Bridging the gap between natural and information modeling languages: an informal approach to information modeling learning

Vinícius Medina Kern and André Luiz Moraes Ramos

Abstract **— Information modeling is an early stage of database design. It deals with the structure of information in a certain business or domain. Information modeling is studied in database courses using the Entity-Relationship approach, or one of its dialects. This article presents an informal, pragmatic approach to the learning of information modeling, in which students answer true or false to assertions about given models. It is based on similarities between natural and information modeling languages. This approach has demonstrated to be an effective, rapid way to sharpen modeling skills, as our initial statistical results show.**

Index Terms **— Information modeling, database design, Entity-Relationship modeling, IDEF1X, Engineering education.**

I. Introduction

Information modeling deals with information structure and constraints in a certain domain. This early stage of database design is abundant with complexity and conflicting interests. Conceptions at this stage determine critical business rules.

Similarities between natural and information modeling languages' syntax have been discussed in the literature [1]-[4]. Nevertheless, information modeling languages resemble first order logic. The gap between this and the natural language used by experts in a domain area to describe how their business works is a source of misconceptions for the inexperienced modeler (and even to expert ones, at times).

The building blocks of information modeling are simple to understand, but developing expertise is hard. Most textbooks simply present a modeling language's basic features and a few toy examples that not necessarily lead the apprentices to develop the understanding they need.

This paper presents a novel approach to information modeling learning in which the student, after a brief introduction to modeling language syntax and semantics, is confronted with hard questions about given models.

Contrasting with natural languages, information modeling languages state only non-ambiguous sentences (if the model is free of technical errors). Therefore, an assertion about a model can always be answered (if applicable) as true or false.

The answer can be compared with the correspondent business rule given by a business expert, in natural language. If they match, the model complies with the business rule. If they don't, the model has to be redesigned.

The motivation for this study came from a classroom experience. The professor noted that, in the first tries in these true-or-false exercises, some students tended to get about half of their answers right. This is the expected success rate for someone who never heard of information modeling.

One specific student had a very poor grade in a first test. After having questioned the professor about the reasons for all of his rights and wrongs, he was able to choose the correct answer for all questions in a second test. This suggested that the approach's effectiveness to develop sharp modeling skills was worth researching.

The next section introduces the basic features of the IDEF1X information modeling language. Then, a brief comparison of similarities of natural and information modeling languages is presented. Next, the approach to improve information modeling learning is detailed. An account of results is presented. The Conclusion summarizes the article and gives recommendations for extension and improvement of this brief experiment.

Manuscript received on December 15, 2001. (Deadline date).

V. M. Kern, UNIVALI at São José, Brazil, Computer Engineering and Computer Science, kern@eps.ufsc.br; A. L. M. Ramos, UNIVALI at Biguaçu, Brazil, Psychology, andrelmr@big.univali.br.

2. Information modeling and IDEF1X

The modeling language used in this study is IDEF1X [5], a dialect of the Entity-Relationship model introduced by [6]. IDEF1X is an American standard that includes a language formalization and a method for relational database design.

The task of information modeling is to delineate the nature of information [7]. The modeler identifies **what** to store, it is not important at this point **how** to process stored information.

However, to model the nature (structure and constraints) of information is to create rules that restrict the way in which information can be processed. Neither database application programs, nor the database administrator, using an interactive interface, can break these high-level rules. That's why errors in information modeling are so critical.

IDEF1X has a simple syntax. **Entities** represent classes of things with the same **attributes**, or characteristics of these things. Entities can have associations of structural nature called **relationships**.

Fig. 1 illustrates the graphical representation of entities, relationships, and attributes. In this university model, the entity Department has the attributes idDept (an identifying code) and nameDept (name). The entity Course has the code from the Department it belongs to (idDept), a course number

(noCourse), course name (nameCourse), and number of credits (credits).

