This document is published in:

Software & Systems Modeling 12 (2013) 4-October, pp 871-889

DOI: 10.1007/s10270-012-0234-3

Guidelines for representing complex cardinality constraints in binary and ternary relationships

Dolores Cuadra · Paloma Martínez · Elena Castro · Harith Al-Jumaily

Abstract Ternary relationships represent the association among three entities whose constraints database designers do not always know how to manage. In other words, it is very difficult for the designer to detect, represent and add constraints in a ternary relationship according to the domain requirements. To remedy the shortcomings in capturing the semantics required for the representation of this kind of relationship, the present paper discusses a practical method to motivate the designer's use of ternary relationships in a methodological framework. The method shows how to calculate cardinality constraints in binary and ternary relationships and to preserve the associated semantics until the implementation phase of the database development method.

Keywords Ternary associations · Conceptual models · Logical models · Model transformations · Database methodology

1 Introduction

Databases are among some of the most important components in the development and management of information systems. Good database design should preserve the integrity, consistency and safety of the data and simplify the management of information systems. With the continuous changes in business organizations and the proliferation of models witnessed during the information technology boom, it has become clear that the first phases in database design are crucial for the obtainment of a flexible, adaptable and complete system. Conceptual data models are presented by Moody and Shanks [42] as a key factor in development cost and flexibility, as well as in a system's ability to integrate with other systems and meet user requirements. Conceptual models have been proposed in database design as tools to obtain physical/logical independence. This independence is achieved from the abstraction level upwards. One of the first of such approaches was by Abrial [1] which explains the advantages of using semantic models for problem resolution and knowledge representation. This approach dealt with semantic models in such areas as artificial intelligence, software engineering and database design and discussed how to use these models as a starting point for the representation of the Universe of Discourse (UD) in a natural and intuitive way.

As discussed in Parent et al. [43], a conceptual model is that which should be able to provide a direct mapping between the perceived real world and its representation without distortion or ambiguity. In general, conceptual models are defined as being easy to learn and use, intuitively understandable, independent of implementation and able to represent any domain. Graphical or visual languages for making schemata are used to facilitate discussion with, and validation by, domain experts. Although conceptual models can enhance the properties discussed above (i.e., ease of learning, intuitiveness, etc.), particular studies [44,45] have identified particular problems. For instance, the use of the term 'simplicity' in describing a conceptual model refers to the

Communicated by Dr. Perdita Stevens.

D. Cuadra (✉) · P. Martínez · E. Castro · H. Al-Jumaily
Computer Science Department, Carlos III University of Madrid,
Avd. Universidad, 30, 28911 Leganés, Spain
e-mail: dcuadra@inf.uc3m.es

P. Martínez
e-mail: pmf@inf.uc3m.es

E. Castro
e-mail: ecastro@inf.uc3m.es

H. Al-Jumaily
e-mail: haljumai@inf.uc3m.es

simplicity of the concept, not necessarily to the level of complexity of the model itself. Nevertheless, 'simplicity' in this context has sometimes been incorrectly interpreted as meaning that only a few elements are provided for real-world modeling. For example, binary models are models which have no n-ary (i.e., no $n > 2$) relationships—as in the Case*Method [7] or IDEF1X [11]—or models that only define maximum cardinality constraints associated with relationships [24,52]. This type of model does not have a sufficient number of elements for the complete representation of certain real domains. Nevertheless, it is important not to confuse simplicity (i.e., a lack of complexity) with the 'simplicity' of a conceptual model. In this latter sense, 'simplicity' does not imply that a model is basic or trivial. The approach taken in Gemino and Wand [28] used the cognitive theory of multimedia learning to examine the truth of the perception that a model with mandatory properties is more difficult than one with optional properties. However, the experiment showed that the conceptual model with mandatory constraints is in fact easier to use. This means that the apparent complexity of the conceptual model can serve to clarify the real-world representation. It is very important to give a formal description of each conceptual element, to avoid mistakes in the use of these elements and to facilitate easy learning. According to Harel and Rumpe [32], one source of confusion at the moment when conceptual model elements are to be distinguished is the muddling of the distinction between the meanings of 'data' and 'information', an error which can be avoided by reflection upon the difference between syntax and semantics. Language consists of a syntactic notation—that is, an infinite set of legal elements—together with a semantic meaning of those elements and those two aspects must be clearly defined and differentiated. Conceptual syntax is visual while semantics are not always specified.

As discussed in Davies et al. [21], one of the most frequently-used conceptual models is the extended entity-relationship (EER) model. Primarily due to the adequate level of abstraction of the elements included in the EER model, the model is a precise and comprehensive tool for the representation of requirements in information systems. Since the proposal of the original EER model by Chen [16] in 1976, several extensions and variations, as well as different diagrammatic styles, have been defined [41,43]. Other widely-used conceptual models include the object-role model (ORM) [30] and the Unified Modeling Language (UML) notation with particular emphasis on the class diagram [56]. Each of these has been extended to represent more semantics, including new or redefined elements. However, the proliferation of conceptual models has also given rise to some confusion inasmuch as the same notation or syntax can have different semantics depending on the model used. To give one example, while the cardinality constraint in a ternary relationship has the same notation for the Entity-Category-Relationship (ECR) model [24] and UML class diagrams, it also has different semantics. These semantics define the constraints in two different ways, using the Merise method [50] in the case of the former and Chen's approach [29] in the case of the latter. Semantic ambiguity seriously affects designers, both in the design process (e.g., the mapping between a conceptual to relational schema) and when the schema integration comes from several conceptual models. From a logical perspective, cardinality constraints have been dealt with. In one recent work [34], a logic-based propositional tool was presented for the acquisition of integrity constraints that are strongly linked with functional dependencies.

Furthermore, having conceptual models that are easy to learn and use, intuitive, independent of implementation and powerful does not necessarily guarantee, however, that the modeling process will be simple. The designer must be able to detect conceptual elements in the domain specifications which generally appear textually and through interviews with the domain expert. This process, known as the abstraction process, maps the domain specifications in conceptual elements such as entities, attributes or relationships. The process can be understood as a translation of sentences in natural language into elements belonging to a visual language. This translation is not trivial and several studies have focused on determining the rules necessary for its automation. Some heuristic rules to facilitate this process have been offered, for instance, in studies focusing on the types of problems encountered by novice designers in the abstraction process [9,10]. Many experts agree that conceptual elements of associated semantics are more difficult to detect and understand than to represent. For example, relationships are more difficult to understand as their degree increases. In Castro et al. [15], a number of common mistakes in relationship modeling are presented. These results suggest that CASE tools could be of help in relationship modeling, a proposition specifically addressed by certain studies.

The approach taken by the present paper focuses on binary and ternary relationships and the definition of their cardinality constraints. A study of several approaches to the representation of cardinality constraints is undertaken and the practical method presented later is advocated for in hopes that it may bring experts a bit closer towards a consensus view on how to handle cardinality constraints. As this is framed within the relational database methodology, the transformation rules outlined in the majority of textbooks on database design [8,24,31,39,46,52] will be employed and extended.

Regarding the structure of the remainder of the present paper, Sect. 2 studies the definitions of the different relationships in the best-known conceptual models focusing on ternary relationships and their cardinality constraints. In Sect. 3, a practical method for calculating cardinality constraints is presented that responds to shortcomings found in a previous study on relationships. While the methods are addressed to the EER model, they could nevertheless also be applied to other conceptual models that follow the Merise method or Chen's approach, for example, the UML class diagram using Chen's approach and therefore, the method to complete constraints will be the Merise and Participation cardinalities (see Sect. 3.1.). Similarly, for the IDEF1X model using the Merise method, to complete the cardinality constraints, Chen and Participation must be defined (see Sect. 3.1.). Section 4 presents a set of transformation rules for the preservation of semantics in the relational phase of database development. In Sect. 5, empirical results from the evaluation of the proposal's utility are presented. Finally, Sect. 6 offers concluding remarks, describes ongoing work in the area and proposes additional areas for future research.

2 Related work

The present section attempts to elucidate the distinction between syntax and semantics in order to define the cardinality constraints in the most relevant conceptual models. As mentioned earlier, the confusion between these two concepts is often the cause of mistakes made by novice database designers. As one of the keys to the correct modeling of cardinality constraints is the clarification of the relationship concept [6], this section first presents different definitions and representations for the relationship concept and their properties.

2.1 Review of relationship definitions

In general, a relationship is defined as an association between elements. This simple-sounding definition, however, has several different interpretations. For example, Thalheim [54] presents the higher-order entity-relationship model (HERM) which defines relationships between relationships and relationships between higher degree relationships. Elmasri and Navathe [24], nevertheless, only consider the relationship between entities while Teorey [52] distinguishes groups of entities and relationships through clusters. Therefore, the association or relationship can be between entities, relationships or clusters of entities. Compounding this complexity, the term, 'association', can also have different senses depending on the perspective of the speaker. Thus, 'marriage' can be seen as a relationship usually existing between a man and a woman or as an entity representing a social and legal concept. This duality often produces confusion in the modeling of this conceptual element.

Furthermore, models such as the ORM [30] not using the relationship element define associations between objects as the set of predicates that are distinguished by a name and applied to a set of roles compiling a set of *Facts*. Although the representation is different, the concept itself is the same. The definition of association by the ORM is similar to that in McAllister [41].

Other well-known conceptual models [24,30,41,54] share the instances of a relationship definition; that is, the instance is an element of the Cartesian product between instances of entities, although the HERM [54] also includes the cluster and associations among relationships. Unfortunately, most designers are confused about n-ary or higher relationships and either do not use them or often use them erroneously. Mistakes regarding the degree of relationship choice, incorrect cardinalities and errors in the transformation of conceptual schemata to relational schemata are common (see Batra and Antony [10] and Castro et al. [15]).

2.2 Review of the syntax and semantics of cardinality constraints

The problems in relationship modeling can be found in the detection and representation of its properties, such as cardinality constraints. Detection problems in ternary relationships are described in Castro et al. [15] a study presenting experimental results aimed to demonstrate difficulties encountered by novice students and practitioners in the use of ternary relationships. Some of the relevant factors detected that influence the modeling of ternary relationships include domain knowledge and intersection data (i.e., attributes in ternary relationships). The representation of cardinality constraints was another of the problems detected that lead to an incorrectly modeled relationship.

Cardinality constraints can be defined as the limit on the combination of relationship instances or, in other words, the description of the valid subset of the Cartesian product of instances belonging to the relationship.

