

© 2009 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

# Educational Experiences Detecting, Using, and Representing Ternary Relationships in Database Design

Dolores Cuadra, Ana Iglesias Maqueda, Elena Castro, and Paloma Martínez Fernández

**Abstract**—Conceptual models are applied as the first step in software design methodologies for collecting the semantics involved in the universe of discourse. Nevertheless, the abstraction process creates some misunderstandings for novice designers, such as difficulties in modeling some constructs and in understanding the semantics that they represent. This paper presents a thorough study of errors detected among Database Design students in Computer Science Engineering when they apply the abstraction process to generate a conceptual schema using a specific model. Specifically, the paper focuses on errors made in the design of ternary relationships. Some heuristics are proposed in order to help novice designers avoid these common errors, and an experimental study is presented to compare the number of errors made by the students before and after applying these heuristics.

**Index Terms**—Abstraction activities, computer science education, data models, conceptual modeling.

## I. INTRODUCTION

CONCEPTUAL modeling is one of the most important phases in every software development process methodology. This phase must reflect the semantics of the problem domain, and the semantics should comply with two different requirements: they must be understood by the domain experts, and they must represent the complete semantics as far as possible, as it is necessary to keep them in a specific methodology from one phase to the next. Therefore, the conceptual model must be able to provide direct mapping without distortion between the perceived real world and its representation [1]. In general, conceptual models should be easy to learn and use, intuitive, independent of the implementation, and able to represent any domain [2]–[4]. However, some experiences [5]–[7] demonstrate that the use of conceptual models does not totally fulfill these requirements because the application of the abstraction process is not trivial. The abstraction process is defined as the activity of generalization that separates out what is important for a concept or an observable phenomenon

This work was supported in part by the Software Process Management platform project Modeling, Reuse and Measurement (TIN2004/07083), by the Spanish Ministry of Science and Innovation, and by the Universidad Carlos III de Madrid, which supports the APEINTA research and innovation project.

The authors are with the Computer Science Department of the Carlos III University of Madrid, 28911 Leganés, Madrid, Spain (e-mail: dcuadrapmf@inf.uc3m.es; aiglesiapmf@inf.uc3m.es; ecastropmf@inf.uc3m.es).

in order to retain only the information that is relevant for a particular purpose. The conceptual model is an important tool for computer science engineers to acquire because the abstraction process involved is crucial to solving problems. In the authors' experience, conceptual design falls into three activities: *detection*, *use*, and *representation*. These activities are performed when modeling the requirements specifications for the domain of interest. When reading the specifications, as a first step, the designer tries to map nouns, verbs, and adjectives with constructs of the model that s/he is using [8]. The choice of which construct to use is defined in the *detection activity*. The definition of the properties of the construct, and how they are applied according to the domain, is made in the *use activity*. The correct use of notations and syntactic rules is defined in the *representation activity*.

This article introduces several problems encountered by Database students analyzing a specific domain using a conceptual model. Specifically, the analysis was focused on modeling ternary relationships because these present several problems in their detecting, using, and representing [6]. The ternary relationship is frequently modeled as a combination of binary relationships, but this solution is not always applicable [3], [9]–[11]. This paper is focused on conceptual design, which is one of the most important disciplines in the training of new computer science engineers within the process of convergence with the European Space of Higher Education. The results could also be applied to the topics of database and software engineering.

The paper is structured as follows. Section II outlines related work reporting experiences in conceptual modeling. Section III describes the case study. Section IV describes the first phase of the study, giving an analysis of the errors made in the modeling of the ternary relationships and proposing some heuristics in order to help novice designers avoid some common errors in this process. Section V presents the second phase of the study, which examined whether the heuristics proposed to help students in the conceptual modeling were indeed of use. Finally, the conclusions of this paper are presented.

## II. RELATED WORK

Given that a conceptual schema is a communication tool among users and designers, simplicity and usability are necessary features. Graphic or visual languages for building schemes are used to facilitate discussion and validation with domain experts or users. Moreover, an experimental case [12] demonstrates how conceptual models should be able to improve the performance of users in their interaction with software artifacts.