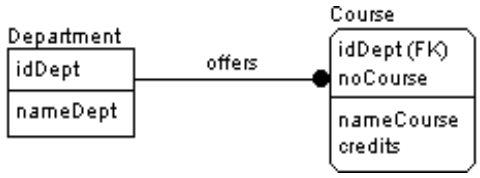

Fig. 1. A simple information model

Key attributes are those whose values identify an entity instance. They are represented in the upper part of the entity box. For instance, each Course in fig. 1 is identified by the code of the Department that offers the Course and the number of the Course.

There is a structural association between departments and courses according to fig. 1: a department may offer courses, and each course is offered by a department. Every relationship corresponds to a **key migration** annotated by FK (foreign key), like in idDept (FK).

IDEF1X relationships can be any one of the tree leaves of fig. 2. The relationship in fig. 1, for example, is a specific connection **identifying** relationship, typical of whole-part associations. The identity of Department becomes part of the identity of Course – Department's key becomes part of Course's key.

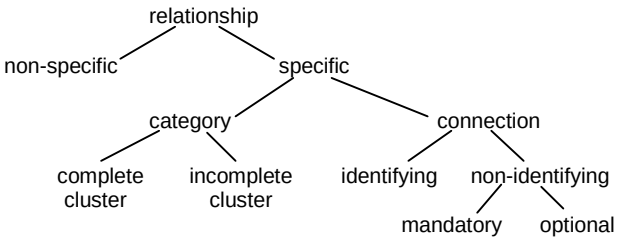

Fig. 2. Relationship types in IDEF1X

The broader classification of IDEF1X relationships regards specific and non-specific relationships. Specific relationships are one-to-one or one-to-various relationships. For instance, Department-Course is one-to-various. Choosing a specific Course leads to one (specific) Department that offers it.

Non-specific relationships are various-to-various. They cannot be implemented in relational databases and have to be translated into two or more specific relationships. The discussion about why this is so and how to solve the problem are out of the scope of this article.

Specific relationships can be of two types: connection or category. Category relationships represent type-subtype or "is a" abstractions. Fig. 3 illustrates category relationships.

A category cluster is **complete** if each instance of the generic (supertype) entity is associated with (exclusively and necessarily) one instance of one of the category (subtype) entities. For example, each Business party in fig. 3 is either a Department or an Employee.

A category cluster is **incomplete** if a specific instance of the generic entity can be associated with (exclusively) one, or none of the categories. Fig. 3 shows an incomplete cluster where Customer is the only category.

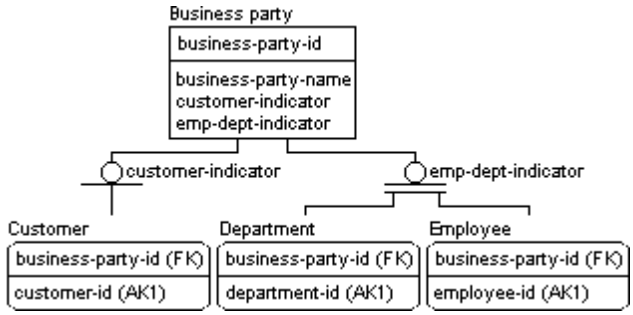

Fig. 3. Incomplete and complete category clusters [8]

Fig. 3 also exhibits the syntax used to indicate that an entity has more than one key. Additional keys are annotated as **alternate keys**, in the form (AKn). Customer, Department and Employee have one alternate key each.

Specific connection relationships are also known as **parent-child** relationships. They can be **identifying**, as illustrated in fig. 1, or non-identifying. The key, or at least part of the key migrated through non-identifying relationships does not take part in the child entity's key.

Fig. 4 shows the representation for **mandatory** non-identifying relationships, e. g. Employee manages Project, and **optional** non-identifying relationships, e. g. Employee is chief supervisor of Project. This means that every Project must have a manager, and may have a chief supervisor. Role names (manager-id and chief-supervisor-id) were assigned to the migrated attributes empl-id to avoid ambiguity in Project.

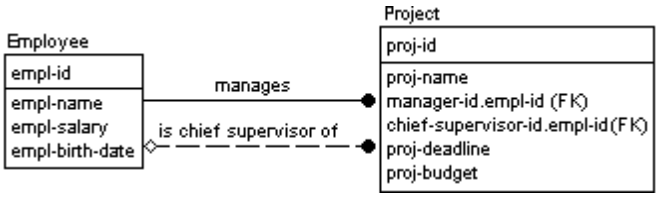

Fig. 4. Non-identifying relationships.

