Teaching Conceptual Modeling in ER: Chen Worlds

Natalya G. Keberle[1] and Ivan V. Utkin [1, 2]

[1] Zaporozhye National University, Zhukovskogo st. 66 69063 Zaporozhye Ukraine
nkeberle@gmail.com
[2] Prima Development Group
vanjonito@mail.ru

Abstract. Whilst algorithmic modeling is taught intensively both in school and higher education, conceptual modeling, or modeling of data to be used by algorithms, is less highlighted in the teaching curricula. However, understanding basic conceptual modeling principles plays a very important role in practice, as the cost of wrong solutions taken at the level of conceptual modeling is usually high. Tools, accelerating learning of conceptual modeling, are rare, at least freely available or mentioned in the literature. We present our work in progress – a system called "Chen Worlds" reflecting the focus on ER paradigm of conceptual modeling, describe its use cases, architecture and technical solutions undertaken.

Keywords. Conceptual modeling, entity-relationship diagram, teaching ER

Key Terms. TeachingProcess, TeachingMethodology, ConceptualModeling

1 Introduction

A course in databases and information systems, even introductory, pays attention to the architecture of a database system and to the process of a database system building, part of which is conceptual modeling. The analysis of student works in conceptual modeling for databases, particularly in ER notation [1], has envisaged a set of common mistakes, coming from either total misunderstanding of ER notation (rather seldom, nevertheless), or from misconception of specific use cases. Provision of detailed feedback on resolving misconceptions for interested students raises the syntactical validity of their ER diagrams produced essentially. Without clear understanding of the syntax of ER notation it is hard to produce conceptual models which correctly reflect a domain at the level of semantics. We acknowledge that teaching semantically correct conceptual modeling requires much more time for practice, and restrict ourselves with a modest aim – to create an easy-to-use and friendly environment to learn syntax of ER diagrams.

The paper is structured as follows: in the Section 2, we describe and classify patterns of misconception of ER notation, detected during the analysis of students' works. Section 3 presents the related work and fruitful ideas inspiring the presented approach to teaching ER. The architecture of the system "Chen Worlds" and the use cases are sketched in the Section 4. Section 5 presents concluding remarks.

2 Common Mistakes in Understanding ER

The notation of ER uses small and simple set of basic constructs: entities, attributes (including primary, optional, multiple), relationships, weak entities and generalizations. Well known complex entities – aggregations and associations are derived from this set.

All the mistakes in understanding ER are divided into syntactic and semantic. Unsurprisingly, usage of incorrect syntax is conditioned by a poor understanding of a domain modeled. However, we have assumed that domain modeling skill requires some experience and time, and during the introductory course we can only teach some well known heuristics for semantically correct conceptual modeling.

Let's illustrate the idea with typical examples, taken from works of students. The absence of a primary key for an entity is a syntax mistake (see Fig. 1a), whereas usage of single-valued attribute as a primary key for an entity having all other properties as multiple-valued attributes shows a semantic misconception of a domain (see Fig. 1b).

There are more ambiguous examples, like the one depicted on Fig. 1c. Non-differentiation between the values of a property (e.g. "rain", "snow', "hoar-frost" etc. are some values of a property "name" of an entity "Precipitation") and composite properties (e.g. "address" is a composite property with atomic parts "city", "street", "house', "apartment", "postal code") could be a semantic misconception.

However, under certain circumstances, namely, assuming that the values of properties "rain", "snow", "hoar-frost" are ranging in Boolean data type, the diagram, depicted on Fig. 1c, could be semantically correct.

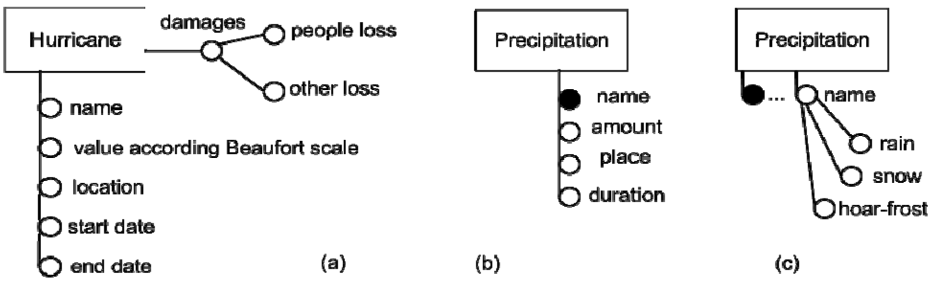

Fig. 1. Some typical mistakes in ER diagrams: (a) – no primary key; (b) – single-valued primary key with multi-valued non-key attributes; (c) – usage of attribute values as attributes.

