MAPPING COMMON ERRORS IN ENTITY RELATIONSHIP DIAGRAM DESIGN OF NOVICE DESIGNERS

Rami Rashkovits[1] and Ilana Lavy[2]

[1]Department of Management Information Systems, Peres Academic Center, Israel
[2] Department of Information Systems, Yezreel Valley College, Israel

ABSTRACT

Data modeling in the context of database design is a challenging task for any database designer, even more so for novice designers. A proper database schema is a key factor for the success of any information systems, hence conceptual data modeling that yields the database schema is an essential process of the system development. However, novice designers encounter difficulties in understanding and implementing such models. This study aims to identify the difficulties in understanding and implementing data models and explore the origins of these difficulties. This research examines the data model produced by students and maps the errors done by the students. The errors were classified using the SOLO taxonomy. The study also sheds light on the underlying reasons for the errors done during the design of the data model based on interviews conducted with a representative group of the study participants. We also suggest ways to improve novice designer's performances more effectively, so they can draw more accurate models and make use of advanced design constituents such as entity hierarchies, ternary relationships, aggregated entities, and alike. The research findings might enrich the data body research on data model design from the students' perspectives.

KEYWORDS

Database, Conceptual Data Modelling, Novice Designers

1. INTRODUCTION

The database is an essential part of almost every business software system. It is responsible for managing the system's data, including input validation and integrity, applying business rules, and the source for various business reports. Improper design of the database will inevitably lead to functionality errors; hence a proper design of the data model is crucial [1].

Many implementations of databases were developed, however, the relational model [2,3] is the most common, and the primary choice for most software systems. The relational model requires the definition of tables each consists of records containing various data fields to describe business entities such as products, customers, and alike. Records relate to each other using key fields that are a subset of the table's fields.

To design a good relational schema, (i.e., tables, fields, and keys), one needs to understand the system's requirements as stated by the customer. These requirements' specifications usually refer to possible user-system interaction scenarios specifying data inputs and outputs. The data model is then extracted from these requirements, to support the specified system functionality [4].

Entity-Relationships-Diagram (ERD) is a visual model vastly used to describe business entities, their attributes, and their relationships with each other, introduced by Chen [5]. Enhanced-ERD

(EER) is an extension of the basic model, including more design concepts such as super and sub-type, and inheritance constraints. Sketching an ER (Entity-Relationships) model is usually the first step in database design. Upon completion, The ER sketch is then translated into a logical data model using translation rules [6,7]. The result is a set of tables, fields, primary and foreign keys, realizing the relationships between records according to their participation constraints. The data model is then tested for normalization, to ensure minimal redundancy.

Barta and Antony [8] found that novices can identify entities and attributes correctly but encounter many difficulties regarding cardinality and connectivity of relationships. Antony and Batra [9] have suggested a consulting tool to assist novice designers with the construction of the ERD. According to Batra [10], novice designers encounter many difficulties, mostly concerned with cognitive complexity; among them, are flexibility for errors, lack of immediate feedback, and information overload. As a result, data models designed by novice modelers tend to be inaccurate and erroneous. Enhanced ERD model includes variety of new elements such as hierarchies, aggregations and weak entities, which their proper use was not explored in previous research. These complex elements may cause novice designers many difficulties, and result in bad database designs.

In the current research, we rexamine novices for proper identification of entities and relationships as well as exploration of more advanced ERD concepts such as entities' aggregations entities' hierarchies. And relationships between strong and weak entities. We analyse database designs of novice designers according to Anderson et al. [11] that formulated their version for levels of understanding based on the SOLO taxonomy [12]. We classified Novice designers' knowledge and understanding of the ERD model into the taxonomy levels, and measured their designs accordingly. We believe that identification of common errors and discussion of their causes and characteristics may be valuable for educators and practitioners.

The research questions are:

(1) What are the common errors novice designers make when designing a data model?
(2) What is the distribution of errors?
(3) What are the underlying reasons for these errors and how can they be avoided?

In what follows we present a brief theoretical background and related works, followed by the course of the study, and the obtained results and discussion. We continue with instructional recommendations ending with concluding remarks.

2. THEORETICAL BACKGROUND

In this section, we provide a brief theoretical background to address the research framework. We first provide a brief description of Anderson et al. [11] revised taxonomy referring to levels of understanding [13] as a theoretical framework to understand the sources of the students' difficulties. Then, we discuss the complexities lies in the process of database design, followed by short description of the SOLO taxonomy for levels of understandings. Finally we provide a short survey of related work.

2.1. The complexity of Database Design

Database design is a complex task, certainly for novice designers. The design process is usually based on written descriptions. Such description can be blur and may not contain all possible uses of data, hence may contribute to bad database design. However, even when descriptions are clear,

comprehensive and accurate, database design is still difficult. While novices find entities not difficult to model they face more difficulties with relationships [8], especially when ternary relationships are involved [8,14]. The difficulties in identifying the correct set of relationships among the entities specified lies in the potentially large number of possible relationships. The designer has to identify those relationships that maintain the semantic without redundancy and keep a degree of related entity to the minimum. Doing so improves the chances to create the ultimate single data model appropriate for the applications built on.

Yet another obstacle novice designer has to overcome is the identification of weak-entity-sets, containing entities that depend on other entities for their existence [14]. Identifying weak entities as regular ones would result in an erroneous solution, in which records of the weak type would not have a primary key to distinguish one from another. For instance, if a primary key of Award would consist only by category and year, we will not be able to distinguish between <'best picture', 2016> given by Academy Awards and the award with the exact same name and year given by Cannes Festival. Adding the organization name (the key attribute of the Organization table) solves the problem, enabling the identification of each award separately.