Several experiments, guidelines, heuristics, and definitions of metrics have been proposed for the conceptual model. Certain coarse-grained rules to detect constructs have already been seen in the most frequently cited studies by Chen [13], [14]. For instance, in a textual requirements specification, an entity or class could be detected by a noun, an attribute by an adjective, and a relationship or association by a verb. These approaches are usually a good starting point for both practitioners and teachers because they show clear and simple guidelines for detecting the constructs. Other outstanding approaches are [5] and [6], which present some experiments focused on detecting the most difficult constructs for conceptual modeling [15] and list rules and heuristics to improve the design using this model. These approaches deal with two basic constructs: the entity<sup>1</sup> and the relationship.<sup>2</sup> Specifically, these papers show common errors when the degree and correspondence for binary and ternary relationships are determined. However, the detected errors only cover *detection* and *use* activities, and cardinality constraints have never been considered. The work presented in this article extends these heuristics and takes into account notation errors and cardinality constraints in ternary relationships.

The proliferation of conceptual models has motivated several experimental studies about which conceptual model is more appropriate, such as [16], where the entity/relationship (ER) model is compared with object-oriented and relational models. This approach is centered on five constructs (entity, descriptor or attribute, identifier, relationship, and generalization) and uses the correctness measure defined by [17] to analyze experiment results. The facets of a relationship were studied in particular, including reflexive, binary, and ternary relationships, because relationship is the most significant construct in this assessment. Moreover, an interesting study carried out by [18] shows a global comparison between three of the most extended conceptual models: ER, Unified Modeling Language (UML), and Object Role Model (ORM). The proposal analyzes the notation and specified constraints in ER and ORM models, and establishes a set of mappings that are later used for inclusion in the UML class diagrams. Two samples are studied in [19] (researchers and practitioners) to evaluate the quality of the conceptual modeling scripts from the user perspective, and the relationships found between researchers and practitioners. These approaches focus on the representation aspects.

On the other hand, some research has been published as to which model is easier to apply. In [20], the correlation between clarity and complexity are explained through different experiments. One of the main conclusions is that the use of complex constructs does not imply that the resulting diagram will be less clear. However, [21] studies the evolution of the ER model based on more than 100 contributions published in the top journals and demonstrates how the model has been extended to improve its expressiveness.

Some metrics have been defined to assess the quality of conceptual schemes. A more in-depth study is described in [22]. Twelve metrics are defined, but only six are taken into account

because of previous pilot studies that demonstrate their significance. The number of entities, relationships, and generalizations are the main indicators.

To summarize, the research on the conceptual model has centered on graphical notations [16], definition of metrics [22], [23], experiments on design errors [6], [17], [24], new definition of constructs [26], and redefinition of constructs to avoid the usual design errors [9], [10], [25]. Some of these experimental approaches were implemented in a CASE tool, as in [26]. All of them are focused on basic conceptual elements like entities, attributes, or binary relationships. However, some research papers [3], [25], [27] reveal that the ternary relationship construct is more usual than designers believe, but they avoid its use to simplify the conceptual scheme even though this can involve semantic loss.

The contribution of the present work is based on an in-depth study of conceptual design errors in the design of ternary relationships and is made according to three abstraction activities: detection, use, and representation. The detection of errors in each activity should lead to the reason for the error. The designer's lack of experience could be the reason for an error in the detection process. Semantic mistakes, or an unclear definition of a conceptual construct, can be the cause of an error when the use process has occurred. Lastly, a representation error appears to be due to the lack of standard representation—in other words, the proliferation of models—for a given conceptual construct. From an analysis of these errors, the authors propose some heuristics to improve the student's skills in modeling. These heuristics are included in an e-learning platform<sup>3</sup> [28] to give students guidance on dealing with ternary relationships.

### III. CASE STUDY DESCRIPTION

Computer Science students at Carlos III University of Madrid learn to model using several conceptual models (EER [15] or UML [29]). The work described in this paper was carried out with Database Design students and was divided into two main steps. First, the results of conceptual design exams from several semesters (from 2000 to 2005) were gathered in order to analyze the most common errors made by students. About 250 randomly selected exams were analyzed. All the exams give a description of the requirements to be modeled. The solution is a conceptual schema with five or six classes, six or seven associations (both binaries and ternaries), and at least one generalization. The second step of this work was to define a set of heuristics to be followed during the ternary relationships modeling process. These heuristics were incorporated into a teaching innovation project supported by Carlos III University within the European Space for Higher Education framework to test if they proved useful to students. Lastly, an experimental study is presented that compares the number of errors made by students both before and after applying the proposed heuristics.

The first analysis examines the difficulties students have when detecting, using, and representing generalizations and associations as the main constructs in conceptual models. This evaluation studies the number of students who failed for each

<sup>1</sup>“Entity” or “class” will be used interchangeably throughout this paper.

<sup>2</sup>“Association” or “relationship” will be used interchangeably throughout this paper.