From the syntax of ER notation and following the analysis of the students' works in conceptual modeling, we have constructed a list of core syntactical misconceptions, a notable part of which is given below:

1. If two different entities possess exactly the same set of attributes, they could be merged.

2. Duplication of attributes across several entities is not allowed, especially if the attributes are foreign keys.
3. Primary key should be introduced for every concept in an ER diagram.
4. Primary key attributes cannot be optional, they should be mandatory.
5. Primary keys of relationships are not allowed on an ER diagram.
6. Cardinalities of a relationship are from the set {1, M}.
7. Participation cardinalities (optionality) of a relationship are from the set {0,1}.
8. A relationship cannot be directly related to another relationship.
9. An entity cannot be directly related to another concept.
10. Composite properties should have as parts only properties, not values of properties.
11. Weak entity cannot be related to strong entity with cardinalities, different from (1,1):(1,M).
12. Participation cardinalities for aggregate entity and its parts are to be mandatory.
13.

To resolve at least these misconceptions and to help tutor and students in understanding conceptual modeling in ER at least at the level of syntax we exploit several techniques, described in the next section.

3 Accelerating Learning, Root Questions and Gaming Environment

According to the review of motivations for achievements in mathematics [2]: "…If students realize that their successes are meaningful and result both from their abilities and from a high degree of effort, they are likely to believe that they can do mathematics if they try…". Understanding conceptual modeling from our point of view is more to efforts put to understand the notation first and to apply it for different domains.

The idea of accelerating learning when teaching conceptual modeling as part of a course in databases and information systems has roots in one of the works of Professor Emeritus Jeffrey Ullman[1] at Stanford University. Prof. Ullman with colleagues had created Gradiance On-Line Accelerated Learning System[2] – a service for on-line training in solving simple-to-tricky exercises in various fields of Database Systems, Compilers, Automata Theory and Operation Systems.

Following the idea of a root question [3] they had introduced sets of exercises, addressing the main problems of the particular field of study, sets of possible mistakes, and sets of hints (and in some cases, solutions) to avoid those mistakes.

The idea of immersion of learning to gaming environment is widely used in practice for various fields of study. Of particular interest are the systems for teaching programming, like ALICE [4] – for object-oriented programming, PictoMir [5] and KuMir [6] – well known and respected environments aiming at teaching children and secondary school pupils basics of algorithmic thinking. The main distinguishable

[1] http://infolab.stanford.edu/~ullman/
[2] http://www.gradiance.com/

feature of such systems is immediate visualization of the choices a scholar makes, and consequent visualization of the solution by the system.

For example, in PictoMir there are: Environment – a part of the Universe, usually, a plain surface made of squares, Robot-Performer – a robot, able to move on that surface one step back or forth, rotate 90 degrees to the left and to the right, and Learner – usually a child, that carries out a task, e.g. "Let Robot-Performer fill in with some color all the squares at the corners of the surface", writing an algorithm of the kind "Move 1 step forth, Fill the square, Turn 90 degrees left, Move 1 step forth…". Algorithm writing is also replaced with picking up the proper symbol of the robotic language, like "left arrow", "turn 90 degrees left arrow" and so forth. The solution proposed by a scholar is executed "as is" step by step, and it's easily seen at what step of the algorithm Robot-Performer fails.

For Conceptual Modeling in ER we initially have a set of graphic primitives and a set of connection rules for primitives. Adopting the idea of a root question we propose an environment for building, visualization and validation of conceptual models in ER notation, described in the next section.

4 Chen Worlds: Use Cases, Architecture and Technical Solutions

Chen Worlds – is a cross-platform software system for learning, building, visualization and validation of conceptual models in ER notation.

We have identified two main roles of actors in Chen Worlds: a Scholar that is a person, who learns conceptual modeling for databases, and a Tutor, who prepares teaching materials.

A Scholar use case

A Scholar uses GUI to access the system. His/her aim is to obtain certain knowledge on how to use Chen notation in conceptual modeling for databases. The system proposes a Scholar a set of examples, each of which is oriented on answering of one root question.

Each example is presented as a text, shown to a Scholar, or as a picture, depending on the type of question. There are one or several correct answers (there are situations when several solutions are correct in Chen notation), which are usually not shown to a Scholar until he or she submits at least one solution. The questions are created according to the root question technique [3].

A Scholar may also use the system as a guide to notation, choosing a theme from a list of themes dedicated to conceptual modeling in ER.