The hierarchy of classes is also difficult to identify by novice designers [16,17]. The designer must capture all the similarities and differences of the entities involved and design a proper hierarchy in which all subtypes inherit all attributes and relationships of the supertype. Some designers may decide on flatting the hierarchy by merging similar entity-sets unifying their attributes, or deepen the hierarchy by splitting entities into subtypes over trivial attributes (e.g., married employees are separated from single ones to capture the spouse's name). The result of improper hierarchy would be a data model either containing many null values (for irrelevant attributes of merged entity-sets) or containing redundant tables.

Converting the ER model into a relational model contains additional potential errors. While entities and relationships are easy to convert, the conversion of a hierarchy of entity-sets may be confusing. Novice designers may miss the overlap/disjoint label or the partial/total participation of the inheritance relationship, and create redundant tables, with possibly duplicate attributes.

In this research, we would like to reassure the findings in [14] concerning ternary relationships and explore more types of errors done by novice designers related to unary relationships, weak entities, and the hierarchy of entity-sets.

2.2. Levels of understanding

Basing on bloom's taxonomy [13], Anderson et al. [11] formulated their version to Bloom's levels of understanding: Remembering: Retrieving, recognizing, and recalling relevant knowledge; Understanding: Constructing meaning from written, and graphic messages through interpreting, exemplifying, classifying, summarizing, inferring, comparing, and explaining; Applying: Carrying out or using a procedure through executing, or implementing; Analyzing: Breaking material into constituent parts, determining how the parts relate to one another and an overall structure or purpose through differentiating, organizing, and attributing; Evaluating: Making judgments based on criteria and standards through checking and critiquing; Creating: Putting elements together to form a coherent or functional whole; reorganizing elements into a new pattern or structure through generating, planning, or producing.

In the research literature, there are several taxonomies by which learning processes and levels of understanding are classified. Biggs and Collis [12] developed a system for classifying the quality of students' work, known as the SOLO taxonomy. The main advantage of the SOLO taxonomy, in relation to other educational hierarchies, is its generality: it is not content-dependent, making it

usable across several subject areas. The SOLO taxonomy has five levels of understanding that can be encountered in learners' responses to academic tasks [18]:

1. Prestructural — the task is not accessed appropriately, and/or the student has not understood the task;
2. Unistructural — one or several aspects of the task are picked up and used (level of understanding is nominal);
3. Multistructural — several aspects of the task are learned but are treated separately. The student still lacks the "full picture" (understanding is equivalent to knowing about);
4. Relational — the task's components are integrated into a coherent whole, with each part contributing to the overall meaning (understanding in the form of appreciating relationships);
5. Extended abstract — the integrated whole at the relational level is reconceptualized at a higher level of abstraction, which enables generalization to a new topic or area. Namely, the whole derived at the previous level is conceptualized at a higher abstract level so that it can now be used in different settings (understanding as a transfer of concept and as involving metacognition).

The SOLO taxonomy has been used fruitfully to classify students' work and to identify approaches used in the area of learning course material in post-secondary school settings. For these reasons, this research utilized the SOLO taxonomy to assess students' levels of learning. We used the SOLO taxonomy due to the objective criteria that it provides for measuring students' cognitive abilities [18]. Students' knowledge and understanding of the ERD model was accrued incrementally, in a similar way to the measures in the taxonomy.

2.3. Related Works

Huang [19] tested students' performances to experts when modeling data. It was found that without adequate domain knowledge, modelers cannot perform well in conceptual modeling no matter how good the fit is between problem domains and modeling techniques.

Leung & Bolloju [20] classified errors frequently committed by novice systems analysts in developing domain models using the Unified Modeling Language (UML). Their results include errors grouped into different quality categories and some observations on relationships among errors from different categories.

Moody & Shanks [1] developed a framework for evaluating the quality of data models and choosing between alternative representations of requirements. For any set of user requirements, there are many possible models, each of which may have different implications for database and systems design. In the absence of formally defined and agreed criteria, the choice of an appropriate representation is usually made in an ad hoc way, based on opinion. The evaluation framework proposed consists of four major constructs: qualities (desirable properties of a data model), metrics (ways of measuring each quality), weightings (relative importance of each quality), and strategies (ways of improving data models). Using this framework, any two data models may be compared objectively and comprehensively. The evaluation framework also builds commitment to the model by involving all stakeholders in the process: end-users, management, and the data administrator and application developers.

In his paper, Kesh [21] describes the development of a model, associated metrics, and methodology for assessing the quality of the ER model. The model was developed by investigating the causal relationships between ontological and behavioural factors influencing data quality. The methodology describes the aggregation of the scores on various metrics to

calculate an overall quality score for an ER model, and use of the model to identify problem areas if the individual quality scores on different factors do not meet organizational standards. Further possible improvement of the model and future research issues are also discussed.

The studies above focused mainly on the design outcome and the classification of the design error. None of the studies explored the difficulties leading to the design faults and their roots. Most of the previous research focused on basic elements of the ERD, namely, entities and relationships. More advanced concepts such as weak entities or aggregated entities were not explored. Hierarchies of entities were explored in the context of UML designs, aiming at software classes, not necessarily related to database design. The current study addressing the aspects of difficulties and their roots, and explore novice designers understanding of more enhanced ERD concepts as well as the basic ones.