Cardinality is another important feature of relationships. It is related to the number of instances of one entity that can be associated with a specific instance of the other entity. From the perspective of a child or category entity, the cardinality is one, with the exception of the optional non-identifying relationship. In this case, the cardinality is zero-or-one. In fig. 4, for instance, a Project has, as chief supervisor, zero or one Employee.

From the perspective of a generic entity in a category cluster, the cardinality is always zero-or-one, with the additional constraint of category exclusiveness. In fig. 3, for instance, a Business party is, exclusively and imperatively, zero or one Employee, and zero or one Department.

From the perspective of a parent entity in a connection relationship, there are several possibilities. The default cardinality is zero-or-one-or-various, represented graphically

by the black dots in figs. 1 and 4. However, it is possible to assign other parent-to-child cardinalities, as shown in fig. 5.

Fig. 5 also illustrates the use of IDEF1X **notes**. Notes are written in natural language when there is no way to represent an information constraint as entity, relationship, or attribute.

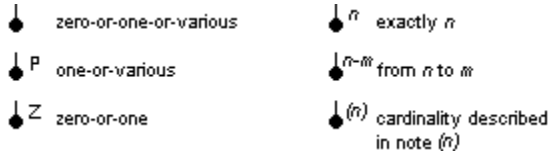

Fig. 5. Parent-to-child cardinalities in IDEF1X

The graphic language syntax just described has similarities with first order logic, with semantic and pragmatic implications that students sometimes don't realize. The next section tries to bridge the gap between the logic-based IDEF1X language and natural language.

3. Natural and Information Modeling Languages

A formal approach to the definition of semantic constraints in databases is available [9]-[11]. The most commonly used semantic constraints in information modeling are [3]: functional dependencies, keys, inclusion dependencies, exclusion dependencies, and cardinality constraints.

Newcomers to information modeling may find it difficult to follow the sound mathematical basis behind these constructions. Moreover, the number of semantic constraints that have to be checked grows exponentially with the number of attributes [3].

Similarities between natural and information modeling languages' syntax have been discussed in the literature. The correspondence between Entity-Relationship diagrams and English sentence structure was studied [1]. The relationship between natural languages and information modeling was announced as an important research area of information modeling [12].

When it comes to learning information modeling techniques, informal approaches can be much more intuitive and easy. Reference [2] observes that natural language sentences express semantic constraints intuitively.

In IDEF1X, any relationship can be read as two sentences, using the formula:

$$<SIA> <NS> <VP> <Q> <NP>. \quad (1)$$

where *SIA* stands for singular indefinite article, *NS* stands for noun in singular form, *VP* stands for verb phrase, *Q* stands for a quantifier, and *NP* stands for noun in singular or plural form.

The *SIA* turns the subject of the sentence into a specific instance. *NS* completes the subject – it is the name of an entity. *VP* is the transitive verb or verb phrase that serves as relationship name when the reading is made from parent to child entity (the inverse verb phrase must be taken if the reading goes in the opposite direction). The quantifier (*Q*) is the relationship's cardinality in the appropriate direction. Finally, *NP* is the object – it is the name of the other entity (with plural reading when the sentence is formed from parent to child).

In summary, (1) can be unfolded in two sentences, one for each direction of the relationship, where parent and child can be changed by generic and category, if appropriate:

A(n) <*parent entity name*> <*relationship name*> <*parent-to-child cardinality*> <*plural of child entity name*>. (2)

A(n) <*child entity name*> <*inverted relationship name*> <*child-to-parent cardinality*> <*parent entity name*>. (3)

The relationships in fig. 4, for instance, can be read using (2) and (3) as:

An Employee manages zero, one, or various Projects.
A Project is managed by exactly one Employee.

An Employee is chief supervisor of zero, one, or various Projects.
A Project has as chief supervisor zero or one Employee.