<sup>3</sup><http://pid.basesdatos.uc3m.es/dadbd08/login.php>. Use “invitado” as user and “invi.tad0” as password for logging in.

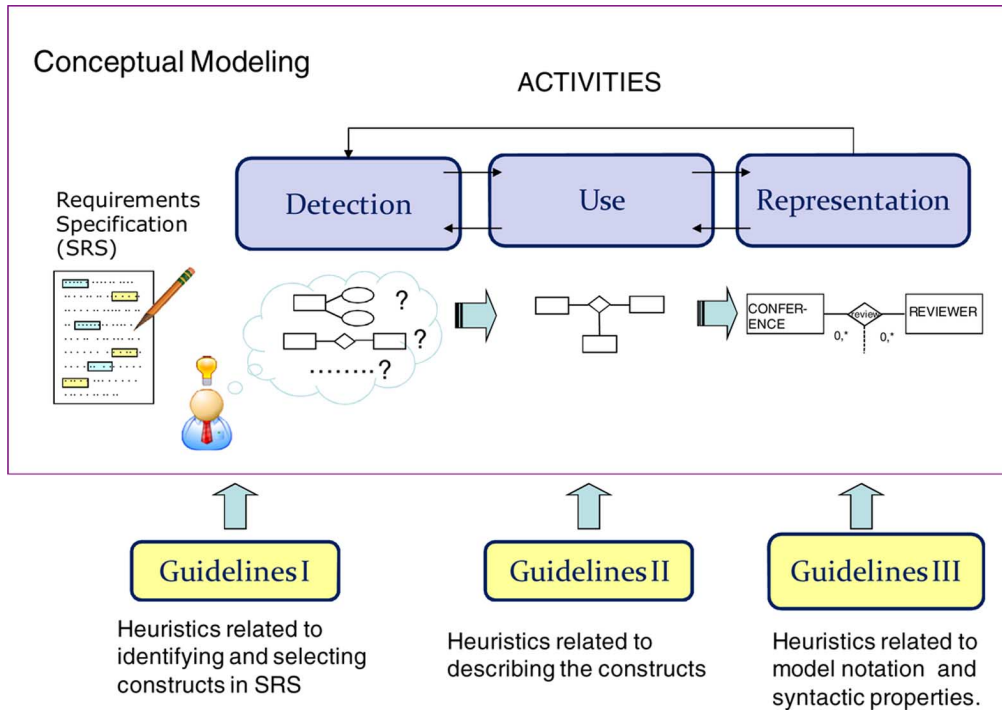


Fig. 1. Conceptual modeling activities.

abstraction mechanism (Fig. 1), in which activity they failed (detection, use, or representation), and the reason/s for each student's design errors. After this analysis, a set of expert recommendations (heuristics or guidelines) were defined in order to avoid these errors in the future. Subsequently, these recommendations were included in an e-learning platform. A set of exams of 80 students from 2006 and 2007, subsequent to the introduction of the e-learning platform, was selected in order to analyze its impact on the students' designs by evaluating whether these recommendations help the students avoid some common errors in their conceptual schemas.

#### IV. PHASE I: COMMON CONCEPTUAL ERRORS

As shown in Fig. 1, this section analyzes the systematic errors carried out in the abstraction process. The study consists of a comparison between the teachers' proposed solutions and the students' solutions. Conceptual design errors were categorized taking into account how many errors were in the representation, use, or detection process according to the experts' design criteria.

Ternary associations are one of the most difficult concepts to be detected and well represented in conceptual design. Examining the experimental exercises, four main common errors were detected for this construct.

**HAE<sub>1</sub> (High Association Error): Cardinality Mistakes:** This error is produced when the students need to represent the number of times an entity relates to other entities. This relationship property is known as the cardinality constraint. Frequently, novice designers are not able to specify cardinalities correctly. The problem has its origin in the way in which they ask for the participation of one entity in the context of the other two entities. These errors can be considered use or representation errors according to the way in which novice designers carry out the

scheme. Sometimes they mix the minimum or/and maximum cardinality constraint in one entity, causing a semantic loss, and sometimes they partially or totally forget the cardinality representation in one or more entities.

**HAE<sub>2</sub>: Representation With Binary Relationships:** The attempt to represent a ternary relationship by means of binary relationships is a common error in use and detection activities. The problem usually results in a lack of the necessary criteria to provide the semantics specified by the requirements.

