Verbalizing Business Rules: Part 11

Terry Halpin
Northface University

Business rules should be validated by business domain experts, and hence specified in a language easily understood by business people. This is the eleventh in a series of articles on expressing business rules formally in a high-level, textual language. The first article [2] discussed criteria for a business rules language, and verbalization of simple uniqueness and mandatory constraints on binary associations. Article two [3] examined hyphen-binding, and verbalization of internal uniqueness constraints that span a whole association, or that apply to *n*-ary associations. Article three [4] covered verbalization of basic external uniqueness constraints. Article four [5] 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 [6] 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 [7] considered verbalization of value constraints. Article seven [8] examined verbalization of subset constraints. Article eight [9] discussed verbalization of equality constraints. Article nine [10] covered verbalization of exclusion constraints. Article ten [11] dealt with verbalization of internal frequency constraints on single roles. This 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).

Verbalization of multi-role frequency constraints

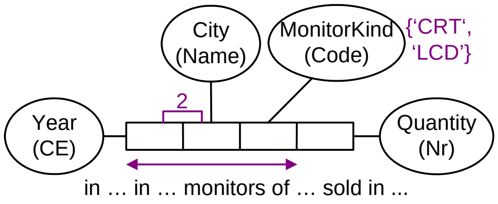
Consider the report displayed in Table 1, which lists an extract of details about sales of computer monitors. In this domain, cities may be identified simply by their name. There are two kinds of computer monitor: cathode ray tube (CRT); and liquid crystal display (LCD). As a way of ensuring one aspect of completeness, the following business rule applies: for each year and city combination, if we record sales figures at all, we must record the sales figures for both kinds of monitor. For example, the Year-City pair "2003, Paris" appears twice in the table, once for each kind of monitor. If "2003, Paris" appeared just once in the table (for CRT, say, but not for LCD) the above business constraint would be violated.

Table 1 Sales of computer monitors.

Year	*City*	*Monitor kind*	*Quantity sold*
2003	Paris	CRT	3000
2003	Paris	LCD	2000
2004	Paris	CRT	3500
2004	Paris	LCD	3500
2003	Tokyo	CRT	4000
2003	Tokyo	LCD	3000
…	…	…	…

Figure 1 schematizes this report in Object-Role Modeling (ORM), using the quaternary fact type "in Year in City monitors of MonitorKind sold in Quantity". The set listing {'CRT', 'LCD'} besides MonitorKind denotes a value constraint, restricting instances in each population of codes for that type to be one of those two values. The arrow-tipped bar on the fact type denotes a uniqueness constraint, indicating that each Year-City-MonitorKind combination in the population of the fact type is unique, and hence is associated in this fact type with at most one Quantity.

The "2" besides the line connecting the roles played by Year and City indicates the frequency (number of occurrences) for each Year-City combination that instantiates that role-pair. So each Year-City pair that appears in the fact table population must appear there exactly twice. Because more than one role is involved, this is a *multi-role frequency constraint*.



2003	Paris	CRT	3000
2003	Paris	LCD	2000
2004	Paris	CRT	3500
2004	Paris	LCD	3500
2003	Tokyo	CRT	4000
2003	Tokyo	LCD	3000

Figure 1 An ORM schema with sample population for Table 1.

Given the uniqueness constraint across the first three roles, and the value constraint on MonitorKind, the frequency constraint completes the declaration of the business rule cited earlier (i.e. for each year and city combination, if we record sales figures at all, we must record the sales figures for both kinds of monitor). This frequency constraint may be verbalized in either of the following ways:

Each Year, City combination occurs zero or 2 times in the population of
"in Year in City monitors of MonitorKind sold in Quantity".

Each Year, City combination that occurs in the population of
"in Year in City monitors of MonitorKind sold in Quantity"
does so 2 times.

The above verbalization patterns may be used if the object types playing the constrained roles play just one role in the fact type. If a constrained object type plays more than one role in the fact type, its role occurrences may be distinguished by subscripting the object type variable. For example, a frequency constraint of 2 on the first two roles of the fact type "from Year to Year the number of staff of Gender changed by Quantity" may be verbalized thus:

Each Year$_1$, Year$_2$ combination occurs zero or 2 times in the population of
"from Year$_1$ to Year$_2$ the number of staff of Gender changed by Quantity".

