency specifications may be specified using a comma-delimited list (e.g. “3, 5..7, ≥ 10” for “3, or between 5 and 7 inclusive, or at least 10”).

Single role frequency constraints may also apply to n -ary predicates. As a ternary example, consider the report in Table 4. The first column lists the standard colors (unlike custom colors, standard colors are always given unique names). The standard colors Red, Green and Blue are primary in the sense that the other colors may be formed by combining the primary colors in different ratios. For example, Plum is composed of 153 parts of Red (out of maximum possible 255), 51 parts of Green, and 102 parts of Blue. In this table, the nulls indicate that the composition of the Autumn color is unknown.

Table 4 Primary color composition for various standard colors.

Standard Color	Primary Color	Color Intensity
Plum	Red	153
Plum	Green	51
Plum	Blue	102
Red	Red	255
Red	Green	0
Red	Blue	0
Autumn	?	?

Figure 8 shows one way to model this structure in ORM. The frequency constraint of 3 on the first role of the ternary may be verbalized as shown below.

Each StandardColor that includes a PrimaryColor in a ColorIntensity plays that role 3 times.

As usual, the existential quantifier verbalized here as “a” may be reworded as “some” or “at least one”. Here “plays that role” is read in the normal way (i.e. in the least restrictive sense). In other words, for each of the three occurrences of any given StandardColor that plays the role, it is possible that the associated instances of PrimaryColor in the fact table may differ, and likewise for the associated instances of ColorIntensity.

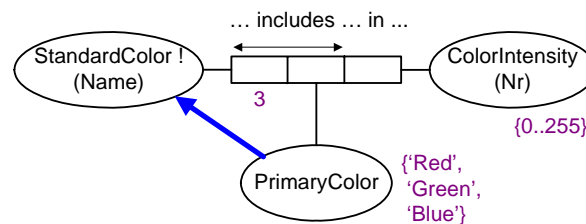


Figure 8 An ORM schema for Table 4.

Although the above situation may be modeled with a ternary association in UML using classes for StandardColor, PrimaryColor, and ColorIntensity, it would be more usual in UML to treat ColorIntensity as an attribute of an association class representing an inclusion association between StandardColor and PrimaryColor. The next article will include examples of frequency constraints over associations that would typically be modeled as n -aries in UML.

That completes our coverage of internal frequency constraints on single roles, and their verbalization. The next article considers verbalization of more complex frequency constraints: multi-role frequency constraints (applying to sequences of two or more roles); and external frequency constraints (involving more than one predicate).

References

1. Halpin, T. A. 2001, *Information Modeling and Relational Databases*, Morgan Kaufmann, San Francisco.
2. Halpin, T. A. 2002, 'Metaschemas for ER, ORM and UML Data Models: A Comparison', *Journal of Database Management*, vol. 13, No. 2, pp. 20-29, Idea Publishing Group, Hershey PA, USA.
3. Halpin, T. A. 2003, 'Verbalizing Business Rules: Part 1', *Business Rules Journal*, Vol. 4, No. 4 (April 2003), URL: <http://www.BRCommunity.com/a2003/b138.html>.
4. Halpin, T. A. 2003, 'Verbalizing Business Rules: Part 2', *Business Rules Journal*, Vol. 4, No. 6 (June 2003), URL: <http://www.BRCommunity.com/a2003/b152.html>.
5. Halpin, T. A. 2003, 'Verbalizing Business Rules: Part 3', *Business Rules Journal*, Vol. 4, No. 8 (August 2003), URL: <http://www.BRCommunity.com/a2003/b163.html>.
6. Halpin, T. A. 2003, 'Verbalizing Business Rules: Part 4', *Business Rules Journal*, Vol. 4, No. 10 (October 2003), URL: <http://www.BRCommunity.com/a2003/b172.html>.
7. Halpin, T. A. 2004, 'Verbalizing Business Rules: Part 5', *Business Rules Journal*, Vol. 5, No. 2 (February 2004), URL: <http://www.BRCommunity.com/a2004/b179.html>.
8. Halpin, T. A. 2004, 'Verbalizing Business Rules: Part 6', *Business Rules Journal*, Vol. 5, No. 4 (April 2004), URL: <http://www.BRCommunity.com/a2004/b183.html>.
9. Halpin, T. A. 2004, 'Verbalizing Business Rules: Part 7', *Business Rules Journal*, Vol. 5, No. 7 (July, 2004), URL: <http://www.BRCommunity.com/a2004/b198.html>.
10. Halpin, T. A. 2004, 'Verbalizing Business Rules: Part 8', *Business Rules Journal*, Vol. 5, No. 9 (September, 2004), URL: <http://www.BRCommunity.com/a2004/b205.html>.
11. Halpin, T. A. 2004, 'Verbalizing Business Rules: Part 9', *Business Rules Journal*, Vol. 5, No. 12 (December, 2004), URL: <http://www.BRCommunity.com/a2004/b215.html>.
12. Halpin, T., Evans, K., Hallock, P. & MacLean, B. 2003, *Database Modeling with Microsoft Visio for Enterprise Architects*, Morgan Kaufmann, San Francisco.
13. Object Management Group 2003, *UML 2.0 Infrastructure*, URL: <http://www.omg.org/uml>.
14. Object Management Group 2003, *UML 2.0 Object Constraint Language*, URL: <http://www.omg.org/uml>.