Cardinality constraints have many associated syntaxes and semantics [26]. Syntax is represented through visual languages and is not standard since it depends on the relationship definition adopted by the conceptual model chosen. While some conceptual models represent only maximum cardinalities [52], others represent both maximum and minimum cardinalities with labels close to or opposite the site associated with the entity. For example, Fig. 1 (left) shows minimum and maximum cardinalities represented by labels which, following Elmasri and Navathe [24], are located opposite the site of the entity associated with this cardinality. The semantics collected for the model are 'an employee works in one and only one department'. The same semantics can also be collected from Fig. 1 (right) using Teorey notation [52] where the shaded area represents a maximum cardinality of n. This graphical notation only allows maximum cardinalities of 1 or n.

The semantics of cardinality constraints are also ambiguous. Two principal approaches can be distinguished [26]. First, Chen's constraints are an extension of the mapping

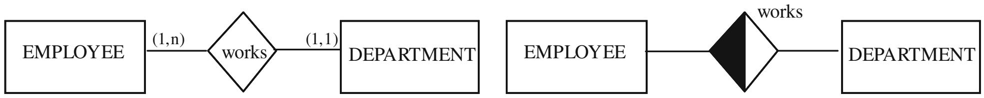


Fig. 1 Representing cardinality constraints in ECR model (*left*) and with Teorey notation (*right*)

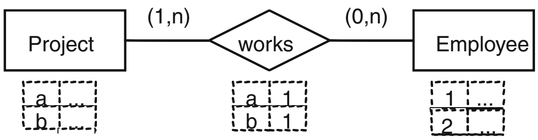


Fig. 2 Merise method for the representation of cardinality constraints

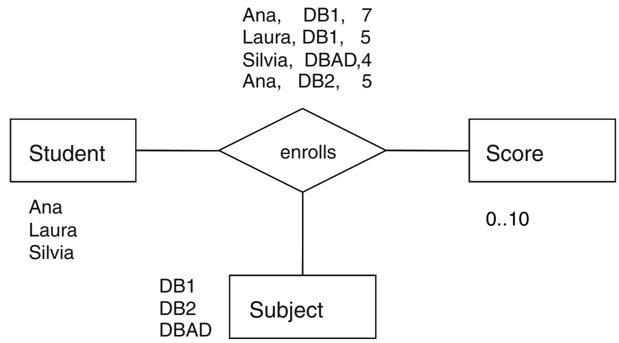


Fig. 3 Ternary relationship and its associated instances

constraint (i.e., a special case of cardinality constraint that considers only the maximum cardinality which for binary relationships, for example, can be 1:1, 1:N or N:M) [16]. They have been adopted or extended in different data models. Second, the Merise method [50] incorporates participation semantics in a relationship. For binary relationships, the two approaches agree in cardinality constraints (except in graphical notations or syntax) and represent the same semantics.

For both approaches, two different interpretations can be made in calculating cardinality constraints depending on whether entity instances (i.e., potential instances or incomplete instances) or relationship instances (i.e., real instances) are taken into account [29].

The first semantic interpretation specifies the maximum and minimum number of instances of an entity or an entity subset that can participate in relationship instances. For a binary relationship, the interpretation is that if the entity's minimum cardinality is zero, the entity's participation in a relationship is optional. In another case, the $i \geq 1$ minimum cardinality forces each entity instance to participate in the relationship i-times. The studies of Elmasri and Navathe [24] and Halpin [30] are among those using this interpretation, known as the Merise method or participation constraint.

Figure 2 shows an example of a binary relationship according to the Merise method. The related semantics are 'every *project* instance must participate in the *works* relationship, but an *employee* instance need not be in the *works* relationship'. The cardinality constraints are represented by a label next to the associated entity.

Most conceptual models [7,11,24,30,31,39,46,48,52] use this interpretation with a different representation but the same semantics. The semantics can be calculated by looking at the entity instances and counting the maximum and minimum number of times that they appear in the relationship. This interpretation is termed '*EntityLook*' because it focuses on instances of the entity. Nevertheless, there is a problem with this interpretation. While mandatory participation forces all entity instances to participate in the relationship, what would happen if participation were not mandatory? Taking the previous example, for instance, if an employee works on a project, he must work on at least two projects. However, as there are employees who do not have any associated projects, how may this specification be represented? Notice that this constraint cannot be represented with minimum and maximum cardinalities.

The second semantic interpretation looks at relationship instances and is termed '*RelationshipLook*'. Setting one of the components or roles in the relationship, the interpretation obtains the maximum and minimum number of different combinations of that component or role with the remaining components or roles. This approach is used in Halpin [30], but the semantic values are constrained to being greater than zero. If this interpretation is applied to the previous example shown in Fig. 2, while the *Project* cardinality constraint does not change, the *Employee* cardinality becomes (2,n).

In summary, while the binary relationship is satisfactorily defined by examining the syntax and semantics, a deficiency becomes apparent in the case of optional participation.

2.3 Motivation

For ternary relationships, this deficiency becomes even more important due to the fact that the Merise method and Chen's approach do not agree (some examples of this assertion are presented in [49]). Looking at Fig. 3, the constraints associated with the *Score* entity can be obtained by considering *Score* instances and seeing that the latter's participation is mandatory in *enroll*. However, regarding the *enroll* role, the *score* role or component could be optional. This participation constraint or Merise method depends on *EntityLook* or *RelationshipLook*, respectively. Another constraint is observed in the *enroll* relationship: the *student* and *subject* combination is unique (known in the relational framework as functional dependency). The combination of instances defines the other constraint type, Chen's approach, yet the combination between two entities is not clear. Is the combination between two entity instances (i.e., potential instances)? Or is the combination between two components (i.e., roles) in the relationship (i.e., real instances)?

The interpretation of potential instances determines a (0,1) cardinality insomuch as there are instance combinations between *Student* and *Subject* entities that do not appear in the *enrolls* relationship, for example, the *(Ana,DBAD)* combination. However, the real instance interpretation determines a (1,1) cardinality and only considers the instances in *enrolls*.

Therefore, cardinality constraints could be classified according to either the Merise method or Chen's approach, as well as according to the EntityLook (potential instances) or RelationshipLook (real instances) interpretations.

Cardinality constraints are divided into four types, numbered from (1) to (4). The semantics associated with the maximum cardinality is the same for each of these types. Nevertheless, a problem is encountered with the minimum cardinality. The first type of cardinality constraint (1) is the Merise method [50] which uses the *EntityLook* interpretation. By this interpretation, the minimum cardinality n, where n is greater than zero, represents the mandatory participation for all instances associated with an entity in a relationship (e.g., a student must be enrolled in at least one subject). The zero cardinality is the optional participation of entity instances in a relationship. The second constraint type (2) is Chen's approach with the *EntityLook* interpretation, reflecting that when the minimum cardinality is zero, participation is optional. An example of this is that the (*Ana*,*DBAD*) combination does not participate in the relationship (Fig. 3). If the minimum cardinality is 1, all instance combinations must appear at least once in the relationship. Constraint types (3) and (4) look at the relationship. They set either one or two components using the Merise method or Chen's approach, respectively, and count how many times these appear in the relationship. The minimum cardinality of zero only makes sense if the relationship admits components without information. Some approaches treat the relationships with unknown or inapplicable information. Allowing this kind of cardinality directly affects the relationship definition. Most conceptual models define cardinality constraints as being of the fourth type, with the UML class diagram [56] or the EER model used by Elmasri and Navathe [24] being relevant examples.

According to the reviews presented in the previous sections, some conclusions can be extracted.

1. The inclusion of ternary relationships in conceptual models is not a problem with regard to the simplicity property. This kind of association appears in real-world problems and conceptual models must provide some mechanism to represent it.
2. Both syntactical and semantic definitions are ambiguous. The problem is not the lack of formal proposals that solve this ambiguity, but rather the lack of understanding to use this relationship and manage its constraints.
3. The use of the Merise method or Chen's approach to manage the cardinality constraint is not sufficient in and of itself. To reflect all semantics associated with a ternary relationship, both approaches must be used jointly.

This paper presents a practical method for the use of binary and ternary relationships with their cardinality constraints while ensuring the complete management of the relationship semantics. The present section has discussed the different ways in which the cardinality constraints associated with a relationship may be obtained. In this discussion, however, particular deficiencies in the semantics of the cardinality constraints for both binary and ternary relationships have been detected and analyzed. While a recent study by Calí [12] has addressed this issue, it has shown cardinality constraint representation through relational structures, but in a highly formal manner. By contrast, the present paper proposes a practical method based on the work of Jones and Song [38] and McAllister [41] for the use, representation and validation of semantics without information loss or other, previously discussed ambiguities. In the latter of these two studies, a complete approach for the representation of cardinality constraints in n-ary relationships is presented. Nevertheless and similar to the study by Calí, the paper gives a very formal approximation that is difficult to deal with. Finally, the approach in Jones and Song [38] deals with transformation rules, yet it does not include the use of the Merise method.

3 Defining cardinality constraints: syntax and semantics

The present section begins with the definition of certain basic elements used in the conceptual model. Subsequently, several cardinality constraints associated with relationships are proposed, along with numerous results that verify these constraints.

The paragraphs directly below present the definitions of the basic elements used in the conceptual model, namely 'entity' or 'class' and 'relationship' or 'association'. Through such formal definitions, the cardinality constraints of a relationship will then be built.

Definition 1 Let $E = (A_1, \ldots, A_n, \{id_j(E) | 1 \leq j \leq n\})$ be an entity type such that A_i is an attribute defined in a data domain and $\{id_j(E) | 1 \leq j \leq n\}$ a set of unique keys that identifies the instances of E[1].

[1] A 'unique key' is a single column (or set of columns) that uniquely identifies (or identify) each row in a table.

The Primary Key (PK) can be distinguished in the unique key set. The PK is mandatory and is one of the constraints imposed in most conceptual models.

E^t is defined as the set of instances for E. Elements belonging to E^t and denoted by $e^t \in E^t$ are vectors whose ith component verifies that $e_i \in dom(A_i)$ and that each instance is unique.

The set of key instances is denoted by $\#E = \Pi_{PK}(E)$, where Π is the projection of E on PK.

Definition 2 Let $R = (r_1E_1, \ldots, r_nE_j, A_1, \ldots, A_s)$ be a relationship R with order n and attributes A_i where r_i is the role of E_k in R.

R^t is the set of the instances of R and r^t is an element of this set whose representation is a vector with n components. The ith component is a key instance of the entity which participates with the ith role (denoted by PK_i).