You are asked to design a data model to store information about the film industry. The data model should refer to production companies, movies, cast (actors), crew (people hired to produce movies such as directors, make-up artists, camera operators, etc.), awards, and reviews. A production company (Studio) has a unique name, open and closing years. Studios produce various movies, on each we store title, short description, year, and genre (e.g., action, comedy). Titles of multiple movies may overlap, however, not at the same year. Some movies are part of a series (e.g., Harry Potter, The Matrix) and for such movies we keep precedence between one another. On each movie we store full cast and crew, and for each one we keep the role performed in the movie, and the pay received. The roles by the crew and cast are extracted from a fixed list (e.g., director, costume designer, leading actress, supporting actor), each role has name and description. Cast and crew individuals can take part in various movies, performing different roles. One can even play multiple roles in the same movie (e.g., Clint Eastwood was the director and leading actor in the movie 'Million Dollar Baby'). We store on each individual (cast or crew) name, gender, and date of birth. There may be many individuals sharing identical names, however, the combination of name and date of birth is unique. On cast we store a list of photos, on crew we store recommendations. Individuals (crew and cast) may be awarded for the role they performed in a particular movie. The awards are given each year by various organizations (e.g., Academy Awards, Cannes Festival) to movies and individuals in various categories (e.g., best actor, best screenwriter, best picture, best comedy). Each organization has name and website. Each organization may provide a single award each year in each category, however, movies and individuals can be awarded more than once, in multiple categories and multiple organizations. Award categories can be shared by multiple organizations (e.g., 'best actor', 'best movie'). Reviews are given by film critics on which we store name, short biography and a photo. The review itself comprised of rank of 1-5 scale and a textual judgement.

Suggest a relational model that optimally captures all the information above. Use Entity Relationships Diagram for the design phase.

P.S. Do not add additional attributes.

Figure 1: The problem

3. THE STUDY

In what follows, we present information regarding the study participants, the data collection, and analysis methods.2.6. Section and sub-section headings

3.1. The Study Participants and the Course of the Study

The study was conducted within the "introduction to database management systems" course. This course is part of the Information Systems (IS) curriculum [22], and given as a mandatory course in the second academic year (out of three) in the Information Systems (IS) department in a regional academic college. During the course the students are becoming acquainted with the principles of designing database models according to the following phases: (1) Conceptual design captured via ERD; (2) Logical design captured by the relational model; (3) Physical design implemented using files and indexes. After these phases, the students learn SQL commands to manipulate data stored in the database. As to the learning of conceptual design, the students are exposed to the symbols used in the ERD diagram and the meaning of each, while practicing a few case studies followed by a class discussion.

Sixty-five MIS second-year students (in the academic year of 2018-2019) participated in the study. Approximately two-thirds of the students were males, and the rest were females. To be able to receive a genuine picture as regards the students' difficulties in applying the concepts of the ERD design, the students were asked to provide a conceptual model addressing a given requirement in the final exam. Students do their best to succeed in the exam and hence they make many efforts to provide the best design they are capable of. All students' provided solutions were considered for this study, and the students are considered a good representative of novice designers.