A Tutor use case

A Tutor uses either GUI or usual text editor to create examples of ER diagrams. His/her aim is to construct different examples, demonstrating the applicability of a particular problem addressed with a root question. A Tutor edits a set of rules for detection of core misconceptions (see Section 2).

The architecture of the system is presented in Fig. 2. The system consists of VISUALIZER, SOLVER, PARSER and GUI components.

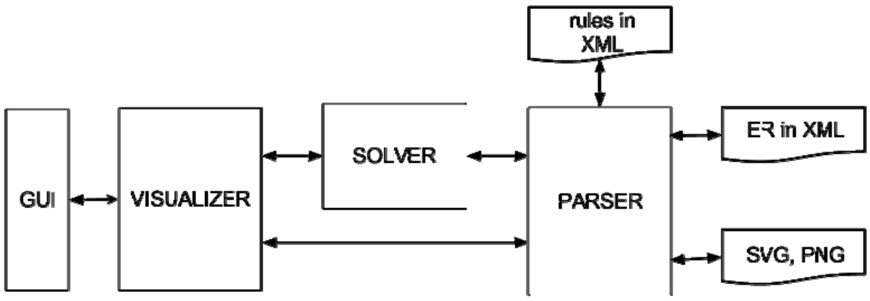

Fig. 2. Architecture of Chen Worlds system.

VISUALISER shows a task (a text, a picture, or both), shows a hint in the text of a task, and shows a hint in a picture. VISUALIZER provides a palette of graphical primitives of Chen notation.

In order to correctly present an ER diagram after reading its XML encoding, VISUALISER performs initial layout task, assigning each entity, relationship and generalization particular absolute places on a working space. Attributes, cardinalities and optionalities are placed relatively to entities/relationships they belong to.

With respect to the user role, VISUALISER may consult SOLVER and restrict the applicability of elements of the ER notation, avoiding syntactically incorrect diagrams (suitable in Tutor mode to save time). For Scholar mode VISUALISER allows arbitrary combinations of elements.

A solution submitted by a user is parsed into XML presentation and evaluated by SOLVER. SOLVER compares a solution submitted by a Scholar with respect to a correct answer of the task, detects mistakes, using a set of predefined rules, already created by a Tutor, and depending on which rules are violated, returns to VISUALISER a hint code for a user. If several mistakes are appeared in one submitted solution, SOLVER first returns hint code of highest priority (which means the most serious mistake), and in case a Scholar correctly resolves the addressed mistake, re-checks the rules again.

PARSER reads a file of a task (in XML), saves a solution (both in XML and in graphical presentation) created by a Scholar, saves a task (both in XML and in graphical form) created by a Tutor, reads a file of rules for detecting mistakes.

Examples of tasks are written as XML documents with an XML schema, substantially extending developed in [7] XML Schema with cardinalities, optionalities and generalizations. Additional XML Schema was developed for presentation of rules for detecting mistakes.

5 Concluding Remarks

Understanding conceptual modeling plays an important role in building useful and extensible database systems. There are many tools facilitating the process of construction of conceptual models (e.g. Computer Associates ERWin Data Modeler, IBM Rational, a lot of others), but most of them just use correspondent notation (e.g. IDEF1X, Crow's Foot, UML etc.) and do not teach it. Tools facilitating the process of learning conceptual modeling are rare.

Taking classical ER notation as an example it is shown that there exist both syntactic and semantic misconceptions of the notation itself, blocking its proper usage and leading to serious problems in future database schema construction.

Chen Worlds system is currently oriented on teaching classical ER notation, however the principles of the system could be applied to other complex graphical notations, e.g. IDEF1X, UML Class diagrams, as the hardest part of their teaching is in preparation of sets of core syntactic (and in general, semantic) misconceptions.

References

1. Garsia-Molino, H., Ullman, J.D., Widom, J.: Database Systems: the Complete Book. Pearson Prentice Hall, 1203 p. (2009)
2. Middleton, J. A., Spanias, P.A.: Motivation for Achievement in Mathematics: Findings, Generalizations, and Criticisms of the Research. J. Research in Mathematics Education, 30(1), pp. 65--88 (1999)
3. Ullman, J. D. Gradiance On-Line Accelerated Learning Guide for Authors. http://www.gradiance.com/downloads/auth-guide.pdf
4. ALICE, http://www.alice.org
5. PictoMir, http://www.pictomir.ru
6. KuMir, http://www.niisi.ru/kumir
7. Jin, S., Kang, W.: Mapping Rules for ER to XML Using XML schema. In: Proc. 10th Southern Association for Information Systems Conference. Jacksonville, Florida, USA, 9-10 March, 2007, pp.211-216 (2007)