For example, the employee entity could be defined as *Employee* = (*ID_Emp, NSS, Name, Address;* {*ID_Emp, NSS*}) where *ID_Emp* is the PK. *Employee* instances are $Employee^t$ ={(*001, 123087400, Charlie Smith, 265 Soldier Home Rd*), (*002, 560236098, Lanna Harrison, 143 Alabama Rd*)}. Considering the relationship between an employee and supervisor, the relationship *Supervise = (Is_Supervised Employee, Supervises Employee)* can be defined. *Supervise* has degree 2 and one instance of this relationship could be $Supervise^t$ = *{(002,001)}*, or the semantic representation, 'Lanna Harrison is supervised by Charlie Smith'.

3.1 Cardinality constraints

The proposal presented here combines cardinalities from the Merise method and Chen's approach with entity participation. Based on the cardinality constraints explained in Sect. 2, a new notation is added to consider different cardinality constraints associated with n-ary relationships (in this case, ternary relationships) and reflect the most important semantics. Also as discussed in Sect. 2, conceptual models have two ways of representing cardinality constraints, namely Entity-Look and RelationshipLook. Our proposal uses both of these to complete and extend the cardinality constraint definition.

The approach presented here depends upon three concepts—entity participation, Merise cardinality constraints and Chen cardinality constraints—based on binary relationships and is defined below.

3.1.1 Entity participation cardinality constraint

Let $R = (r_1E_1, \ldots, r_nE_j, A_1, \ldots, A_s)$ be a relationship where E_j is the entity participation cardinality and r_k a role. When the participation E_j in R with role r_k is optional, cardinality can be defined as $C(r_kE_j, R) = 0$. When the participation of E_j in R with role r_k is mandatory, cardinality

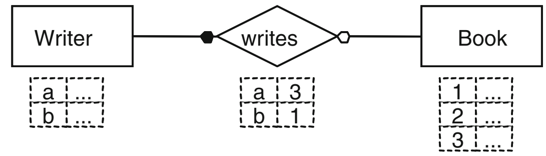


Fig. 4 Example of entity participation cardinality

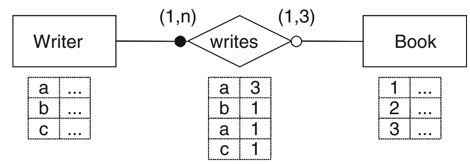


Fig. 5 Merise cardinality combined with participation cardinality

can be defined as ($C(r_kE_j, R) = 1$). A formal expression is presented below:

$$C\left(r_kE_j, R\right) = 1 \text{ iff } \forall \mathrm{PK}_i \in \#E_j,\ \exists r^t \in R^t \text{ such that } r_k^t = \mathrm{PK}_i \text{ where } r_k^t \text{ is the } k\text{th component of the } r^t. \text{ In the other case, } C\left(r_k\ E_j,\ R\right) = 0 \quad (1)$$

This mandatory entity participation can be represented in a conceptual model by a black circle at the endpoint of a line between the entity and the relationship, an example of which can be found below in Fig. 4.

The example shows that all *writer* instances participate in the *writes* relationship, but a *book* need not participate in the *writes* relationship since a book can be written by an anonymous writer. This constraint is associated with instances of one entity and determines the participation type of this entity in the relationship.

3.1.2 Merise cardinality constraint

Let $R = (r_1E_1, \ldots, r_nE_j, A_1, \ldots, A_s)$ be a relationship. Merise cardinality for role r_i can therefore be defined as $CMerise(r_iE_j, R)$ = (min, max) where min ∈ *{1…N}*, max ∈ *{1…N} U {'n'}* and 1 ≤ min ≤ max, if key instance PK_i of r_iE_j in R^t appears in R^t min and max times, respectively, as minimum and maximum. Formally, then, Merise cardinality can be expressed in the manner presented below:

$$CMerise\left(r_iE_j, R\right) = (\min, \max) \text{ iff } \min \leq |\{r^t \in R^t / r_i^t = \mathrm{PK}_i\}| \leq \max\ \forall \mathrm{PK}_i \in r_iE_j \quad (2)$$

$|M|$ number of elements in the M set,
'n' no-restricted value,
r_i^t the ith component of the vector, and
r_iE_j the ith component of the r^t vector such that $r^t \in R^t$. As shown in Fig. 5, the representation of Merise cardinality is a label at the endpoint of a line between the entity and the relationship. It represents the requirement that the writers and books related to the *writes* relationship must fulfill.

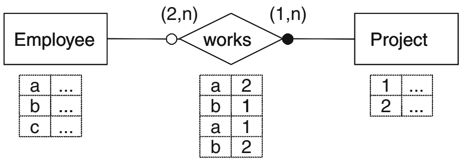


Fig. 6 Cardinality constraint proposal applied to *works* relationship

The example shows that all writers write one or many books and, if all writers of one book are known, up to three writers are stored.

It is important to consider the difference between Fig. 5 and Fig. 4, the latter of which representing the case that a book cannot participate in the relationship *writes* if the author of the book is unknown. Such a requirement cannot be represented with the Merise method. Figure 5 shows how, by combining this method with the participation cardinality, all requirements can be represented.

It is very important to emphasize the semantics that can be represented with the Merise cardinality and participation cardinality in a conceptual schema. These semantics are shown in the following example which represents the requirement: 'A company's employees can participate in projects or not. Those employees who participate in projects are always participants in at least two projects'. As demonstrated in Fig. 6, the method for the representation of this requirement is to divide the constraints into entity participation and participation inside the relationship.

The associated semantics are adequately completed, in that they represent the previously expressed requirements that in the *Employee* there can be instances that do not participate in the *works* relationship, but those that do always participate appear at least twice.

The ternary relationship has greater associated semantics than the binary relationship and different cardinality constraints must be specified to support these semantics. For this reason, the cardinality constraint is defined as follows.

3.1.3 Chen cardinality constraint

Let $R = (r_1 E_1, \ldots, r_n E_j, A_1, \ldots, A_s)$ be a relationship. The Chen cardinality of the role r_i is defined as $CChen(r_i E_j, R) = (\min, \max)$, where min $\in$ *{1…N}*, max $\in$ *{1…N} U {'n'}*, and $1 \leq \min \leq \max$, if one combination of *n*–1 keys given $(PK_1, \ldots, PK_{i-1}, PK_{i+1}, \ldots, PK_n)$ $\in R^t$, appears in R^t min and max number of times as minimum and maximum.

$$\begin{aligned} &CChen\left(r_i E_j, R\right) = (\min, \max) \text{ iff} \\ &\quad \min \leq |\{r^t \in R^t \left|\left(r_1^t, \ldots, r_{i-1}^t, r_{i+1}^t, \ldots, r_n^t\right)\right. \\ &\quad = (\mathrm{PK}_1, \ldots, \mathrm{PK}_{i-1}, \ \mathrm{PK}_{i+1}, \ldots, \mathrm{PK}_n)\}| \\ &\quad \leq \max \forall\, (\mathrm{PK}_1, \ldots, \ \mathrm{PK}_{i-1}, \ \mathrm{PK}_{i+1}, \ldots, \mathrm{PK}_n) \\ &\quad \in r_1 E_1, \ldots, r_{i-1} E_s \ldots, r_{i+1} E_t, \ldots r_n E_j \end{aligned} \tag{3}$$

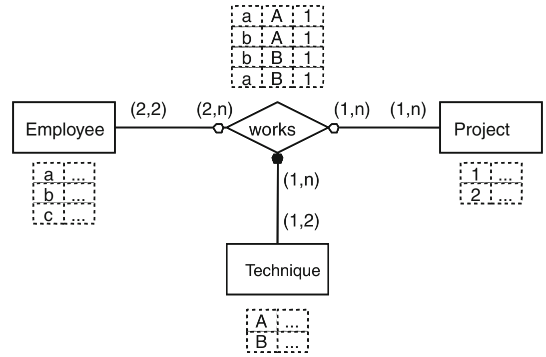


Fig. 7 Example showing cardinality constraint proposal with entity cardinality added

The representation of the Chen cardinality is a label on the opposite side of the role associated with these constraints.

In order to clarify the differences between the Chen and Merise cardinality approaches, it may be instructive to suppose that further conditions apply to the example presented above in Fig. 6. As opposed to the original representation, one may imagine that the techniques which the employees use in each project must be included, taking into account that an employee who participates in a project uses at least one technique and normally two. Additionally, any particular technique applied in a project can be used by only two employees. Figure 7 illustrates how the Merise method collects the semantics of this wording.

Since the Merise method only takes a key instance of entity into account to find its participation with the two other key instances of entities in the relationship, the restriction on the maximum number of techniques applied by an employee on each project cannot be indicated. Moreover, the maximum number of employees who use a particular technique in a specific project cannot be represented either. Chen's approach, however, collects all of the restrictions, given that the approximation combines two key instances of entities to look at the participation of the other key instance of entity in the relationship. Finally, the entity cardinality represents the optional or mandatory participation of each entity in the *works* relationship. According to the example, the *Technique* entity ensures that all their key instances take part in the *works* relationship. The remaining cardinality constraints identify the rules for working on a project, namely that only two employees can work on a project using the same technique. If one employee participates in a project, she/he uses two techniques, at most, and each technique used in a project is used by two and only two employees. Finally, all employees work on the same project with two different techniques or on at least two projects.

3.2 Validation of cardinality constraints

An important aspect of conceptual design is the checking of whether the conceptual schema is well formed and represents the Universe of Discourse (UD) [22,41]. A conceptual schema is well formed if there is an instance that fulfills all schema constraints. If this instance is a real instance, then the schema reflects the UD. The first of these two conditions (or validations) is known as the structural validation and the second as the semantics validation. Both are based on the approach in García-Molina et al. [27] in which integrity constraints in the relational model are sorted into structural and semantics constraints. Other research carried out within the relational model framework includes Hartmann [33] and Ishakbeyoglu and Özsoyoglu [37].

The framework of this paper is the relational database development methodology and, thus, the first validation carried out in the development of the database must be done in the conceptual phase. This validation is extremely important in order to guarantee a good design, refining the schema and proceeding to the next phase in the relational methodology. The matter of the structural validation of the cardinality constraint associated with a relationship is addressed since this type of validation is domain independent and can be carried out through a set of verification rules.

Semantic validation consists of the checking of whether the conceptual schema represents business rules or a correct description of the real world. Therefore, the most efficient semantic validation is that done through domain expert interviews.

The structural validation of cardinalities must make sure that the cardinalities in a relationship are not contradictory. Consequently, at least one relationship instance must be found which verifies all constraints defined with respect to the relationship.