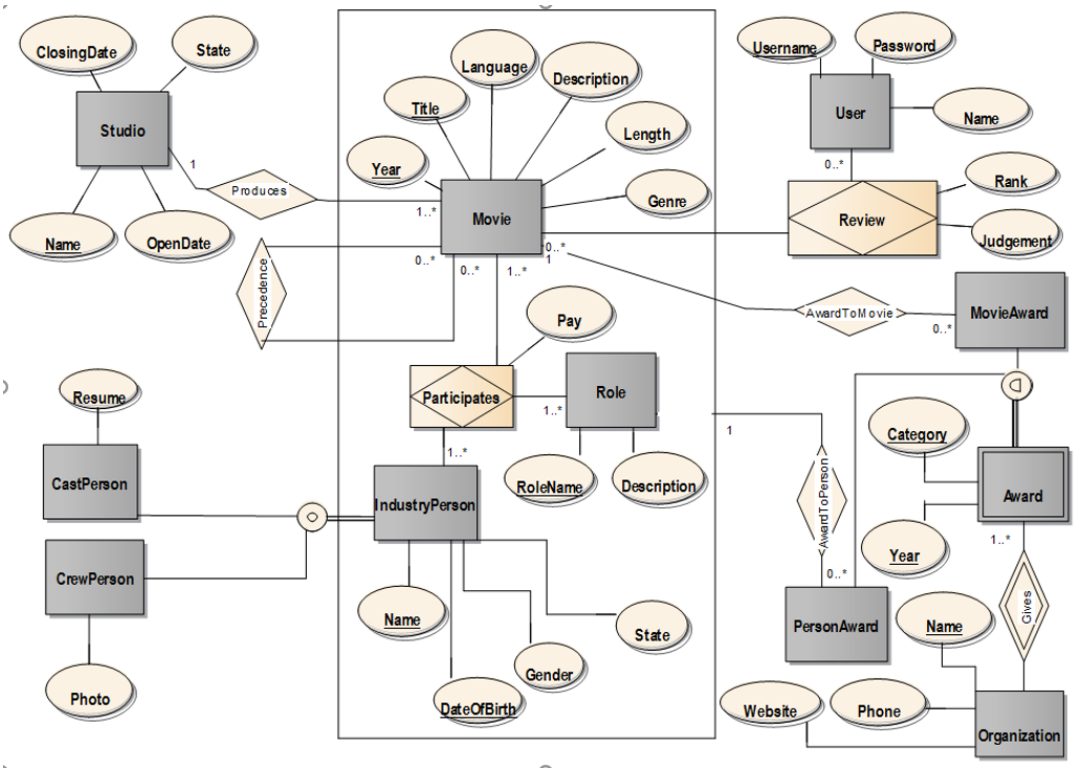

Figure 2: Expected ERD solution

Studio (sName, country, oDate, cDate)
Movie (title, mYear, language, description, length, genre, sName)
IndustryPerson (pName, dob, gender, pState)
CastPerson (pName, dateOfBirth, resume)
CrewPerson (pName, dateOfBirth, image)
Role (rName, GoingWage)
Organization (oName, phone, website)
PersonAward (aName, aYear, aCategory)
MovieAward (aName, aYear, aCategory)
User (username, password, uCountry)
Participates (pName, dateOfBirth, roleName, title, mYear)
Precedence (title, mYear, preTitle, preYear)
AwardsMovie (aName, aYear, title, mYear)
AwardsPerson (aName, aYear, pName, dateOfBirth, title, mYear)
Review (username, title, mYear, rank, text)

Figure 3: Expected Relational Model

3.2. The problem

The problem provided to the study participants is shown in Fig. 1 followed by the expected E/R Diagram shown in Fig. 2, and the expected relational model is shown in Fig. 3.

3.3. Data Collection and Analysis Tools

The analysis process was done in three phases. In the first phase, we made a list of all elements of the ER diagram and classified them into three categories based on their complexity: basic elements (category A), advanced elements based on basic ones (category B), more advanced elements with interconnections between other elements (Category C). the ERD elements and their classification are presented in Table 1.

Based on the above three categories we classified the ERD-related errors according to the SOLO taxonomy [18] to various levels of understandings, as presented in Table 2.

Based on ERD constituents detailed in Table 1, we made a list of all possible design errors according to the 'semantic level' of Lindland et al.'s framework [23]. Next, we built the IMDB problem stated above, which requires the inclusion of all ERD constituents listed in Table 1 for a proper solution, as shown in Figure 1. A list of all ERD elements that should be part of a proper solution is listed in Table 3. The IMDB problem was then provided to the students during their final exam.

The second phase started after collecting the students' solutions. We analyzed the solutions according to the following categories: (1) completeness – where all requirements referring to data stated in the question addressed? (2) correctness – were these requirements addressed properly using adequate ERD elements?

During the analysis, we delved into each of the students' solutions to search for errors related to any of the ER elements listed in Table 1. Only the first occurrence of each error was counted and any multiple occurrences of this error belonging to the same solution were ignored.

Table 1: Classification of ERD constituents

Category	Description	Explanation
A	ERD elements relating to elementary concepts of the model, including attributes, entities, binary relationships, binary relationships' cardinalities, key attributes, and relationship's attributes	These elements are considered to be basic knowledge of the ERD construction
B	ERD elements relating to advanced concepts of the model, including reflexive and n-ary relationships, and their cardinality constraints	These elements require the understanding of the basic concepts of the previous category as a prerequisite
C	ERD elements relating to more complicate concepts, including weak entities, hierarchies of entity-sets, and aggregations	These elements require the understanding of elements defined in categories A and B, and the interconnections between them

Table 2: SOLO levels of understanding in the ERD context

Level of Understanding	Description
Prestructural (1)	The student fails to identify properly most of the problem's basic elements listed in category A.
Unistructural (2)	The student succeeds to identify properly most of the problem's basic elements listed in category A, however, he fails to provide a proper solution to most of the ERD elements listed in categories B and C
Multistructural (3)	The student succeeds to identify properly most of the problem's elements listed in categories A and B, however, he fails to provide a proper solution to most of the ERD constituents listed in category C
Relational (4)	The student succeeds to identify properly most of the problem's elements listed in categories A, B, and C.
Extended abstract (5)	This high level in the SOLO model referring to reconceptualize and transfer of the learned concepts was not addressed in this study

Table 3: ERD elements included in the expected solution

Component	Frequency
Strong Entity	6
Weak entity	1
Attributes	29
Binary relationship	5
Binary relationship with attributes	1
Ternary relationship with attributes	1
Reflexive relationship	1
Hierarchy (one disjoint and one overlapping)	2
Aggregation	1

After completing the analysis, we started the last phase, in which we interviewed ten students chosen according to their provided solution which included various kinds of errors. The purpose of the interviews was to get to the roots of the errors made by the students. Using analytic induction [24] and content analysis [25] we classified the interviews' transcripts into the categories stated in Table 1 and provide possible explanations for the errors made by the students basing on the SOLO taxonomy.

4. RESULTS AND DISCUSSION

In what follows we present results referring to the students' errors in database design and analyze them according to the SOLO taxonomy presented in the previous section. Then we discuss the underlying reasons for the difficulties from the students' point of view as were emerged from the interviews conducted with a representative group of them.

4.1. Classification of Errors

In the analysis process of the ERD solutions provided by the study participants, we found various errors in each of the categories stated in Table 1. In what follows we present these errors and their frequencies as was found in the students' solutions.

4.1.1. Category A

Table 4 includes the list of category A errors and their frequencies.