This syntactic approach to the reading of relationships makes it easy to understand the model's meaning (semantics). However, ref. [12] points out the existence of the pragmatic aspect – the part about which computer scientists are least concerned when dealing with detailed design decisions.

Pragmatics deals with practical aspects of sign usage. There are practical questions that can be asked about the model in fig. 4, for instance: "Can the manager and the supervisor (if existent) of a project be the same employee? Should they be? Shouldn't they?"

This specific problem is known in information modeling as the **dual path** problem, since there are more than one path or series of relationships to associate a Project to an Employee. Incidentally, the answer to the question above is: the model in fig. 4 says nothing about manager and supervisor being the same employee. Therefore, an instance of Project may have the same employee as manager and supervisor, or not.

The problem is: what if the manager and the supervisor of a project shouldn't be the same employee? The model should be changed, i. e., the syntax of the model sentences should change. The approach to improve information modeling learning, presented next, uses true-or-false assertions about a model in order to challenge the students' ability to deal with pragmatic aspects of model interpretation, using the skill they developed to work with IDEF1X's syntax.

4. An Approach to Improve Information Modeling Learning

This section presents an original, informal approach to information modeling learning in which the student, after a brief introduction to the syntax of the IDEF1X modeling language, is confronted with hard questions about given models. The students are then submitted to an examination. After a detailed discussion about the exact reason why each assertion is true or false, they try again and the results of both examinations are compared.

The introduction to IDEF1X prior to the two-round examination includes an account of its similarities with natural language. Equations (2) and (3) in the previous

sections, for example, are suggested for intensive use while examining relationships. The professor stresses the fact that entities are substantives (serving as subject and object of relationship phrases), relationships are transitive verbs or verb phrases, and attributes play the role of adjectives, or characteristics of the entities.

Students are introduced to models such as the one in fig. 6, which shows a university model with departments, undergraduate courses, and their *curricula* (list and sequence of courses). This introduction includes the definitions of entities and attributes. For instance, Department is "an administrative unit of the university"; Precedence constraint is "an association between two courses in the curriculum of a course; one of them comes before the other in the curriculum sequence".

Attributes must also be defined if the model is to be understood. The attribute noCourse, for instance, means "a course discriminating number; it identifies a course if we consider a specific department, but it is not an identifier *per se*." All "id" attributes in fig. 6 are identifying codes.

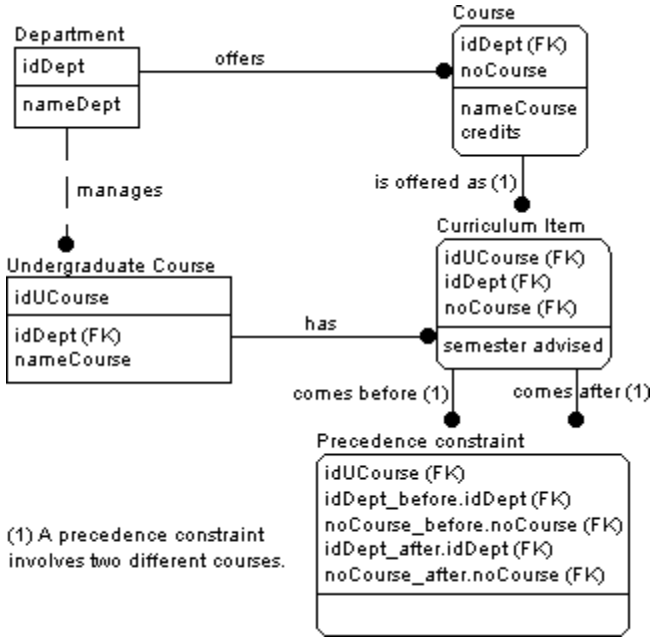

Fig. 6. A university information model

Next, the students are asked to answer T (true) or F (false) to assertions such as:

- Considering a curriculum item, the department that offers the respective course and the department that manages the undergraduate course to which the curriculum item belongs to do not need to be the same.
- An undergraduate course can have various *curricula* (list and sequence of courses) at a time.
- A precedence constraint cannot involve two courses of the same (attribute value of) semester advised.