Formally, let $\mathrm{R} = (\mathrm{r}_1\mathrm{E}_1, \ldots, \mathrm{r}_n\mathrm{E}_j, \mathrm{A}_1, \ldots, \mathrm{A}_s)$ be a binary/ternary relationship:

$$C(E_i, R) = 1 \text{ is not valid iff } \exists a \in \#E_i \text{ so that } \forall r^t \in R^t r_i^t \neq a. \quad (4)$$

CMerise $(r_i E_i, R) = (\min, \max)$ is not valid iff :

(a) $min > 1$ and $\exists a \in r_i E_i$ such that $|\{r^t \in R^t \,|r_i^t = a\}| < \min$ or

(b) $\exists\, \mathrm{PK}_i \in r_i E_i$ such that $|\{r^t \in R^t \,|r_i^t = \mathrm{PK}_i\}| > \max$ (5)

CChen $(r_i E_i, R) = (\min, \max)$ is not valid iff :

(a) $\min > 1$ and $\exists\, \mathrm{PK}_1, \ldots, \mathrm{PK}_{i-1}, \mathrm{PK}_{i+1}, \ldots, \mathrm{PK}_n \in r_1 E_1, \ldots, r_{i-1} E_s, r_{i+1} E_t, \ldots r_n E_j$ such that
$$|\{r^t \in R^t | \left(r_1^t, \ldots, r_{i-1}^t, r_{i+1}^t, \ldots, r_n^t\right) = (\mathrm{PK}_1, \ldots, \mathrm{PK}_{i-1}, \mathrm{PK}_{i+1}, \ldots, \mathrm{PK}_n)\}| < \min \text{ or}$$

(b) $\exists\, \mathrm{PK}_1, \ldots, \mathrm{PK}_{i-1}, \mathrm{PK}_{i+1}, \ldots, \mathrm{PK}_n \in r_1 E_1, \ldots, r_{i-1} E_s, r_{i+1} E_t, \ldots r_n E_j$ such that
$$|\{r^t \in R^t \left| \left(r_1^t, \ldots, r_{i-1}^t, r_{i+1}^t, \ldots, r_n^t\right), = \left(\mathrm{PK}_1, \ldots \mathrm{PK}_{i-1}, \mathrm{PK}_{i+1}, \ldots, \mathrm{PK}_n\right)\}\right| > \max \quad (6)$$

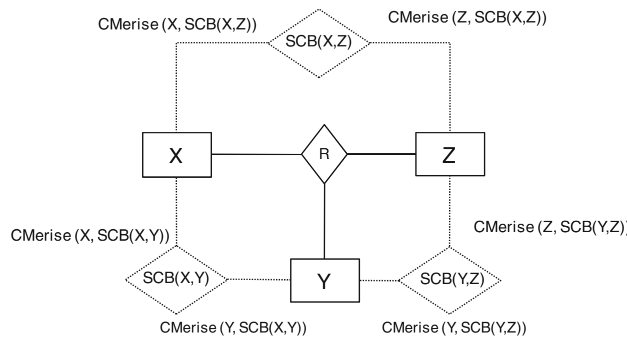


Fig. 8 Semantically constraining binaries related to R

While the valid cardinality definition is easy, its proof is not trivial. This work draws from related approaches such as Camps [13], Jones and Song [38] and McAllister [41]. These approaches are based on EntityLook and their rules have been adapted to the present proposal for ternary relationships. The adapted rules are based on the following notations:

Considering a ternary relationship $R = (X = r_1 E_1, Y = r_2 E_i, Z = r_3 E_j, A_1, \ldots, A_s)$, we denote by $CChen_{\max}(X, R)$, $CChen_{\min}(X, R)$, $CMerise_{\max}(X, R)$, and $CMerise_{\min}(X, R)$ the maximum and minimum cardinality of *CChen(X,R)* and *CMerise(X,R)*, respectively. Furthermore, semantically constraining binaries are needed to validate ternary relationship constraints (the definition appears in Jones and Song [38]). These implicit binary relationships will be denoted by *SCB (X,Y)*. A graphical representation is presented in Fig. 8 to facilitate a greater understanding of the rules notation.

The cardinality constraints associated with R must fulfill the following rules:

Rule 1: **(a)** $CMerise_{\max}(X, SCB(X, Y)) \geq CChen_{\max}(Y, R)$
(b) $CMerise_{\min}(X, SCB(X, Y)) \geq CChen_{\min}(Y, R)$

Rule 2: **(a)** $CMerise_{\max}(X, R) \geq CMerise_{\max}(X, SCB(X, Y))$
(b) $CMerise_{\min}(X, R) \geq CMerise_{\min}(X, SCB(X, Y))$

Rule 3: $CMerise_{\min}(X, R) \geq CMerise_{\min}(X, SCB(X, Y)) x CChen_{\min}(Z, R)$

Rule 4: $CMerise_{\max}(X, SCB(X, Y)) x CMerise_{\max}(Y, SCB(Y, Z)) \geq CMerise_{\max}(X, R)$

Rule 5: $CMerise_{\max}(X, SCB(X, Y)) \geq CMerise_{\min}(X, SCB(X, Y))$

Rule 6: $CMerise_{\min}(X, SCB(X, Y)) x CMerise_{\max}(Y, SCB(Y, Z)) \geq CMerise_{\min}(X, R)$

Rule 7: $CMerise_{\min}(X, SCB(X, Y)) x CMerise_{\max}(X, SCB(X, Z)) \geq CMerise_{\min}(X, R)$

Rule 8: $CMerise_{\max}(X, R) \geq CChen_{\max}(Z, R) + CMerise_{\min}(X, SCB(X, Y)) - 1$

Rule 9: If $CChen_{\min}(Z, R) > 1 \rightarrow CMerise_{\min}(X, SCB(X, Z)) > 1$ and $CMerise_{\min}(Y, SCB(Y, Z)) > 1$

Rule 10: $C(X, SCB(X, Y)) = 0$ iff $C(X, R) = 0$

According to McAllister [41], among these rules exists a minimal set—namely rules 1, 2, 4 and 10—that proves the structural validity of the relationship R. In addition, the total number of cardinalities in an n-ary relationship with n roles is given by $3^n - 2^{n+1} + 1$. For ternary relationships, the total number of cardinalities is 12, proving that the approach does not generate redundancies. The definition of rules provides a tool for the designer to verify cardinality constraints for each ternary relationship and may also help detect other cardinality constraints not noticed by the designer or even the domain expert.

In this way, the rules may aid the designer in her/his work since, through the solution of constraints implicitly associated with binary relationships (represented in Fig. 8 by a dotted line) that are easier to detect or to ask the domain expert about, certain cardinalities for the ternary relationship may also be deduced. In the exam-ple presented in Fig. 7, it might be simpler to ask how many techniques an employee uses. If the domain expert were to respond that there are 4 techniques, at best, then $CMerise_{\max}(Employee, SCB(Employee, Techniques)) = 4$ and, by the application of rule 1(a), the Chen cardinality for *Techniques* cannot be greater than 4 (Rules 1(a): $CMerise_{\max}(Employee, SCB(Employee, Techniques)) \geq CChen_{\max}(Techniques, R)$). Semantically, this implies that an employee for a given project cannot use more than 4 distinct techniques. Once the rule has been applied, it is easier to ask the domain expert, setting the employee and project values (since these are associated with the techniques) and, in agreement with the domain specifications, indicate that an employee uses 2 techniques at most in each project, just as is indicated in Fig. 7. With respect to the range of possible values in which this cardinality could lie, it is limited to a value less than or equal to 4, aiding the designer's work in understanding the domain through simple questions for the expert (i.e., more intuitive questions inasmuch as they deal with binary relationships) as well as in the completeness of the semantics of the ternary relationship.

4 Incorporating cardinality constraint semantics into a methodological framework

Conceptual modeling is very important in the development of information systems and, therefore, also in relational database development. In the previous section, a method was presented for the calculation of cardinality constraints in binary and ternary relationships, as well as the solution of shortcomings in the capture of the semantic constraints required in a database. Therefore, the proposal provides a schema with more semantics, bringing it closer to the real world or UD. While resources are provided to arrive at good designs, the designer must nevertheless know how to use these resources. Validation rules aid in the completion of the design and the fulfillment of the validation schema.

The validation of schemata is quite important inasmuch as, with a correct schema, consistent information is presented in the first phase. This provides a designer with a firm footing as she/he proceeds to the next phase in the method (i.e., the relational phase).

The principal difficulty when a conceptual schema is transformed into a relational schema is one of maintaining consistent information. Generally, it is quite difficult to achieve a complete mapping between both models, as well as maintain their structural and semantic restrictions when moving from the conceptual to the relational model. Restrictions that cannot be applied in the relational model must usually be reflected outside the database management system (DBMS) in application programs in several different ways [13]. Regarding this particular issue, a number of extensions of the relational model have been proposed [17,18,51] that provide a more semantically-oriented model.

Principal rules for transformation are described in most database textbooks [24,25,30,39,46,52]. This section will discuss the transformation of relationships in the relational schema, the main focus of this paper.

The correct transformation of schemata and the constraints expressed therein is necessary in order to preserve their intended meaning. Although the standard relational model [17] was initially unable to sufficiently reflect the semantics that could be present in a conceptual schema, it has been enhanced with specific elements used to preserve these original semantics. New Structured Query Language (SQL) versions have been developed and new integrity constraints have been defined [23,55] so as to offer additional information regarding previously existing SQL integrity constraints. In what follows below, the present proposal applies the basic rules of transformation to map relationships in relations and adds CHECKS and ASSERTIONS to complete the semantics where necessary. The approach is based on SQL 2003 [23]. The transformation of binary relationships will be examined first, followed by that of ternary relationships.

4.1 Binary relationship transformation

The cardinality constraints associated with binary relationships are those of entity participation and CMerise

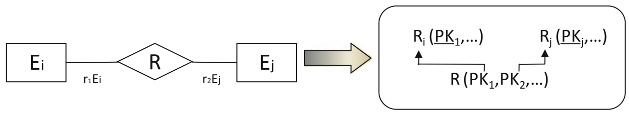


Fig. 9 Canonical form for the transformation of binary relationships

Table 1 Transformation options and structural relational constraints in binary relationships

Cases	$CMerise_{max}$	Structural constraints
0	All > 1	*PK* is (PK_1, PK_2)
1	$CMerise_{max}(r_1E_i, R) = 1$	*PK* is (PK_1)
2	$CMerise_{max}(r_1E_i, R) = 1$	*PK* is (PK_1)
	$CMerise_{max}(r_2E_j, R) = 1$	$UK = \{(PK_2)\}$

cardinality. The first of these establishes whether all entity instances participate in the relationship or if there are entity instances that do not participate, denoted by mandatory or optional participation. The second shows the number of times that the entity key instance appears in the relationship for each entity in the relationship. For the transformation of binary relationships, the canonical form is presented in Fig. 9 below.