Table 4: Category A - distribution of errors

Error	Unexpected but presented	Missing but expected
Strong Entities	20(31%)	19(30%)
Binary relationships	0	16 (25%)
Attributes	3(5%)	15(34%)

Forty-five students (70%) succeeded to provide the complete list of expected strong entities. All of the 19 students (30%) who provided a solution with an incomplete list of entities did not include the 'Role' entity in their solution, and among them, five students did not include the 'User' or 'Award' entities.

Twenty students (31%) added unexpected entities to their solution. Seventeen of them added the 'Movie Series' entity and connected it to 'Movie' using a one-to-many relationship (as shown in Figure 2). They should have modeled the precedence of a series of movies using a reflexive relationship between 'Movie' and itself. Interestingly, none of them did provide any attribute to this unexpected entity.

Three students added the unexpected 'Review' entity with relationships to 'Movie' and 'User' (as shown in Figure 4). They also added an artificial primary key 'no.' to identify each review instance, which explains the five percent of the students who provided unexpected attributes. They should have modeled the review as a relationship between 'Movie' and 'User', with attributes (rank and review), as shown in Figure 1.

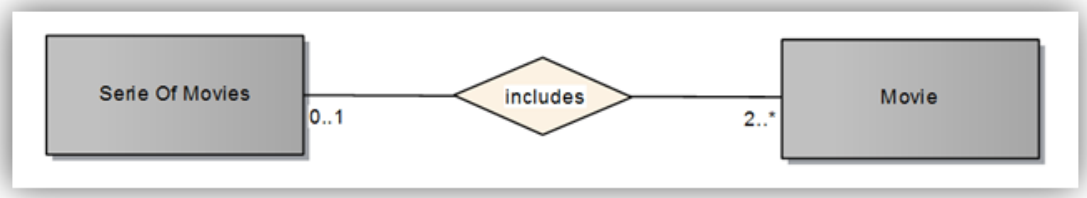

Figure 3: Erroneous interpretation of the reflexive relationship

Forty-Eight students (75%) succeeded to provide the complete list of expected binary relationships. The other 16 students provided four (out of five) binary relationships. Most of these students did not include the relationship between 'Award' and 'Movie'. It is worth noting that almost all of the students succeeded to identify correctly the cardinality constraints of all binary relationships they defined.

As to listing attributes according to the problem requirements, fifty students (79%) succeeded to provide the complete list of the expected ones while the majority of the other 14 students missed one or two of them.

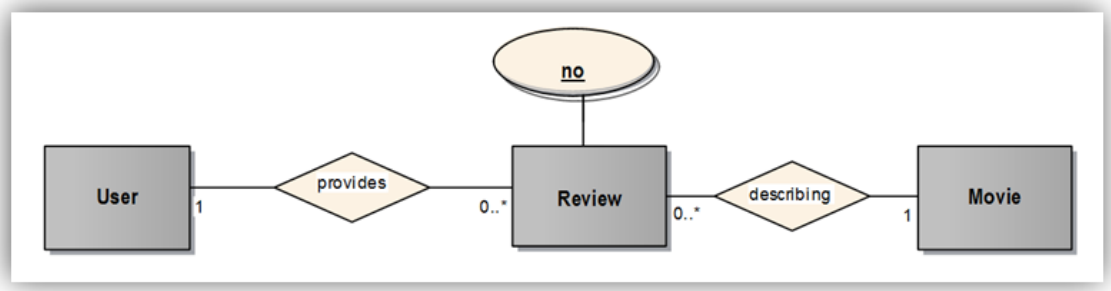

Figure 4: Entity instead of a relationship

4.1.2. Category B

Table 5 includes the list of category B errors and their frequencies.

Table 5: Distribution of the students' category B errors

Error	Unexpected but presented	Missing but expected
Reflexive relationship	0	53 (83%)
Ternary relationship	0	51 (80%)
Key attributes	1(1.5%)	10 (16%)

Only 11 students (17%) provided a solution including the reflexive relationship between the movie and itself to capture the precedence concept. Among the other 53 students (83%), 17 students added erroneously 'Movie Series' entity as explained above, and the other 26 students did not refer in their solutions to the precedence requirement at all.

Only 13 students (20%) provided a solution with a ternary relationship between 'Movie', 'Movie Industry Person' and 'Role' as shown in Figure 1. Among the other 51 students (80%), 19 students did not define 'Role' entity and connected 'Movie' only with 'Industry Person', and 32 students modeled two binary relationships between 'Movie', 'Industry Person' and 'Role' entities as shown in Figure 5. In this solution, one can assign different roles to various industry persons, and assemble different industry persons to various movies. However, it cannot be inferred what is the role of each movie industry person in a specific movie. Hence, this solution does not address the problem requirements.

It is worth noting that all the 12 students who defined ternary relationships between the above three entities, also succeeded to identify correctly the cardinalities of the entities (n:m:k).

Fifty-four students (84%) defined successfully the key attributes of all the entities they defined. Among the other 10 students (16%) most of them did not specify key attributes in one or two entities.