**HAE<sub>3</sub>: Lack of One or More Attributes in the Relationship:** Another common error in detecting, using, and representing this construct is to leave out an attribute when it is specified in the requirements. The problem is not only one of omitting it, but also of designing this attribute as a property of one of the entities instead of in the relationship.

**HAE<sub>4</sub>: Existence of Redundancies:** Sometimes students forget one of the most important axioms in the definition of a conceptual schema: It must be nonredundant or have a controlled redundancy. Respecting this means that the semantics associated with any construct do not get duplicated. The non-controlled redundancy can occur due to the fact that students can model an alternative binary relationship to represent the semantics associated with two of the three entities involved in the ternary relationship.

Table I summarizes the percentage of errors for each former predefined error in the ternary relationship construct.

The most common errors in the detection, use, and representation of ternary relationships seem to be cardinality errors (87.4%). The second most common error (made by 60.9% of the students during the experiment) corresponds to the use of a binary relationship instead of a ternary one. The design of a schema with redundancy is the third usual error at 57.76%, and, finally, only 13.68% of the students did not understand when

TABLE I  
ERRORS IN THE TERNARY RELATIONSHIPS DESIGN

Relationships	Detection	Notation	Use	Total
HAE <sub>1</sub>		X	X	87.40%
HAE <sub>2</sub>	X		X	60.90%
HAE <sub>3</sub>	X	X	X	57.76%
HAE <sub>4</sub>	X		X	13.68%

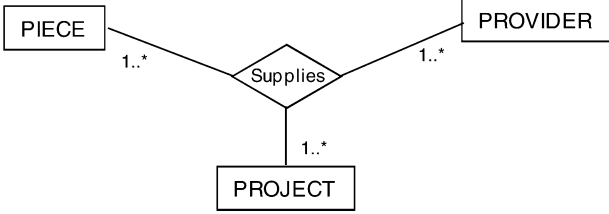


Fig. 2. Cardinality constraints for the example in H<sub>2</sub>-HAE<sub>1</sub>.

an attribute should belong to a relationship. Looking at the results, it seems to be clear that ternary relationships are difficult to model and understand. In fact, the use and detection of ternary relationships are the most difficult tasks in the abstraction process.

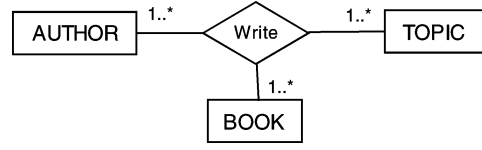
## V. HEURISTICS DEFINITION

After the experiment, some heuristics for each common error were formulated by experts in the field. In total, six heuristics were proposed.

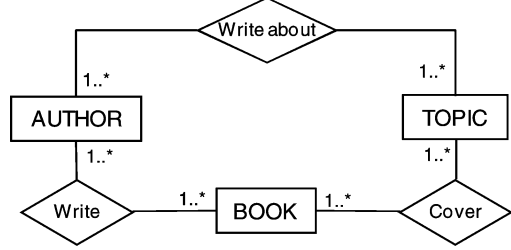
**H<sub>1</sub>-HAE<sub>1</sub>** (Heuristic 1 for Avoiding the HAE<sub>1</sub>): Be careful when reading any requirements specification. When auxiliary verbs such as “may” or “can” appear in the wording, it means a minimum cardinality of zero. Experimental results on detecting ternary relationships depending on familiar or unfamiliar domains and the requirements for textual specification are shown in [30].

**H<sub>2</sub>-HAE<sub>1</sub>**: The way to discover the cardinality of one entity with respect to the others is to ask for the minimum and maximum participation of one entity when the two other entities are fixed. For example, let there be three entities, *Provider*, *Piece*, and *Project*, that keep to the following restrictions: One piece for one project is supplied by at least one provider. A provider can supply a piece to several projects, and a provider can supply several pieces to one project. Fig. 2 displays the cardinalities of the ternary relationship. For instance, the minimum and maximum cardinality for *Provider* are 1 and n (many), respectively, because if it is asked how many providers can be involved in the *supplies* relationship, fixing one piece and one project, the answer is at least one (minimum cardinality 1), so it is possible that it can have many (maximum cardinality n). The other cardinalities are found by following the same reasoning.

**H<sub>3</sub>-HAE<sub>2</sub>**: When a design decision has been adopted, a careful study must be made in order to ensure that this decision takes account of all the semantics and that the ternary relationship cannot be represented by three binary relationships. For example, let there be three entities, *Author*, *Topic*, and *Book*, that participate in the ternary relationship *write*, storing



(a)



(b)

Fig. 3. (a) Ternary relationship for the example in H<sub>3</sub>-HAE<sub>2</sub>. (b) Incorrect representation with binary relationship.