For this transformation, the structural relational constraints are applied according to the maximum Merise cardinality as indicated in Table 1 below.

According to whether the entity participation is mandatory—the *CMerise*$_{max}$ $\neq$ '*n*' and *CMerise*$_{min}$ $>$ 1—the following patterns must be applied for the completion of the semantics (for both roles depending on their cardinalities):

Pattern 1

```
CREATE ASSERTION
C(r1Ei,R)_Mandatory
CHECK (NOT EXITS (SELECT PK1
       FROM Ei WHERE PK1
       NOT IN (SELECT PK1
       FROM R)));
```

Pattern 2

```
CREATE ASSERTION
CMerisemin(r1Ei,R)
CHECK ((CMerisemin(r1Ei,R) ≤
SELECT MIN
(NUMBER_MIN) FROM (SELECT
COUNT(*) FROM R
GROUP BY PK1) AS NUM-
BER_MIN));
```

Pattern 3

```
CREATE ASSERTION
CMerisemax(r1Ei,R)
CHECK ((CMerisemax(r1Ei,R) ≤
SELECT MAX
(NUMBER_MAX) FROM (SELECT
COUNT(*) FROM R
GROUP BY PK1) AS NUM-
BER_MAX));
```

Table 2 Particular cases in the transformation of binary relationships

Particular cases	Cardinality constraint represented
R_i (<u>PK</u>$_i$,...) R_j (<u>PK</u>$_j$,...,PK$_i$)	$CMerise_{max}(r_2E_j, R) = 1$
	$C(E_j, R) = 1$
R (<u>PK</u>$_i$, ... PK$_j$ ...)	$CMerise_{max}(r_1E_i, R) = 1$
	$CMerise_{max}(r_2E_j, R) = 1$
$UK = \{PK_j\}$	$C(E_i, R) = 1$
	$C(E_j, R) = 1$

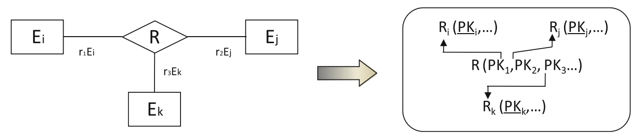


Fig. 10 Canonical form for the transformation of ternary relationships

It is necessary to mention two particular cases here. A binary relationship can also be transformed into two relations (i.e., one for each entity) or one relation that includes the information of the two entities. These occur when some *CMerise*$_{max}$ is equal to one at least in one entity. The last column of Table 2 shows the cardinality constraints reflected through the structural transformation.

The proposal here presented extends the number of cases presented in Fahrner and Vossen [25] and discusses every cardinality constraint approach, including those of the Merise method and Chen's approach.

4.2 Ternary relationship transformation

In this case, it is assumed that a particular ternary relationship is valid and cannot be represented through binary relationships. Most database book references [24,25,28,31] propose ternary relationship transformation as a new relation whose primary key is made up of the primary key of each associated entity. To complete the transformation rules, Fig. 10 presents the canonical form to be used as the starting point for the explanation of different transformations, depending on cardinality constraints. According to the proposal, a ternary relationship has three types of cardinality constraints; to wit, entity participation, *CMerise* and *CChen*. Therefore, a further set of assertions must be defined in order to check the Chen cardinalities.

Unlike binary relationships, not all combinations of *CMerise* and *CChen* cardinalities are feasible. For example, if the $CMerise_{max}(r_1E_i, R)$ = '*n*' (i.e., non-restricted value), it is implied that $CChen_{max}(r_2E_{j,} R)$ and $CChen_{max}(r_3E_{k,} R)$ must also be '*n*' through the application of the validation rules presented in the previous section. Thus, seven options have been distinguished in ternary relationship transformations. Table 3 presents the primary key and unique keys to be added

Table 3 Transformation options and structural relational constraints in ternary relationships

Cases	$CChen_{\max}$	$CMerise_{\max}$	Structural constraints
0	All > 1	All > 1	*PK* is (PK_1, PK_2, PK_3)
1	$CChen_{\max}(r_1E_i, R) = 1$	All > 1	*PK* is (PK_2, PK_3)
2	$CChen_{\max}(r_1E_i, R) = 1$	All > 1	*PK* is (PK_1, PK_3)
	$CChen_{\max}(r_2E_j, R) = 1$		*UK* is (PK_2, PK_3)
3	$CChen_{\max}(r_1E_i, R) = 1$	All > 1	*PK* is (PK_1, PK_2)
	$CChen_{\max}(r_2E_j, R) = 1$		*UK* = $\{(PK_1, PK_3), (PK_2, PK_3)\}$
	$CChen_{\max}(r_3E_K, R) = 1$		
4	$CChen_{\max}(r_2E_j, R) = 1$	$CMerise_{\max}(r_1E_i, R) = 1$	*PK* is (PK_1)
	$CChen_{\max}(r_3E_K, R) = 1$		*UK* = $\{(PK_1, PK_3), (PK_2, PK_3)\}$
5	$CChen_{\max}(r_1E_i, R) = 1$	$CMerise_{\max}(r_1E_i, R) = 1$	*PK* is (PK_1)
	$CChen_{\max}(r_2E_j, R) = 1$		*UK* = $\{(PK_1, PK_3), (PK_2, PK_3)\}$
	$CChen_{\max}(r_3E_K, R) = 1$		
6	$CChen_{\max}(r_1E_i, R) = 1$	$CMerise_{\max}(r_1E_i, R) = 1$	*PK* is (PK_1)
	$CChen_{\max}(r_2E_j, R) = 1$	$CMerise_{\max}(r_2E_j, R) = 1$	*UK* is (PK_2)
	$CChen_{\max}(r_3E_K, R) = 1$		
7	$CChen_{\max}(r_1E_i, R) = 1$	$CMerise_{\max}(r_1E_i, R) = 1$	*PK* is (PK_1)
	$CChen_{\max}(r_2E_j, R) = 1$	$CMerise_{\max}(r_2E_j, R) = 1$	*UK* = $\{(PK_2), (PK_3)\}$
	$CChen_{\max}(r_3E_K, R) = 1$	$CMerise_{\max}(r_3E_K, R) = 1$	

to the relation *R* according to the maximum cardinality constraints.

To complete the transformation rules, the assertions must be created with the same criteria as that used for binary relationships. The patterns (from 1 to 5) will be applied for each mandatory entity participation, for each $CMerise_{\min}(rE, R)$ or $CChen_{\min}(rE, R) > 1$ and $CMerise_{\max}(rE, R)$ or $CChen_{\max}(rE, R) \neq$ 'n'.

Pattern 4
```
CREATE ASSERTION
 CChenmin(r1Ei,R)
CHECK ((CChenmin(r1Ei,R)
 ≤ SELECT MIN
 (NUMBER_MIN)
FROM (SELECT COUNT(*)
 FROM R GROUP BY
 PK2,PK3)
AS NUMBER_MIN));
```
Pattern 5
```
CREATE ASSERTION
 CChenmax(r1Ei,R)
CHECK ((CChenmax(r1Ei,R)
 ≤ SELECT MAX
 (NUMBER_MAX)
FROM (SELECT SELECT
 COUNT(*) FROM R GROUP
 BY PK2,PK3)
AS NUMBER_MAX));
```

The handling of the rules for this approach can be clarified through an example and should explain all steps necessary

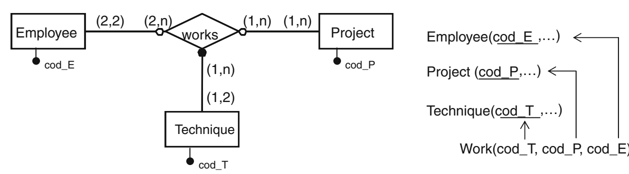


Fig. 11 An example of ternary relationship transformation

for the preservation of ternary relationship semantics. This example has already been explained in Sect. 3 on cardinality constraints; however, here the focus is on the application of rules in accordance with Table 3 and the aforementioned assertions.

Figure 11 shows the canonical transformation associated with the ternary relationship *works*. Case 0 is applied since all maximum cardinality constraints are greater than it (see Table 3). The primary key is (*cod_T, cod_P, cod_E*) in the *Work* relationship. In a following step, assertions are added depending upon cardinality constraints. The *Technique* entity has mandatory participation in *works* and to preserve this constraint, the following assertion must be added (i.e., Pattern 1 is applied):

```
CREATE ASSERTION C(Technique, R)_Mandatory
CHECK (NOT EXITS (SELECT cod_T FROM
      Technique WHERE cod_T NOT IN
      (SELECT cod_T FROM Work)));
```

To check the minimum Merise cardinality in the *Employee* relation, Pattern 2 is applied ensuring that the *cod_E* value appears in the *Work* relation at least twice:

```
CREATE ASSERTION CMerise_min(Employee, R)
CHECK ((2 ≤ SELECT MIN (NUMBER_MIN) FROM
        (SELECT COUNT(*) FROMWork GROUP BY
        cod_E) AS NUMBER_MIN));
```

There are two Chen cardinalities that limit the population in the *Work* relation and they must be checked through the application of Patterns 4 and 5:

```
CREATE ASSERTION CChen_min(Employee, R)
CHECK ((2 ≤ SELECT MIN (NUMBER_MIN) FROM
  (SELECT COUNT(*) FROM
Work GROUP BY cod_T, cod_P)
  AS NUMBER_MIN)); CREATE ASSERTION CChen_max
    (Employee, R) CHECK ((2 ≥ SELECT MAX
(NUMBER_MAX) FROM (SELECT COUNT(*) FROM
  Work GROUP BY cod_T, cod_P) AS NUMBER_MAX));
```

These assertions are created to ensure that a project using a particular technique must have two and only two employees allocated to that technique. The other business rule is that a single employee allocated to a project may use one or two techniques. The following assertion ensures its fulfilment (Pattern 5):

```
CREATE ASSERTION CChen_max(Technique, R)
CHECK((2 ≥ SELECT MAX (NUMBER_MAX) FROM
(SELECT COUNT(*) FROM Work GROUP BY cod_E,
cod_P) AS NUMBER_MAX));
```

Following the application of the transformation rules, it may be concluded that the maximum cardinality constraints in one relationship establish the relation structure, that is, the relation definition. For this reason, the first step in mapping the relationship to relation is to assign the respective case according to Table 3 and then to add the assertion associated with constraints about the population of R.