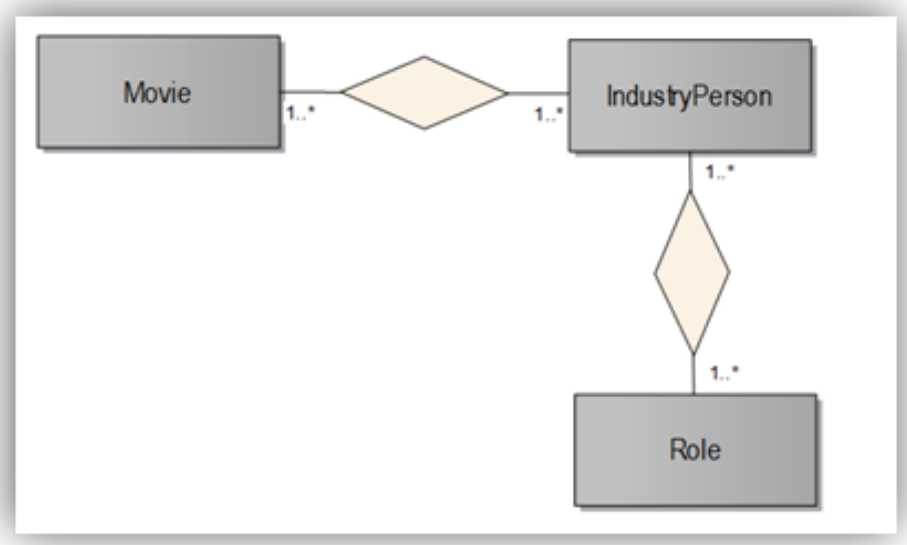

Figure 5: Erroneous replacement of ternary relationship by binary relationship

4.1.3. Category C

Table 6 includes the list of category C errors and their frequencies.

Table 6: Distribution of the students' category C errors

Error	Unexpected but presented	Missing but expected
Weak entities	0	56(88%)
Hierarchies	7(11%)	55(86%)
Aggregations	0	53(83%)

As to weak entities, fifty-six students (88%) provided a solution that does not specify a weak entity in their model. As shown in Figure 1, the 'Award' entity is modeled as a weak entity, depending on the 'Organization' entity. Although the 'Award' entity appears in many of the students' solutions, it appears as a regular entity and not as a weak one, ignoring its dependency on the organization. The students specified the attributes 'name' and 'year' as a combined key, though the problem requirements specifically declare that various organizations can provide awards with the same name each year.

As to hierarchies, fifty-five students (86%) failed to provide correct hierarchies as shown in Figure 1. Among them, 50 students identified correctly the cast-and-crew hierarchy, while only five provide a faulty solution for that hierarchy. Two faulty solutions added 'Actor' 'Director', 'Photographer', and other roles as sub-entities of the 'Industry Person' instead of 'Cast' and 'Crew' sub-entities, two others provided two unrelated 'Crew' and 'Cast' entities with duplicate attributes and one provided one 'Industry Person' including all attributes of both cast and crew. However, 55 students (including all the 50 students who provided correct cast-and-crew hierarchy) failed to

model the award hierarchy (see Fig. 1). Instead, they provided a variation of the solution shown in Fig. 6, in which the 'Award' entity is connected via two relationships to 'Movie' on one hand, and the aggregation around participation relationship on the other hand.

As to aggregations, fifty-three students (83%) did not provide solutions to address an aggregation. Instead of using aggregation many of them made a binary relationship between 'Award' and 'Industry Person', with 'role' as an attribute, similar to the relationship they drew between 'Award' and 'Movie' (see Figure 6). All the students who did provide an aggregation also specified the correct cardinality constraints (n:m:k).

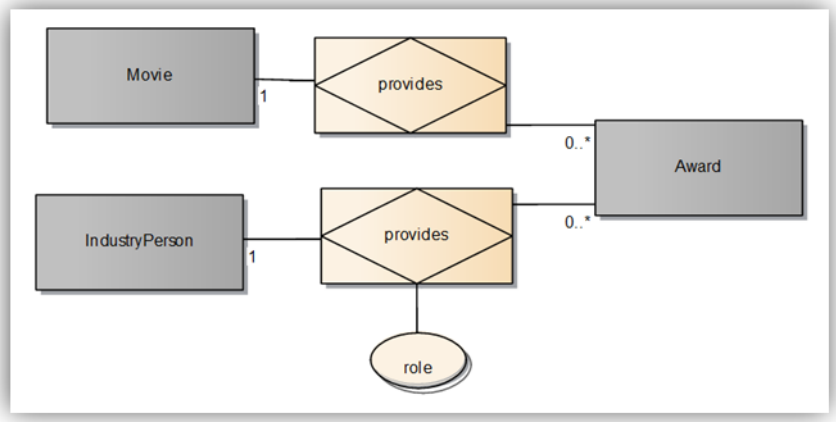

Figure 6: Erroneous replacement of Award concept

4.2. Interviews and Discussion

After scanning all the provided solutions, identifying the errors, and classifying them into the above categories, we conducted interviews with 10 students to gain a better understanding of the reasons underlying the errors made when constructing the solution. The students were chosen upon agreement to participate in the interviews and upon the type of errors found in their solutions. During the interviews, the students were presented with the solution they had provided in the exam and with the correct one. For each error they had done, they were asked to point out the underlying reasons. In what follows we present the interviewees' excerpts and discuss their consequences.

4.2.1. Errors Relating to Elementary ERD Concepts (Category A)

We asked students to illuminate the underlying reasons for omitting attributes, entities, and binary relationships. The following are typical excerpts:

Betty: "I missed 'Role' as an entity. I thought of 'Role' as an attribute of the relationship between 'Industry Person' and 'Movie'. For some reason, I missed the fact that each role is associated with 'going wage'. If I was not stressed during the exams, maybe I would not have done this mistake."