AUTHOR #	BOOK#	TOPIC#
A1	B1	T1
A1	B2	T2
A2	B1	T1
A2	B1	T2
A3	B2	T3
A1	B2	T1

(a)

AUTHOR #	BOOK#
A1	B1
A1	B2
A2	B1
A3	B2

BOOK #	TOPIC#
B1	T1
B2	T2
B2	T3
B1	T2
B2	T1

AUTHOR #	TOPIC#
A1	T1
A1	T2
A2	T1
A2	T2
A3	T3

(b)

Fig. 4. (a) Instances for the ternary relationship. (b) Instances for binary relationships obtaining from projection over ternary relationship.

information about which topics each author wrote about in each book [Fig. 3(a)]. If a novice designer attempts to represent these requirements with binary relationships [Fig. 3(b)], the semantics represented could be different.

According to the data in the Fig. 4(a), Fig. 4(b) represents the binary projections over the relationship *write*. If the query is: “Topics covered by the AUTHOR#A1 in BOOK#B1,” then the colored instances are obtained after the joint operation with the binary relationships (AUTHOR#A1 writes about TOPIC#T1 and TOPIC#T2), while only the TOPIC#T1 in BOOK#B1 is true as is shown in Fig. 4(a). Other counterexamples such as “authors that have written about TOPIC#T2 in BOOK#B1” or “books written by AUTHOR#A1 about TOPIC#T2” are found when the query is formulated through binary relationships [Fig. 3(b)]. Then, the instances shown in the tables in Fig. 4(a) and (b) show that the semantics represented by the ternary relationship could be different to the semantics represented by the binary projections over the relationship *write*.

TABLE II  
SATISFACTION QUESTIONNAIRE

Answers should reflect your opinion from 1 to 5 for each question. 1. Totally disagree, 3. Indifferent, 5. Totally agree
1. Have you ever used the heuristics in the higher-order relationship guide? 1.1. Were they useful? 1.2. Did they clarify your concepts?
2. Would you like to use heuristics for other conceptual model constructs?

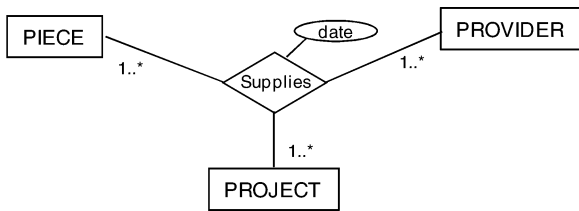


Fig. 5. Attribute in ternary relationships for the example in  $H_4$ -HAE<sub>3</sub>.

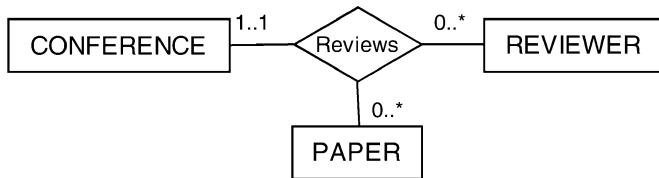


Fig. 6. Ternary relationship for the example in  $H_5$ -HAE<sub>4</sub>.

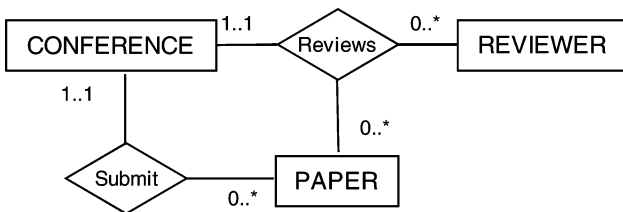


Fig. 7. Additional binary relationship for the example in  $H_5$ -HAE<sub>4</sub>.

$H_4$ -HAE<sub>3</sub>: A property is an attribute of a relationship when its values make sense only when the relationship or association exists. For example, the attribute *date* in Fig. 5 does not belong to the entities *Piece*, *Provider*, or *Project*. This property belongs to the relationship *supplies*. It only makes sense if the relationship exists.

$H_5$ -HAE<sub>4</sub>: When a ternary relationship occurs, the designer must check whether possible binary one-to-many or one-to-one relationships between the entities allow the ternary relationship to be refined. The example (Fig. 6) shows the *reviews* relationship that represents an initial design for the assignment of papers to reviewers in conferences.

In a second design iteration, the *submit* relationship appears, representing that a *paper* is only submitted to one *conference* (Fig. 7). This new constraint then allows the *review* relationship to be decomposed as is depicted in Fig. 8.