Informally, all three assertions are expected to hold with a university environment. However, a careful examination of the model reveals that only the first assertion is true. The second assertion is false because, if two *curricula* were possible, there would be a manner to distinguish them (nevertheless, it is possible to distinguish two courses based on their identifiers – idCourse). The third assertion is false because the model says that a Precedence constraint is an association between two Curriculum items. There are constraints related to Undergraduate course and to Course, but no constraints on the two values of semester advised.

Considering that all three assertions above are expected to be true in a real university, it is possible that students judge the assertions biased by this previous knowledge. In order to eliminate bias, another kind of model is also proposed for interpretation.

Assertions similar to those about the university model can be made about abstract models, i. e., models of meaningless businesses, yet using meaningful designs (since they are written in IDEF1X, which has a well-defined syntax).

It can be demonstrated that the assertion "any specific instance of entity D is associated with one or two instances of entity A", in fig. 7, is true. It is also true that "instances of A and B cannot exist independently in the database; if there is no A in the database, then there is no B, and vice versa".

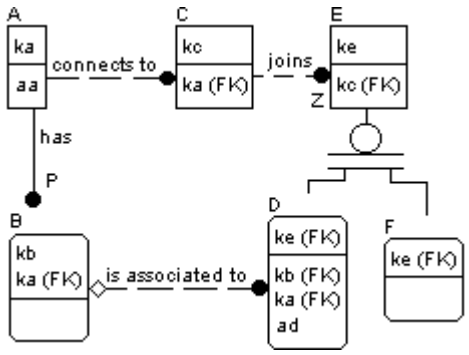

Fig. 7. An abstract information model (meaningless business, but meaningful design)

After a first series of 10 to 15 questions about both models and a thorough discussion about the answers, the students are invited to try again, with new questions. The results of both exercises are then compared.

5. Results and Discussion

Surprisingly for the professor, several students guessed right about 50% of the questions in the first tries. This is the arithmetic mean for a true-or-false test. The professor knows, however, that they are not giving random answers.

It has to be taken into account the fact that the questions are challenging. In such an experiment, in which the students are asked to decode pragmatic but difficult business rules, it is likely or admissible that beginners lack the precision that could be expected from experts. It is noticeable, however, that even experimented computer professionals hardly ever have all questions right.

A two-round paired-data test was prepared in order to measure whether the students improve or not their performance after a syntax-based discussion about the reasons for every True of False answer. The hypotheses are:

H0: Students' performance is not altered after the discussion.
H1: Students' performance improve after the discussion.

Eleven students took the test for both the abstract and the university (concrete) model. The results for the abstract model (fig. 7) are shown in table I. The students reached a very similar success rate (around 70%) before and after the discussion. A T-test measure of -0,15 was calculated, therefore endorsing hypothesis H0. The results are independent of the discussion (considering a significance probability of 5%).

TABLE I
PERFORMANCE MEASURE FOR THE ABSTRACT MODEL

Round	1 (before discussion)	2 (after discussion)
Percent success μ and σ	71.97 and 17.14	70.92 and 16.61
Test t	t = - 0.15	
Degrees of freedom	10	
Significance	t(α=5%) = 1.81 > - 0,15	

Student's performance regarding the questions about the university model (the real-life, concrete model in fig. 6) is reported in table II. The average success rate improved from about 50% to nearly 70%. The T-test measure of 3,80 allows for the conclusion that there is a significant (considering a significance probability of 5%, and even for a significance probability of 1%) improvement in students' performance. Therefore hypothesis H0 should be rejected. Hypothesis H1 is valid – students performance improve after the discussion.

TABLE II
PERFORMANCE MEASURE FOR THE REAL MODEL

Round	1 (before discussion)	2 (after discussion)
Percent success μ and σ	52,02 and 11,54	71,30 and 13,62
Test t	t = 3.80	
Degrees of freedom	10	
Significance	t(α=5%) = 1.81 < t(α=1%) = 2.76 < 3,80	

6. CONCLUSION

This article presented a brief introduction to the IDEF1X information modeling language, together with its resemblance to natural language. An informal approach to information modeling learning designed to challenge students' modeling abilities and to accelerate learning was outlined. Initial results of its application were measured.