Dana: "I missed the relationship between 'Award' and 'Movie'. I thought by mistake that the relationship I drew between 'Award' and the aggregation of 'Industry Person', 'Movie', and 'Role' covers also the movie's award.

Jonathan: "The IMDB story includes many details. I realize now that when I solved the problem I omitted a few attributes. However, I believe that these omissions are not very significant, and I could easily fix the solution. Maybe if I had more time to evaluate my solution I could find more of them on time."

While attempting to address given problem requirements, the designer sometimes classifies some concepts to the wrong set, and when the design is made under time pressure it is more likely to occur, as can be understood from Betty's excerpt. Dana, however, demonstrates a lack of desire nor the ability to get into details of the model. She was satisfied with one relationship she drew assuming that it covers another relationship because she did not reflect on her solution.

As to the "key" concept, most of the students (84%) identified correctly all the keys identifying the entities in all entity-sets defined by them, while the other students did not specify the key attributes. This may be attributed to the fact that during the database course, a great deal of attention was dedicated to the primary key concept not only during the design phase but also during the implementation (SQL-DDL) and deployment (SQL-DML). As a result, most of the students assimilated the primary-key concept, and use it properly also during the design phase. These results are in line with Leung & Bolloju [20], who found that students can identify correctly the entities and relationships involved. As the model becomes complex and includes advanced concepts such as aggregations and hierarchies, even the basic concepts may also become blurred, and confusions with advanced concepts may occur, as can be observed from Dana's excerpt. From Jonathan's excerpt, it can be revealed that novices sometimes find it difficult to deal with problems that include many small details which add to its complexity. The congestion of details makes it difficult to reveal the overall logic structure of the model. These results can be attributed to the fact that novices tend to believe that the quality of the model is derived from its logical structure and the "small" details such as missing or redundant attributes have a minor impact. As a result, they find the process of reflection on their provided solution to be tedious and redundant [20].

We can conclude that the amount of errors made by the students in this category is the fewest among all three categories. Almost all of the students identified correctly the major entities, attributes, and binary relationships. In terms of the SOLO model [18] we may assume that majority of the students demonstrate a Unistructural level of understanding, which is the basic level of understanding that requires one to be able to detect basic elements of ERD. This result might stem from the intuitive "finger rules' learned in class. They are taught that when analyzing the problem's text to elicit entities, attributes, and relationships, the students applied the rules they have learned in class. Nouns are usually transformed into entities or attributes, and verbs into relationships. The difference between entities and attributes lies in the level of importance of each to the entire model. If one concept is described using many characteristics and relates to other concepts, it is probably an entity, otherwise, it is just an attribute.

4.2.2. Errors Relating to Advanced ERD Concepts (Category B)

As to the underlying reasons for omitting reflexive and ternary relationships, as well as key attributes, the following are typical excerpts from the students' interviews:

David: "I added the 'Movie Series' entity to address the requirement of keeping precedence between movies. The idea of using a reflexive relationship for that purpose just didn't cross my mind. Now I can see that my solution is awkward, since this entity does not have attributes at all, and it assumes that all movies are part of some series, which is an obviously wrong assumption."

Dana: "I didn't refer to the precedence between movies in my solution, since I didn't know how to address this requirement. I spent quite a few moments during the exam but since I could not see how this can be solved I moved on. Even now after seeing the solution, I find it difficult to imagine a relationship between two movies."

Ilan: "Instead of using a ternary relationship, it is much easier to model the problem using two binary relationships (Fig. 5) and avoid higher-order relationships. I find it much more difficult to capture the multiplicity constraints between three participators than it is with a couple."

Ron: "I didn't see the need to attach 'Role' to the participation relationship. Most people play the same role in every movie they take part in. Actors always play, makeup artists do the same job in all movies, and so on. I thought it is sufficient to define a role for each person and that role is dragged automatically to each movie this person is associate with."

Both Dana (who did not refer to the 'precedence' requirement in her solution) and David (who defined the faulty 'Movie' Series' entity) did not consider the reflexive relationship as a possible solution. Dana further claims that she finds it difficult to imagine a relationship between two movies (i.e., a reflexive relationship) which hints at the fact that reflexive relationships are not intuitive to novice designers. Dana's excerpt reveals the complexity embedded in the reflexive relationship. Namely, some of the entities in a reflexive relationship have a dual role. In this case, a movie can be both a successor movie to another one and a predecessor to others. Understanding the above duality requires a Multistructural level of understanding [18]. The lack of ability to cope with concepts necessitating a multi structural level of understanding resulted in the low percentage of solutions addressing a properly reflexive relationship. 41% of the provided solutions there was not found any reference to the requirement 'precedence between movies in a series'.

Many others (27%) provided faulty solutions with a redundant entity of 'Movie Series' connected via 1:n relationship to 'Movie'. The later solution is faulty since adding this entity does not solve the 'precedence' requirement, as it does not impose order on the movies. Moreover, the redundant entity includes redundant attributes such as series name, number-of-movies, etc. It should be specified that many solutions did not specify attributes for this entity at all. Leaving it in an unclear state. These results support the findings in [26] that student modelers had significantly bigger difficulty in identifying unary relationships than expert did.