Additionally, there are three relationships forming a cycle, and a careful study must be performed of how to acquire the information required by the semantic assumptions. If this can be achieved in more than one way, the relationship that motivates

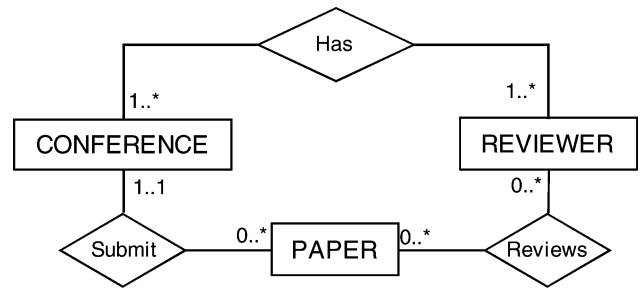


Fig. 8. Decompositions in Binary relationship.

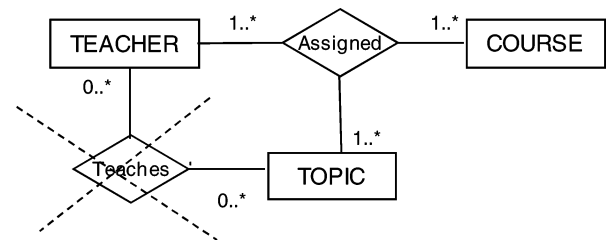


Fig. 9. Redundant Binary relationship for the example in  $H_6$ -HAE<sub>4</sub>.

the redundancy must be eliminated. In the cycle shown in Fig. 8, there is no redundancy.

$H_6$ -HAE<sub>4</sub>: when a conceptual scheme contains both a ternary relationship and a binary relationship between two of the three entities included in the ternary relationship, it must be determined whether the information that can be collected in the binary relationship is also stored in the ternary relationship. If this is the case, the binary relationship must be deleted. Fig. 9 shows a ternary relationship, *assigned*, among “teacher,” “course,” and “topic,” showing how a topic in a course is taught by several teachers, a teacher teaches a topic in different courses, and, finally, a teacher in a course teaches several topics. If a novice designer includes the *teaches* binary relationship between “teacher” and “topic” (a teacher teaches several topics, and a topic is given by several teachers), this new relationship represents redundant data and has to be dropped from the scheme.

## VI. PHASE II: USING HEURISTICS

These heuristics were provided to the Database Design students in 2006 and 2007, both via the e-learning platform and in the classroom by the teacher. The students were then asked to complete an exam including one exercise on conceptual modeling and were asked to fill out a satisfaction questionnaire about the heuristics (Table II).

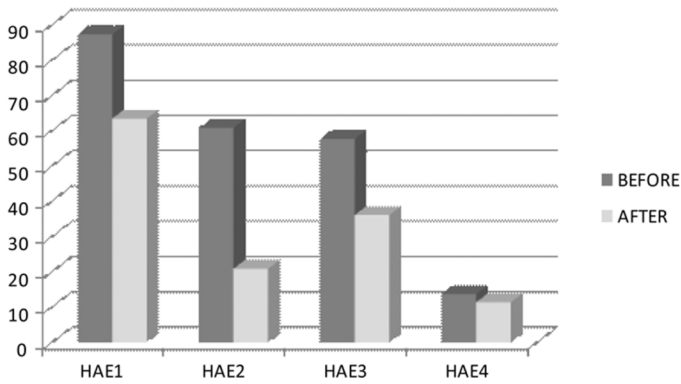


Fig. 10. Results of ternary associations.

Ninety percent of the students answered affirmatively regarding the use of ternary relationships heuristics and indicated that they would like to use heuristics for other conceptual constructs. The experimental results are shown in Fig. 10. The bar charts show the percentage of errors before and after applying the proposed heuristics; as can be seen, the number of errors made by students was reduced by the use of the heuristics.

The most significant improvements were in the error HAE<sub>2</sub> (representation with binary relationships), at 39.95%; for cardinality errors (HAE<sub>1</sub>) and the detection of attributes (HAE<sub>3</sub>) in the relationship, 23.88% and 21.57% of the answers were correct, respectively, whereas the detection of redundancies error only decreased by 2.26%. However, this is a significant result given the difficulties of the redundancy analysis. To summarize, this experimental study observed that the proposed heuristics mainly helped the student in two activities in the abstraction process: the detection and the use of ternary relationships.