Figure 2 displays a class diagram in the Unified Modeling Language (UML) notation [13] for our Table 1 example. The population is omitted because UML provides no convenient way of displaying it. The "{P}" after City.name indicates that this attribute provides the preferred identifier for cities, and hence is unique; this notation is not part of standard UML. Similarly, the "{P}" after Year.nr_CE indicates that the preferred identifier for years is their common era number. MonitorKind is specified as an enumeration type, with its possible values listed as if they were attributes.

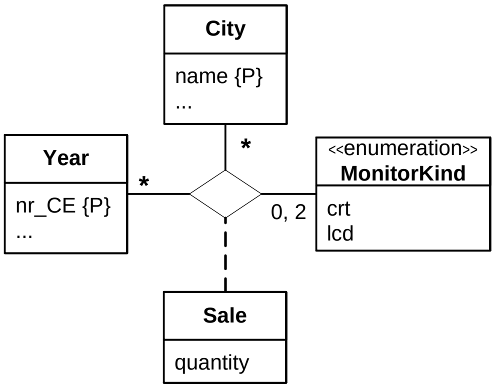


Figure 2 UML class diagram for Table 1.

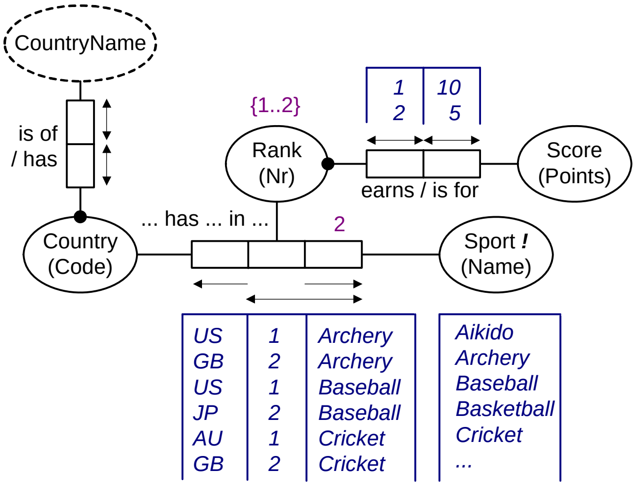
Instead of using a quaternary association in Figure 2, we used a ternary association that may be declared in ORM as "in Year in City monitors of MonitorKind were sold". We then reified this as the association class Sale, and then attached the quantity information as an attribute of Sale. The multiplicity constraint "*" on Year's role indicates that each City-MonitorKind pair in the Cartesian product of the populations of City and MonitorKind is associated in that association with zero or more years. The * on City's role has a similar interpretation. The multiplicity constraint of "0, 2" on MonitorKind's role corresponds to the ORM frequency constraint discussed earlier: each Year-City pair in the Cartesian product of the populations of Year and City is associated in that association with zero or two instances of MonitorKind. The "0" case is included in each multiplicity constraint, since Year, City and MonitorKind may play other roles in the global schema, and we are making no claim that their roles in this fact type are mandatory. The earlier verbalization patterns for the ORM frequency constraint may be adapted to express the "0, 2" multiplicity constraint, simply by using "Sale" for the ternary association, e.g.

Each Year, City combination occurs zero or 2 times in the population of Sale.

The verbalization patterns discussed may be extended in obvious ways to cater for frequency ranges and combinations (e.g. 2..5; ≤ 5; ≥ 2; 3, 5..7) using techniques discussed in the previous article [10].

Some problems with multiplicity constraints on n-ary associations

Just as a UML multiplicity constraint cannot capture a mandatory constraint on an *n*-ary association if the constraint applies to fewer than *n*-1 roles [6], a UML multiplicity constraint cannot capture a minimum frequency constraint greater than 1 on an *n*-ary association if the constraint applies to fewer than *n*-1 roles. For example, consider the ORM model depicted in Figure 3. In this domain, if a sport has a country's rank recorded for it, it must list exactly the first and second ranked countries. Here the frequency constraint of 2 indicates that each sport that plays in the ternary fact type does so twice. In the context of the value and uniqueness constraints, this entails that each instance in the population of Sport has either zero or two rankings recorded for it.



1	10
2	5

US	1	Archery
GB	2	Archery
US	1	Baseball
JP	2	Baseball
AU	1	Cricket
GB	2	Cricket

Aikido
Archery
Baseball
Basketball
Cricket
...

Figure 3 ORM schema with a frequency constraint requiring each sport to have 0 or 2 ranks.

This frequency constraint cannot be captured by a multiplicity constraint in UML. In the UML class diagram shown in Figure 4 it is declared informally as a note. Alternatively, we could assign a name to the ternary association (e.g. Ranking), and then verbalize the constraint as: Each Sport occurs zero or 2 times in the population of Ranking. In UML the constraint could also be declared formally in the Object Constraint Language (OCL) notation [14]; however that notation is typically too technical for most business people.