5 A practical experience managing ternary relationships

The present proposal includes cardinality constraint definitions in order to avoid confusion and ambiguities in the design process. To do this, the proposal sets the terminology and method for calculating cardinality constraints, thereby making the design task easier. This section presents experimental evidence to evaluate particular aspects of the proposal including (a) the correct detection of specifications, (b) confidence in use and (c) accuracy in the representation of specifications. Experiments were carried out with designers who have experience handling several different conceptual models and using various approaches to calculate cardinality constraints.

Purpose

The aim of the experiment is to verify if the proposed approach and notation accomplish the goal for which they have been defined, that is, whether approach C, due to its clearer semantics and despite its seemingly greater complexity, may nevertheless allow designers to more easily detect cardinalities through the specifications indicated in the questionnaire (designers are given the abstraction process).

The experiment is designed to compare both the interpretation and representation of cardinality constraints according to three different approaches or techniques: Chen, Merise and the present proposal. The main questions posed for specific study are

- *How many specifications are well-detected for each method?* The answer to this question is attained by calculating the number of specifications from the experimental text that participants properly detected for each method.
- *Which method is easier to use?* The question is analyzed according to the participants' level of confidence in their answers and the correctness of the representation and interpretation of the cardinality constraint.

Methodology

- *Research Model* The present study focuses on the EER data model, since the main concern here is the representation and interpretation of cardinality constraints in ternary relationships. Three methods of representation and interpretation are presented.
- Independent variables:
  - *Data model* The participant applies each of the three methods to represent and interpret the semantics of cardinality constraints. The experiment is focused particularly on the cardinality constraint associated with ternary relationships since these constraints are ambiguous despite there being different methods for their representation and interpretation. The three approaches presented are labeled A, B and C which correspond to Chen, Merise and our proposal, respectively.
  - *Task factor* Two main tasks are directed towards performance. The first task is focused on the detection of how the participant associates the domain specifications depending on the approach used (i.e., A, B or C) and her/his level of confidence. The second task concerns cardinality constraint representation according to the three methods.

- *Dependent variables* The first of these variables is correctness. The correctness applied to modeling will be measured through cardinality constraints defined with respect to ternary relationships. For each method, the participant is scored according to the comparison between the real solution and his or her own solution, taking into account the number of specifications represented in the schema, the level of confidence in the participant's solution and the cardinality constraint representation.

Hypotheses

The hypotheses to be taken into account by the study are

- H1: There will be no significant difference between approaches A, B and C regarding the number of specifications represented in the schema;
- H2: There will be no significant difference between approaches A, B and C regarding the level of participant confidence in their use;
- H3: There will be no significant difference between approaches A, B and C regarding the modeling of cardinality constraints.

Strategy

Prior to the experiment, a session was held with the participating designers to explain these three different methods for representing and calculating cardinality constraints in ternary relationships. The first two of these were approaches A and B, the Chen approach [16] and Merise method [50], respectively, that is, the most popular methods for determining cardinality constraints. The third was method C, the cardinality constraints technique proposed here, consisting of a combination of Entity participation, *CChen* and *CMerise*.

The material was conveyed in general terms, although some examples were given to ensure a better understanding of each approach. The designers were familiar with the traditional approaches, Chen (approach A) and Merise (approach B), but not with the terminology. The main objective for this session, therefore, was to equip the participants with this related terminology.

The informational session took place one half hour prior to the execution of the experiment. The 30 test participants (see description below) were called and advised that during the session they would be able to ask as many questions as they pleased, while not being able to ask any at the time of the actual experiment.

A questionnaire, divided into two main parts, was given following the informational session (see Appendix A). The first part consisted of six domain specifications for a ternary relationship to be modeled by the cardinality constraint methods. The second part had three subsections, one for each method. For each method, the designer had to respond whether one specification could be represented and how confident he/she was in the correctness of his/her answer, as well as to give the cardinality constraints representation.

Table 4 Summary of participant characteristics

	Participant characteristics
Years of data modeling experience (mean/stdev)	6.41 (2.85)
Years of design teaching experience (mean/stdev)	4.20 (1.5)
Number of Data Models	two (18)
	greater than two (12)
Practical Expertise	three years (20)
	Every participants with Chen approach
	greater than three (10)
	5 with Chen approach and 5 with Merise approach

Experimental results

Subject demographics Subjects consisted of 30 expert designers, all of whom being teachers of topics related to database design. The design teachers also have experiences in the industry as database consultants. All of them are familiar with and use more than two conceptual models. The participants were chosen among 33- to 45-year-old graduates of the Carlos III University of Madrid. Thus, it could be established that each subject possessed sufficient training in ER modeling and that their design experience included the use of other models (e.g., UML or the Merise data model). A summary of relevant participant characteristics is shown in Table 4 below.

The results for determining whether differences were to be found between methods are presented here below.

Number of specifications Hypothesis 1 deals with the number of specifications represented for each method. This means that for each method and specification, the participant may indicate, using numbers from 1 to 6, his/her level of confidence that the specification has been well detected with the method or, alternatively, by abstaining from the use of numeric indicators in order to indicate the inapplicability of the method.

Due to the nature of the sample, Fisher's least significant difference test (a simple multiple range test) has been used. The mean test statistically proves the difference between samples (Table 5). Samples are denoted by N_esp_A, B or C which represents the number of well-detected specifications for each method as judged by study participants. Samples

Table 5 Screen shot of multiple range tests for the number of the specifications captured for each method

Multiple range tests		
Method: 95.0 percent LSD		
	Mean	Homogeneous groups
N_esp_A	2.83333	X
N_esp_A	3.66667	XX
N_esp_B	5.16667	X
Contrast	Difference	+/− Limits
N_esp_A–N_esp_B	0.833333	1.71107
N_esp_A–N_esp_C	−1.5	1.71107
N_esp_B–N_esp_C	∗ − 2.33333	1.71107

The statistical tool uses the European convention of a comma for the decimal point

* Statistically significant difference.

This table applies a multiple comparison procedure to determine which means are significantly different from which others. The bottom half of the output shows the estimated difference between each pair of means. An asterisk has been placed next to 1 pair, indicating that this pair shows a statistically significant difference at the 95.0% confidence level. At the top of the page, 2 homogenous groups are identified using columns of X's. Within each column, the levels containing X's form a group of means within which there are no statistically significant differences. The method currently being used to discriminate among the means is Fisher's least significant difference (LSD) procedure. With this method, there is a 5.0% risk of calling each pair of means significantly different when the actual difference equals 0

were run through a normal distribution test prior to the application of the Fisher LSD test.

According to the data shown in Table 5, the average contrast between these samples indicates that method B shows a statistically significant difference, at the 95% confidence level, with method C. The number of specification differences seen between approach B and C is due to the fact that approach B is the lowest in the representation of specifications. Explained in another way, there are six specifications and with approach B designers represented 2.8 of them, on average, taking into account those which both can and cannot be applied with this method (i.e., of the six specifications, approach B can be applied to only three, leaving three others which should not be indicated with numeric evaluations by the participants). Thus, of the six specifications, approach B is only applied well in 2.8. This stands in contrast with approach C for which participants detected 5.1.

In general, the rest of the average comparisons are not statistically significant. It can be concluded that while study participants had no previous familiarity with approach C (i.e., it had only just been explained to participants during the training session), they were nevertheless able to correctly apply it. In other words, the participants were able to distinguish between which specifications were and were not able to be represented with each method (Table 5).

Table 6 Average in self-confidence level in answers for each approach

Method	Approach	Average
A	Chen	3.91
B	Merise	2.94
C	Proposal	4.18

Table 7 ANOVA table to study the differences in self-confidence level between methods

ANOVA Table					
	Analysis of variance				
Source	Sum of squares	Df	Mean square	F-ratio	P-value
Between groups	5.07003	2	2.53502	2.05	0.1633
Within groups	18.5487	15	1.23658		
Total (Corr.)	23.6188	17			

The statistical tool uses the European convention of a comma for the decimal point

The ANOVA table decomposes the variance of the data into two components: a between-group component and a within-group component. The F-ratio, which in this case equals 2.05002, is a ratio of the between-group estimate to the within-group estimate. Since the P-value of the F-test is greater than or equal to 0.05, there is not a statistically significant difference between the means of the 3 variables at the 95.0% confidence level

Level of confidence Participant confidence has been measured here using a Likert scale with six possible levels. When a participant detected and represented a cardinality constraint associated with one text specification, she/he was required to measure how confident she/he was in this assessment. A participant's level of confidence could be expressed with a number value between one and six.

More specifically, the participant was asked to read a specification or requirement and then indicate with a number between 1 and 6 in the appropriate space of the questionnaire the degree to which she/he was confident that the specification could be represented by the indicated approach. In this case, all of the values indicated in the questionnaire were taken into account regardless of whether the specification was properly represented by each method. The objective of these confidence level measurements was to determine the method with which the participants were most familiar.

The average for confidence l evels f or e ach s ample is shown in Table 6. According to the results, method B presents a lowest confident level than the others methods, but the ANOVA test (Table 7) indicated that there is no statistically significant difference between groups.

These results suggest to the authors that designers either do not understand this approach or have some doubts in its use. Contrary to what the authors had expected, approach C in the representation of cardinality constraints was seen by study participants as similar to the other approaches with which they were more familiar. This reinforces the hypothesis

Table 8 Comparison of self-confidence for each specification

	1	2	3	4	5	6
Method A						
Correct solution	6	0	0	6	0	6
Average of self-confidences	4	3	0	6	4	5
Method B						
Correct solution	0	6	6	0	6	0
Average of self-confidences	3	2	2	3	3	4
Method C						
Correct solution	6	6	6	6	6	6
Average of self-confidences	5	6	4	5	6	5

Table 9 Real solution with experimental solution comparison

Comparison of means	Confidence interval
Real solution-A	−3.12035 to 0.78702
Real solution-B	−3.89327 to 4.22661
Real solution-C	−0.0433447 to 0.422661

advanced by the authors here that some design confusion is caused by the fact that syntax and semantics are not set in the elements of the conceptual model.

The main conclusion to be drawn from the experiment is that the designers felt comfortable with the method proposed and learned it easily, given that their level of confidence in approach C was higher than those reported for the other approaches with which they had much greater familiarity.

Correctness Correctness is measured as a comparison of the real solution and the solution proposed by designers according to the definition given in Moody and Shanks [42], first taking into account whether, for each approach, the participants correctly carried out the abstraction process (what specifications can be represented with each method) and, second, whether these were represented well.

In the table presented in Table 8, the first row shows the real solution for each specification, while the second row displays the average participant confidence score taken for each specification. Correctness is measured as the difference between the first and second rows. A score of zero indicates that the corresponding participant solution is correct, while negative values indicate that some participants believed that a specification could be represented by an incorrect method. As can be observed, in approaches A and B, some of the participants appear to have committed errors while attempting to represent certain specifications. For approach A, this was the case with specifications 2 and 5, while for approach B, this was the case with specifications 1, 4 and 6.