## VII. CONCLUSION

The work presented here helps novice designers in conceptual modeling to avoid common errors in their schemas by applying new methods in Database Design, a topic within the framework of the European Space for Higher Education. The work studies in-depth ternary relationships because, although this construct is found in the real world, designers avoid their use because they are difficult to detect, are complex both in use and representation, do not appear in some CASE Tools, and their implementation is usually handled by binary relationships.

The case study presented in this article is divided into two phases. The first analyzes the systematic errors (independent of the domain) in 250 exams, classified according to the abstraction process, and proposes heuristics for improving the conceptual model. The second proposes a set of heuristics and guidelines to be considered during the modeling process. These heuristics were incorporated into a Database Design e-learning platform. The exams analysis carried out in the first step and new exams taken by students using the proposed heuristics permitted the errors made by the novice designers before and after these recommendations to be compared. In addition, student questionnaires allowed the heuristics to be fine-tuned. The proposed heuristics reduced the percentage of errors in detection and use, the two most frequent forms of errors.

## REFERENCES

- [1] C. Parent, S. Spaccapietra, and E. Zimányi, "Spatio-temporal conceptual models: Data structures, space and time," in *Proc. 7th ACM Int. Symp. Adv. Geographic Inf. Syst.*, Kansas City, MO, 1999, pp. 26–33.
- [2] J. R. Abrial, "Data semantics," in *Proc. IFIP Working Conf. Data Base Manage.*, Amsterdam, The Netherlands, 1974, pp. 1–59.
- [3] A. Badia, "Entity-relationship modeling revisited," *SIGMOD Record*, vol. 33, no. 1, pp. 77–82, Mar. 2004.
- [4] D. Harel and B. Rumpe, "Meaningful modeling: What's the semantics of semantics?," *IEEE Comp. Soc.*, vol. 31, no. 10, pp. 64–72, Oct. 2004.
- [5] D. Batra and J. Davis, "Conceptual data modelling in database design: Similarities and differences between expert and novice designers," *Int. J. Man-Machine Studies*, vol. 37, pp. 83–101, 1992.
- [6] D. Batra and S. R. Antony, "Novice errors in conceptual database design," *Eur. J. Inf. Syst.*, vol. 3, no. 1, pp. 57–69, 1994.
- [7] P. Martínez, A. De Miguel, D. Cuadra, and E. Castro, "Data conceptual modeling through natural language: Identification and validation of relationship cardinalities," in *Proc. Inf. Resources Manage. Assoc.*, Anchorage, AK, May 2000, pp. 500–504.
- [8] P. Martínez and A. García-Serrano, "The role of knowledge-based technology in language applications development," *Expert Syst. Appl.*, vol. 19, no. 1, pp. 31–4, 2000.
- [9] D. Cuadra, P. Martínez, and E. Castro, "Dealing with relationship cardinality constraints in relational database design," in *Effective Databases for Text & Document Management*. Hershey, PA: IRM, 2003, pp. 288–317.
- [10] G. Génova, J. Llorens, and P. Martínez, "Semantics of the minimum multiplicity in ternary associations in UML," in *The UML. Modeling Languages, Concepts, and Tools*, M. Gogolla and C. Kobryn, Eds. New York: Springer-Verlag, 2001, vol. 2185, pp. 329–342.
- [11] L. G. Jiménez, "REERM: Reenhancing the entity-relationship model," *Data Knowl. Eng.*, vol. 58, pp. 410–435, 2006.
- [12] M. Ben-Ari and T. Yeshno, "Conceptual models of software artifacts," *Interaction on Comput.*, vol. 18, no. 6, pp. 1336–1350, 2006.
- [13] P. Chen, "English sentence structure and entity-relationship diagrams," *Inf. Sci.*, vol. 29, no. 2–3, pp. 127–149.
- [14] P. Chen, "The entity-relationship model—Towards a unified view of data," *Trans. Database Syst.*, vol. 1, no. 1, pp. 9–36, 1976.
- [15] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 5th ed. Reading, MA: Pearson/Addison-Wesley, 2007.
- [16] C. Liao and P. C. Palvia, "The impact of data models and task complexity on end user performance: An experimental investigation," *Int. J. Human-Comput. Stud.*, vol. 52, pp. 831–845, 2000.
- [17] D. Batra, J. A. Hoffer, and R. P. Bostrom, "Comparing representations with relational and EER models," *Commun. ACM*, vol. 33, pp. 126–139, 1990.
- [18] T. A. Halpin, "Metaschemas for ER, ORM and UML data models: A comparison," *J. Database Manag.*, vol. 13, no. 2, pp. 20–30, 2002.
- [19] A. Maes and G. Poels, "Evaluating quality of conceptual modelling scripts based on user perceptions," *Data Knowl. Eng.*, vol. 63, pp. 701–724, 2007.
- [20] A. Gemino and Y. Wand, "Complexity and clarity in conceptual modeling: Comparison of mandatory and optional properties," *Data Knowl. Eng.*, vol. 55, pp. 301–326, 2005.
- [21] S. Patig, "Evolution of entity-relationship modeling," *Data Knowl. Eng.*, vol. 56, no. 2, pp. 122–138, 2006.
- [22] M. Piattini, "Defining and validating metrics for assessing the understandability of entity-relationship diagrams," *Data Knowl. Eng.*, vol. 4, no. 3, pp. 534–557, 2008.
- [23] D. Moody, "Theoretical and practical issues in evaluating the quality of conceptual models: Current state and future directions," *Data Knowl. Eng.*, vol. 55, no. 3, pp. 243–276, 2005.
- [24] D. Batra and H. Zanakis, "A conceptual database design approach based on rules and heuristics," *Eur. J. Inf. Syst.*, vol. 3, no. 3, pp. 228–239, 1994.
- [25] M. Balaban and P. Shoval, "MEER—An EER model enhanced with structure methods," *Inf. Syst.*, vol. 27, pp. 245–275, 2002.
- [26] S. Hartmann and S. Link, "Collection type constructors in entity-relationship modeling," *Conceptual Model.—ER*, vol. 4801, pp. 307–322, 2008.
- [27] B. Thalheim, *Entity-Relationship Modeling: Foundations of Database Technology*. Secaucus, NJ: Springer-Verlag, 2000, pp. 55–93.
- [28] L. Moreno, A. Iglesias, E. Castro, and P. Martínez, "Using accessible digital resources for teaching database design: Towards an inclusive distance learning proposal," in *Proc. 13th SIGCSE Conf. ITCSE*, Madrid, Spain, 2008, pp. 32–36.