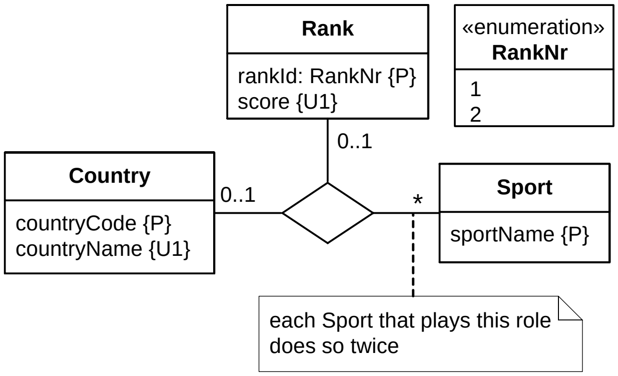


Figure 4 UML class diagram for the ORM schema in Figure 3.

Verbalization of external frequency constraints

All the frequency constraints considered so far have been internal constraints, applying to a single predicate. A frequency constraint is a generalization of a uniqueness constraint (where the frequency = 1). Just like uniqueness constraints, frequency constraints also have an external version, where the constraint applies to roles from two or more predicates. In practice, such *external frequency constraints* are far less common than external uniqueness constraints; but they may occur in practical cases, so we need a way to verbalize them.

Consider the ORM schema shown in Figure 5. As discussed in previous articles, a circled “u” denotes an external uniqueness constraint [4, 5]. The left-most external uniqueness constraint indicates that in this context each Semester-Student-Course combination is associated with at most one Enrollment. The other external uniqueness constraint declares that in this context each Year-SemesterNr combination is associated with at most one Semester.

The circled “≤ 2” denotes an external frequency constraint, indicating that in this context each Student-Course combination is associated with at most two enrollments. In other words, each student may enroll at most twice in the same course—this business constraint is adopted by many universities.

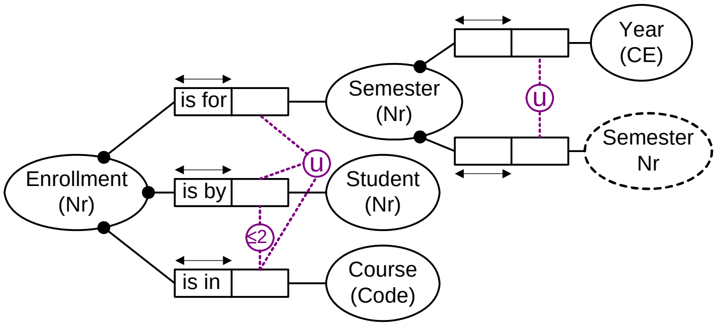


Figure 5 Example of an ORM external frequency constraint.

To clarify the semantics of the external frequency constraint, suppose we modeled the enrollment information instead using the ternary fact type Student enrolled in Course for Semester. In this case, the ternary would have a spanning, internal uniqueness constraint (corresponding to the first external uniqueness constraint), as well as an internal frequency constraint of “≤ 2” over the first two roles (corresponding to the external frequency constraint).

The external frequency constraint in Figure 5 may be verbalized in several ways. Some of the main alternatives are as follows:

Each Student, Course combination occurs at most 2 times in the context of:
  Enrollment is by Student; Enrollment is in Course.

Context: Enrollment is by Student; Enrollment is in Course.
  In this context: each Student, Course combination is associated with at most 2 instances of Enrollment.

Given any Student and Course
  there are at most 2 instances of Enrollment
    where that Enrollment is by that Student and is in that Course.

As with internal frequency constraints, verbalization of external frequency constraints may cater for exact frequencies, various frequency ranges, and combinations (e.g. 2..5; ≥ 2; 3, 5..7) using techniques discussed in the previous article [10].

That completes our coverage of frequency constraints. The next article considers verbalization of ring constraints, and the article after that will consider verbalization of subtype constraints.

References

1. Halpin, T. A. 2001, *Information Modeling and Relational Databases*, Morgan Kaufmann, San Francisco.
2. 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.
3. 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.
4. 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. Halpin, T. A. 2005, 'Verbalizing Business Rules: Part 10', *Business Rules Journal*, Vol. 6, No. 4 (April, 2005), URL: http://www.BRCommunity.com/a2005/b229.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.