In accordance with the confidence intervals represented in Table 9, no empirical evidence exists that one particular approach is better represented than another insofar as 0 is also included in these intervals. Nevertheless, observing the

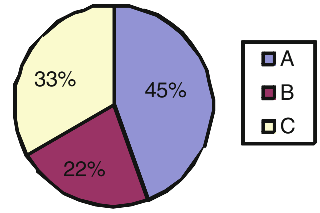


Fig. 12 The percentage of the correctly represented constraints

wide varieties of values for approach B, it is clear that this approach is the most variable among the three.

With respect to the representation of cardinalities in ternary relationships and as shown in Fig. 12, for each of the three approaches, the percentage of correctly represented constraints was calculated following the assignment of the number 1 to all correct representations and 0 to all incorrect representations.

Analysis and discussion

The experiment was carried out with 30 participants, all of whom being either database teachers or database consultants. Despite this common professional background, however, it was not possible to fully eliminate bias from the study due to the different experiences and abilities of each participant. This fact notwithstanding, certain conclusions may still nevertheless be drawn.

Regarding specifications, method C is that which detects the greatest number. In general, these results ought to be taken with a certain degree of caution since the participants can intuit the general nature of the experiment. As a result, they attempt to represent all of the specifications presented in the text, independently of the method. In other words, the participants do not know to reject the specifications which cannot be detected with methods A or B.

While always taking into account the possibility of subject bias as discussed in the previous paragraph, in terms of participant confidence levels, results show method C to be similar to method A, and method B to have obtained the worst results of the three. With regard to the difference in confidence observed for the methods A and B, the authors believe this to be attributable to the participants' greater familiarity and experience with the former. As for the proposed method C, the recorded confidence level was quite high, particularly when considering participants' complete lack of previous experience with the method. These results are likely due to the fact that method C clearly establishes what 'relationship' or 'association' mean, allowing it to facilitate the detection of constraints. As discussed in Sect. 2 of the present study, these characteristics of the proposed approach are not necessarily present in the other approaches.

Finally, method C presents a level of correctness slightly lower than that of method A; however, the representation of cardinalities is more correct with the latter.

Despite the preliminary nature of the study, a number of conclusions may nevertheless be drawn regarding the proposed method and in comparison with the other methods observed.

These results make clear that the abstraction process to detect which specification could be collected with each approach was not properly carried out by participants. While the present study did not go to lengths to study the causes of this fact, it is possible that lack of experience in the detection and use of ternary relationships could at least partially be blamed.

It is also clear that method B was not well understood by participants, given its significantly lower participant confidence score. Method C, however, was correctly applied and enjoyed a high level of participant confidence. Furthermore, method C appears to behave in a manner quite similar to that of method A.

The results show that the solution arrived at by approach C is the closest to the real solution while the solution arrived at by approach B is the farthest.

6 Conclusions and future work

Conceptual models are used in the first phases of database development in which requirement specifications are acquired and then represented in a conceptual schema which is simple and easily understood by domain experts. When these initial phases are finished, the conceptual schema is usually transformed to a relational schema to be implemented in a relational database management system.

The most common properties of conceptual models are that they should be easy to learn, intuitive, independent and powerful enough to represent any domain. However, the authors have noted some associated problems [19]. Sometimes, for instance, conceptual models mistake simplicity for a limitation of expressivity. For example, binary models are simple, but may not represent some domain specifications. Another problem is that of how to handle conceptual elements when these are not clearly defined. This generates ambiguity for the user and when the element is the relationship and its cardinality constraints, problems worsen.

To avoid ambiguity, this proposal formally defines the syntax and semantics of relationships and cardinality constraints. Relationship definitions and cardinality constraints in the most often used conceptual models have been studied. Types of cardinality constraints provide integration and comparison between schemata of different conceptual models [35]. This study provides results on the cardinality constraints in ternary relationships. Since ternary relationships do not reflect all their properties through cardinality constraints, it is necessary to add more expressivity to the cardinality constraints, as well as to clarify the concept.

The method to manage the ternary relationship semantics separates cardinalities into two types: first, the participation concept is considered to be a concept associated with instances of the entity and second, instance combinations in the relationship. The entity cardinality defines the optional or mandatory participation of entity instances associated with the relationship. Furthermore, cardinality constraints associated with the relationship are defined to reflect how combined components belong to relationship instances. Cardinalities are denoted *CMerise* or *CChen* depending on whether one or $n-1$ components are set to calculate the number of different combinations in relationship instances.

A set of structural validation rules has also been presented to help in the detection of mistakes made by designers when defining cardinality constraints in ternary relationships. Finally, a preliminary empirical study has demonstrated that the proposed approach, in comparison with other approaches, facilitates the detection and representation of ternary relationship.

For use in the logical phase in the database development methodology, an exhaustive set of rules has been presented to transform the cardinality constraints proposed in this work. Cardinality constraints offer a dynamic capability that involves the definition of ASSERTIONS to avoid semantics when an updating operation occurs in the database. Integrity constraints used in our proposal are based on SQL 2003 [23].

Some ongoing work in the field focuses on the implementation of these assertions through event-condition-action (ECA) rules. In Al-Jumaily et al. [2] a set of triggers associated with cardinality constraints is proposed, as well as an execution model that ensures that termination avoids the cycles of the execution. The Rational Rose CASE [47] was the tool selected for the incorporation of a trigger generator [4,5] which creates a complete relational script without the loss of semantics. Future work will focus on the efficiency of the object-relational database [3] with incorporated triggers.

Other tasks are aimed at helping designers by means of tools incorporated in a CASE tool. This is an important topic given that the abstraction process is not an easy task. Semantics validations in binary relationship are presented in Martínez et al. [40]. A natural language dialog system is developed to provide designers with an easy interface which will provide cardinality constraints according to this conversation. A further objective is an extension of this work to ternary relationships. The incorporation of an Intelligent Tutor to teach database design has been explained previously in Castro et al. [14] and Iglesias et al. [36]. This Tutor will apply the practical method presented in this work to facilitate the cardinality constraints definition, as well as its representation in a conceptual schema. As part of this work, another recent study [19] has been carried out. This study presents experimental results to apply the present method to database design students in a Computer Science bachelor's degree program using a web platform.

Acknowledgments This work forms part of the 'Thuban: Natural Interaction Platform for Virtual Attending in Real Environments' project (TIN2008-02711), the Spanish Ministry of Industry, Tourism and Trade through the project Semants (TSI-020100-2009-419) and also by the Spanish research projects: MA2VICMR: Improving the access, analysis and visibility of the multilingual and multimedia information in web for the Region of Madrid (S2009/TIC-1542).

Appendix A: Experimental materials

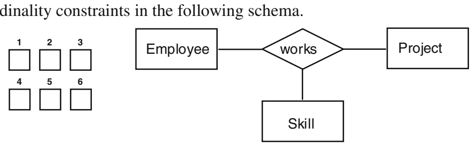
Participants in the experiment completed the following questionnaire:

Domain Specifications

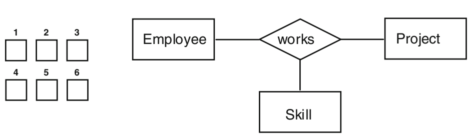
1. Company employees who work on projects use at least two skills, but there are employees who work on projects that do not require skills.
2. The company forbids employees from using more than three skills in their projects.
3. The company can only have 30 projects requiring skilled employees.
4. One employee has a skill that is uniquely associated with one project.
5. Every skill is related to one employee in one project.
6. One employee can use the same skill in different projects.

Model of the ternary relationship and confidence level for each method

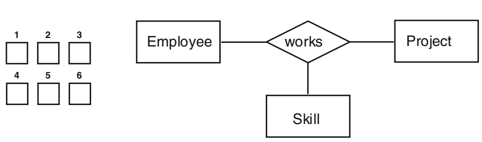
A Directions: Indicate in the square of the specification whether approach A can be represented, and the confidence level for your answer (1-DK/NA, 2-None, 3-Low, 4-Middle, 5- High, 6- Complete). Represent specifications by means of cardinality constraints in the following schema.



B Directions: Indicate in the square of the specification whether approach B can be represented and the confidence for your answer (1-DK/NA, 2-None, 3- Low, 4-Middle, 5- High, 6- Complete). Represent specifications by means of cardinality constraints in the following schema.



C Directions: Indicate in the square of the specification whether approach C can be represented and the confidence level for your answer (1-DK/NA, 2-None, 3-Low, 4-Middle, 5- High, 6- Complete). Represent specifications by means of cardinality constraints in the following schema.