- [29] *OMG: UML 2.0 Superstructure Specification*, OMG Adopted Specification (ptc/03-08-02), 2003.
- [30] E. Castro, D. Cuadra, and P. Martínez, “An empirical perspective of using ternary relationships in database conceptual modelling,” *Model. Inf. Educ.*, vol. 2, pp. 191–199, 2003.

**Dolores Cuadra** received the M.Sc. degree in mathematics from the Complutense University of Madrid, Madrid, Spain in 1995, and the Ph.D. degree in computer science from the Carlos III University of Madrid, Madrid, Spain, in 2003.

Since 1997, she has been an Assistant Lecturer in the Advanced Database Group in the Computer Science Department at the Carlos III University of Madrid. She is currently teaching file organization, database design, and data modeling. She previously worked in the Computer Science Department at Purdue University, West Lafayette, IN, for nearly a year, where she applied her research in spatio-temporal databases. Her research interests include data models, conceptual and logical modeling, and advanced database CASE environments.

**Ana Iglesias Maqueda** received the degree computer science and the Ph.D. degree in computer science from the Carlos III University of Madrid, Madrid, Spain, in 1999 and 2004, respectively.

She has been a Member of the faculty of the Computer Science (CS) Department at the Carlos III University of Madrid since February 2006. Since 2000, she has been with the Advanced Databases Group in the CS Department at the university. She is currently teaching database design and advanced databases.

From January until August 2005, she carried out a post-doctoral stay at the HCI Institute of Carnegie Mellon University, Pittsburgh, PA. She has been working on several national research projects on advanced database technologies, CASE environments, natural language processing, and information retrieval.

**Elena Castro** received the M.Sc. degree in mathematics from the Complutense University of Madrid, Madrid, Spain, in 1995, and the Ph.D. degree in information science from the Carlos III University of Madrid, Madrid, Spain, in 2004.

Since 1998, she has been with the Advanced Databases Group in the Computer Science Department at the Carlos III University of Madrid. She 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.

**Paloma Martínez Fernández** received the degree in computer science and the Ph.D. degree in computer science from the Politécnica University of Madrid, Madrid, Spain, in 1992 and 1998, respectively.

Since 1992, she has been with the Advanced Databases Group in the Computer Science Department, Carlos III University of Madrid, Madrid, Spain, where she is currently teaching database design and advanced databases. She is working on several European and national research projects on natural language processing, information retrieval, advanced database technologies, and software engineering.