The idea for this study came from the professor's informal perception that the approach is effective to make the students sharpen their modeling skills. In one specific case, the performance jumped from less than the arithmetic mean to 100% of correct answers. A performance measure taken with 11 subjects revealed that the impact of the approach is still very significant, with an increase in the average percentage of right answers from roughly 50% to 70%.

Comparing the results for the abstract and the concrete model, it is clear that the students did not improve significantly their abilities in logic after the discussion about their right and wrong answers. However, given the significant improvement regarding the concrete model, it is possible to observe that they used the opportunity to sharpen their understanding of the syntax.

The final success rate for the concrete model is notably close to the success rate for the abstract model (about 70%). It is likely that the second try was taken with a greater focus on the language syntax and less tendency to answer according to ideas about how the business (the university) **should** work.

Although the results suggest that the approach is, indeed, valid to improve students' success rate in the interpretation of a concrete information model, the conclusion is limited by the fact that the sample is small, the two models in the first and second round were the same, and the choice of questions for the first and second rounds might carry different difficulty levels.

The research should be extended to include a broader sample, with varied models and varied questions, in varied order. The inclusion of modeling (i. e., conception) questions, in addition to the model interpretation questions, will allow for the correlation between modeling and model reading skills. It is promising, too, an investigation of students' performance on the pragmatic interpretation of specific types of semantic constraints, as categorized by [3]. This could help to understand what are the constraints in which this pragmatic approach works best, and what are the constraints that maybe deserve another learning approach.

REFERENCES

[1] P. P. Chen, "English sentence structure and Entity-Relationship diagram", *Information Science* 29 (2), Elsevier, pp. 127-149, May 1983.

[2] E. Buchholz and A. Düsterhöft, "Using Natural Language for Database Design", In: *Proceedings Deutsche Jahrestagung für KI 1994* - Workshop "Reasoning about Structured Objects: Knowledge Representation meets Databases", Saarbrücken, 1994.

[3] M. Albrecht, E. Buchholz, A. Düsterhöft, and B. Thalheim, "An Informal and Efficient Approach for Obtaining Semantic Constraints using Sample Data and Natural Language Processing", In: *Proceedings of the International Workshop Semantics in Databases*, Prague, 1995.

[4] B. Thalheim, E. Buchholz, H. Cyriaks, A. Düsterhöft, H. Mehlan, "Designing EER-Skeleton Schemes based on Natural Language", In: *Proc. OO-ERA95/Australia*, Poster Session, Dec. 1995.

[5] NIST (National Institute of Standards and Technology), *Federal Information Processing Standards Publication 184: Integration Definition for Information Modeling (IDEF1X)*, Gaithersburg, MD, December 1993.

[6] P.P. Chen, "The Entity-Relationship model - toward a unified view of data", *ACM Transactions on Database Systems* 1 (1), pp. 9-36, 1976.

[7] J. P. Ashenfelter, "Database Design for Education and Academe", *WebNet Journal* 1 (3), 1999. [Available at http://www.webnetjrl.com/]

[8] T. A. Bruce, *Designing quality databases with IDEF1X information models*, Dorset House Publishing, 547 p., 1992.

[9] B. Thalheim, "Foundations of Entity-Relationship Modeling", *Annals of Mathematics and Artificial Intelligence*, 7, pp. 197-256, 1993.

[10] B. Thalheim, "An overview on semantical constraints for database models", *6th International Conference on Intellectual Systems and Computer Science*, Moscow, Russia, December 1-10, 1996.

[11] B. Thalheim, *Entity-Relationship modeling: foundations of database technology*, Springer-Verlag, 627 p., 2000.

[12] P. P. Chen, B. Thalheim, and L. Y. Wong, "Future Directions of Conceptual Modeling", In: *Conceptual modeling: current issues and future directions*, P.P. Chen et al. (eds.), Berlin: Springer-Verlag, Lecturing Notes in Computer Sciences n° 1565, pp. 294-308, 1998.