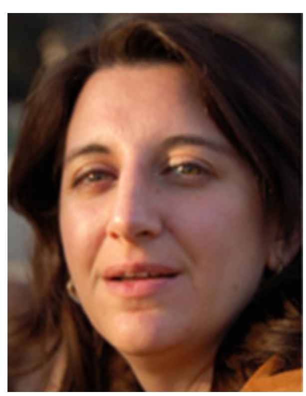
References

1. Abrial, J.R.: Data Semantics, pp. 1–59. North-Holland, Amsterdam (1974)
2. Al-Jumaily, H., Cuadra, D., Martinez, P.: The execution model to transform the associate semantics of cardinality constraints in a relational DBMS. In: Proceedings of the 4st International Conference on Enterprise Information Systems, Ciudad Real, Spain, April 2–6, pp. 819–822 (2002)
3. Al-Jumaily, H., Cuadra, D., Martinez, P.: Applying a fuzzy approach to relaxing cardinality constraints. In: 15th International Conference on Database and Expert Systems Applications, Lectures Notes, Zaragoza (Spain), pp. 654–662, September 2004
4. Al-Jumaily, H., Cuadra, D., Martinez, P.: PANDORA CASE Tool: triggers generating for cardinality constraints in RDBMS. In: Proceedings of the IADIS International Conference WWW/Internet 2003, Algarve (Portugal), pp. 999–1002, November 2003
5. Al-Jumaily, H., Cuadra, D., Martinez, P.: Plugging Active Mechanisms to Control Dynamic Aspects Derived From The Multiplicity Constraint In UML. The Workshop of 7th International Conference on the Unified Modeling Language. Doctoral symposium. Lisbon (Portugal), October (2004)
6. Badia, A.: Entity-relationship modeling revisited. SIGMOD Record **33**(1), 77–82 (2004)
7. Barke, R.: Case*Method: Entity Relationship Modelling. Addison Wesley, New York (1990)
8. Batini, C., Ceri, S., Navathe, S.B.: Conceptual Database Design: An Entity-Relationship Approach. Benjamin/Cummings, San Francisco (1992)
9. Batra, D., Zanakis, H.: A conceptual database design approach based on rules and heuristics. Eur. J. Inf. Syst. **3**(3), 228–239 (1994)
10. Batra, D., Antony, S.R.: Novice errors in conceptual database design. Eur. J. Inf. Syst. **3**(1), 57–69 (1994)
11. Bruce, T.A.: Designing Quality Databases with IDEF1X Information Models. Dorset House, New York (1992)
12. Calí, A.: Querying Incomplete Data with Logic Programs: ER Strikes Back. Parent, C., et al. (eds.) ER 2007, LNCS 4801, pp. 245–260 (2007)
13. Camps, R.: From ternary relationship to relational tables: a case against common beliefs. ACM/SIGMOD Record **31**, 46–49 (2002)
14. Castro, E., Cuadra, D., Martínez, P., Iglesias, A.: Integrating Intelligent Methodological Tutoring assistant in a CASE Platform: The PANDORA Experience, Informing Science + IT Education Conference 2002, Cork (Ireland), May 2002
15. Castro, E., Cuadra, D., Martínez, P.: An empirical perspective of using ternary relationship in database conceptual modeling. Inf. Educ. **2**(2), 191–200 (2003)
16. Chen, P.P.: The entity-relationship model: toward a unified view of data. ACM Trans. Database Syst. **1**(1), 9–36 (1976)
17. Codd, E.F.: A relational model of data for large shared data banks. Commun. ACM **13**(6), 377–387 (1970)
18. Codd, E.F.: Extending the database relational model to capture more meaning. ACM Trans. Database Syst. **4**(4), 397–434 (1979)
19. Cuadra, D., Iglesias, A., Castro, E., Martínez, P.: Educational experiences detecting, using and representing ternary relationships in database design. IEEE Trans. Educ. **53**(3), 358–364 (2010)
20. Cuadra, D., Martínez, P., Castro, E.: Relationship cardinality constraints in relational database design. Encyclopedia of Information Science and Technology. Idea Group Publishing, IV, pp. 2419–2424 (2005)
21. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do practitioners use conceptual modeling in practice?. Data Knowl. Eng. **58**, 358–380 (2006)
22. Dullea, I.: Song, IL.-Y.: An analysis of the structural validity of ternary relationships in entity-relationship modeling. In: Proceedings of the 7th International Conference on Information and Knowledge Management, pp. 331–339, November 1998

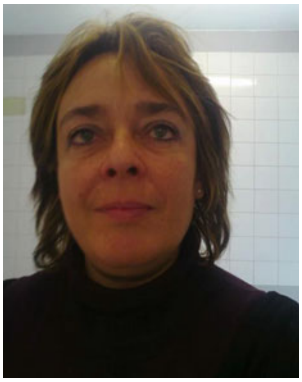
23. Eisenberg, A., Melton, J., Kulkarni, K., Michels, J.E., Zemke, F.: SQL: 2003 Has Been Published. SIGMOD Record, 33, 1, March 2004
24. Elmasri, R., Navathe, S.B.: Fundamentals of Database Systems, 5th edn. Addison-Wesley, New York (2007)
25. Fahrner, C., Vossen, G.: A survey of database design transformations based on the entity-relationship model. Data Knowl. Eng. **15**, 213–250 (1995)
26. Ferg, S.: Cardinality concepts in entity relationship. In: Proceedings of the 10th International Conference on the Entity Relationship Approach (1991)
27. Garcia-Molina, H., Ullman, J.D., Widom, J.: Database Systems: The Complete Book, 2nd edn. Prentice-Hall, Upper Saddle River (2008)
28. Gemino, A., Wand, Y.: Complexity and clarify in conceptual modeling: comparison of mandatory and optional properties. Data Knowl. Eng. **55**(3), 301–326 (2005)
29. Génova, G., Llorens, J., Martínez, P.: The meaning of multiplicity of N-ary associations in UML. J. Softw. Syst. Model. **1**(2), 86–97 (2002). http://www.sosym.org/
30. Halpin, T.: Information Modeling and Relational Databases: from Conceptual Analysis to Logical Design. Morgan Kaufman, San Francisco (2001)
31. Hansen, G., Hansen, J.: Database Management and Design. Prentice-Hall, Upper Saddle River (1995)
32. Harel, D., Rumpe, B.: Meaningful Modeling: What's the Semantics of "Semantics". IEEE Comput. **37**, 64–71 (2004)
33. Hartmann, H.: On implication problem for cardinality constraints and functional dependencies. Ann. Math. Artif. Intell. **33**, 253–307 (2001)
34. Hartmann, H., Link, S., Trinh, T.: Constraint acquisition for entity-relationship models. Data Knowl. Eng. **68**, 1128–1155 (2009)
35. He, Q., Wang Ling, T.: An ontology based approach to the integration of entity-relationship schemas. Data Knowl. Eng. **58**, 299–326 (2006)
36. Iglesias, A., Martínez, P., Cuadra, D., Castro, E., Fernandez, F.: Learning to teach database design by trial and error. ICEIS 2002, Proceedings of the 4st International Conference on Enterprise Information Systems, pp. 500–505, Ciudad Real, Spain, April 2–6, 2002
37. Ishakbeyoglu, N., Özsoyoglu, Z.M.: Maintenance of implication integrity constraints under updates to constraints. J. Very Large Database **7**(2), 67–78 (1998)
38. Jones, T.H., Song, IL-Y.: Binary equivalents of ternary relationships in entity-relationship modeling: a logical decomposition approach. J. Database Manag. **11**, 12–19 (2000)
39. Korth, H.F., Silberschatz, A., Sudarshan, S.: Database Systems Concepts, 5th edn. McGraw-Hill, New York (2005)
40. Martínez, P., De Miguel, A., Cuadra, D., Castro, E.: Data Conceptual Modeling through Natural Language: Identification and Validation of Relationship Cardinalities. Information Resources Management Association (IRMA 2000), Anchorage (EEUU), pp. 500–504, May 2000
41. McAllister, A.: Complete rules for n-ary relationship cardinality constraints. Data Knowl. Eng. **27**, 255–288 (1998)
42. Moody, D.L., Shanks, G.G.: Improving the quality of data models: empirical validation of a quality management framework. Inf. Syst. **28**, 619–650 (2003)
43. Parent, C., Spaccapietra, S., Zimányi, E.: Spatio-temporal conceptual models: data structures, space and time. In: Proceedings of the 7th ACM International Symposium on Advances in Geographic Information Systems, pp. 26–33 (1999)
44. Purchase, H.C., Welland, R., McGill, M., Colpoys, L.: Comprehension of diagram syntax: an empirical study of entity relationship notations. Int. J. Human-Comput. Stud. **61**, 187–203 (2004)
45. Ram, S., Khatri, V.: A comprehensive framework for modeling set-based business rules during conceptual database design. Inf. Syst. **30**, 89–118 (2005)
46. Ramakrishnan, R., Gehrke, J.: Database Management Systems, 3rd edn. MacGraw-Hill (2003)
47. Rational Web site. http://www.rational.com/support/documentation/
48. Rumbaugh, J., Blaha, M., Premerlani, W.J.: Object Oriented Modeling and Design. Prentice-Hall, Englewood Cliffs (1991)
49. Song, I.Y., Evans, M., Park, E.K.: A comparative analisys of entity-relationship diagrams. J. Comput. Softw. Eng. **3**(4), 427–457 (1995)
50. Tardieu, H., Rochfeld, A., Coletti, R.: La Méthode MERISE. Tome 1: Principles et Outils. Les Editions d'Organisation, Paris (1983)
51. Teorey, T.J., Yang, D., Fry, J.P.: A logical design methodology for relational databases using the extended entity-relationship model. ACM Comput. Survey **18**(2), 197–222 (1986)
52. Teorey, T.J.: Database Modeling and Design: The Entity-Relationship Approach, 3rd edn. Morgan Kaufmann, San Mateo (1999)
53. Thalheim, B.: Achievements and problems of conceptual modelling, active conceptual modeling of learning. In: Lecture Notes in Computer Science, vol. 4512, pp. 72–96 (2008)
54. Thalheim, B.: Entity-Relationship Modeling: Foundations of Database Technology. Springer, Berlin (2000)
55. Türker, C., Gertz, M.: Semantic integrity support in SQL:1999 and commercial relational database management systems. Int. J. Very Large Database **10**(4), 241–269 (2001)
56. http://www.omg.org/technology/documents/formal/uml.htm

Author Biographies

Dolores Cuadra received her M.Sc. in Mathematics from Universidad Complutense of Madrid in 1995. In 1997, she joined the Advanced Databases Group, at the Computer Science Department of Carlos III University of Madrid, where she currently works as lecturer. In 2003, she obtained her Ph.D. degree in Computer Science from Carlos III University of Madrid. Her research interests include advanced database technologies, spatio-temporal databases and their applications to Situation management. She has been working in Computer Science Department at Purdue University of West Lafayette (Indiana) for nearly a year, where she has applied her research in Spatio-Temporal database.



Paloma Martínez received a degree in Computer Science and the Ph.D. degree in Computer Science from the Universidad Politécnica de Madrid (Spain) in 1992 and 1998, respectively. Since 1992, she has been with the Advanced Databases Group in the Computer Science Department, Universidad Carlos III de Madrid where she is currently teaching database design and management. Her research lines are human language technologies (multilingual information extraction and retrieval in several domains, question answering, name entity recognition and temporal information management) as well as web accessibility.



Elena Castro received her M.Sc. in Mathematics from the Complutense University of Madrid in 1995. Since 1998, she has been working in the Advanced Databases Group in the Computer Science Department at the Carlos III University of Madrid. In 2004, she obtained a Ph.D. in Information Science from the Carlos III University of Madrid. She currently works as associate professor and is currently teaching Relational and Object Oriented Databases. Her research interests include database conceptual and logical modeling, Advanced Database CASE environments, Information and Knowledge Engineering, Natural Language Processing and E-learning fields.

Harith Al-Jumaily received a Ph.D. degree in Informatics Engineering from the Carlos III University of Madrid in 2006. Since 1999, I have worked at the Advanced Databases Group in the Informatics Engineering Department at the Universidad Carlos III of Madrid. Currently I teach Data Structure and Algorithms, Files and Databases Structure, and Advanced Database Design. My research interests include Database Methodologies, Model-Driven Development, Natural Language Processing, CASE Tools and Active Databases.