According to Ilan, it is much more difficult to detect ternary relationships in the text than it is with binary relationships. Indeed, when binary relationships are described, the text usually specifies both related entities and their relationships in the same sentence. However, when it comes to n-ary relationships, the text describing the relationship sometimes spread over a few sentences. Ilan did not see the ternary relationship. He felt more comfortable with two binary relationships, believing that they cover the requirement. Same with Ron, who added information not written in the text to align the solution with his pre-assumptions based on his experience. Often, designers use binary relationships instead of ternary ones, even when it is erroneous. The ternary relationship is perceived as a complex concept since it is more difficult to detect it, and due to its intricate multiplicity constraints. As a result, novice designers tend to avoid its use [14] although this does not always lead to the right solution that meets the requirements of the problem. From the excerpts of Ilan and Ron we may say that they did not perceive the fact that there is a loss of information derived from their binary-relationships based solution which can be attributed to a lack of multistructural level of understanding.

Most of the students (80%) failed to identify the ternary relationship between 'Movie', Industry Person', and 'Role'. While 19 students (30%) may rely on the fact that they did not define 'Role'

entity, 31 students (50%) who outlined 'Role' entity cannot use that excuse. The solution they provided (Fig. 5) is faulty since it is impossible to draw from the model which role an industry person played in a specific movie. It can only tell which roles are associated with industry persons and which industry persons participated in each movie.

We can conclude that the students encountered more difficulties to identify and model elements included in this category than the elements of the previous category. These difficulties stem from the concepts' complexity involved. Namely, to properly use these concepts (i.e. reflexive and ternary relationships) one has to consider all the consequences of the interconnection embedded in them. Reflexive relationships require the designer to think of each entity involved in a dual role, from both sides of the relationship. Novice designer is not always capable to think this way. Ternary Relationships require the designer to think of each couple of entities against the third entity to set the multiplicity constraints, and again it is not a simple task. Such abilities are necessitating multistructural level of understanding [18] that only a small percentage of novice designers possess. Indeed, a key finding is that most of the students did not identify correctly the non-binary relationships. These results are in line with [14] that students encounter difficulties in identifying and modeling non-binary relationships.

4.2.3. Errors Relating to Advanced ERD Concepts with Interconnections (Category C)

As to the underlying reasons for the absence of weak entities, aggregations, and hierarchies in their provided solutions, the following are typical excerpts from the students' interviews:

Ben: "I missed the fact that Award is a weak entity that depends on the organization. It was not clear to me as I read the text. I modeled it as a regular entity with a relationship to the organization. I added an artificial key (running number) that identifies each award, and I still believe that my solution is correct."

Dorit: "I missed the aggregation over the participation relationship. I'm not sure why it is needed. My solution (Fig. 6) connects 'Award' directly to 'Movie' and to 'Industry Person' and to my opinion it covers the problem requirements sufficiently good."

Shimon: "Thought I made a correct hierarchy regarding the crew-and-cast, I missed the hierarchy of the awards. I can see now why my solution (Fig. 6) is faulty but during the exam, I didn't think that something is wrong with my solution and even if someone would suggest hierarchy I would reject it for sure."

From Ben's excerpt, we may learn that weak entities are not easy to detect. The text does not highlight these entities and the designer has to see the problem of setting a primary key without using other entities' key attributes. Even when noticing the problem Ben's choice to solve the problem was by adding an artificial key, although the text said clearly that no further assumptions are allowed. From Dorit's excerpt, we may learn that aggregate entities are elusive concepts, and novice designer finds ways to disregard them.

From Shimon's excerpt, we may learn that hierarchies that are not specified explicitly in the problem text, would be harder to identify. When the text says explicitly " Cast and crew individuals" it is easier to classify them to the same hierarchy. However, the text does not specify explicitly that there are two kinds of awards, and the reader has to figure it out. Indeed, many of them missed these hints in the text and avoided the 'Award' hierarchy.

As the model elements are more abstract, and distant from the description in the text, novice designers find it difficult to identify and use them in their solution. Aggregations, hierarchies, and

weak entities are not specified in the text at all. They are all abstract concepts used to model complex situations.

To be able to identify and implement correctly complex concepts one has to demonstrate abstract thinking abilities and in terms of the SOLO model [18] one has to be in the relational level of understanding.

As shown in Figure 1, the 'Award' entity is modeled as a weak one, as it depends completely on the organization that provides it. Although most of the students specified the 'Award' entity in their solution, most of them (88%) did not mark it as a weak entity. While most of the students marked 'award name' and 'year' as key attributes of 'Award' entity, Ben recognized the fact that these attributes cannot identify awards entities by themselves, and hence added an artificial key. Ben did not think of a weak entity as a model element appropriate for the modelling of the 'Award' concept, due to the artificial key solution he provided. Weak entities are elusive concepts. Ben preferred to add an artificial key and leave the 'Award' entity strong. Weak entities add complexity that complicates things for novice designers. The other students did not notice that 'award name' and 'year' are not unique and hence cannot serve as a key.

Almost all of the students (92%) modelled the cast-and-crew hierarchy as expected. However, only 9 students (14%) modelled the award hierarchy. Perhaps the correct model of the first stems from the similarity it has to a 'person-customer-employee' they have seen in class. However, the solution (see Fig. 6) provided by most of the students to the 'Award' concept is faulty since it requires each award to be given to both movie and industry-person. Moreover, one can provide by mistake a personal award to a movie, or movie-award to a person. A better model has two separate awards (extending an abstract award containing the common fields and relationships) each for every purpose. It is also worth noting that the award hierarchy is less prominent than the cast-and-crew hierarchy, as the lower-level entities do not have attributes of their own, and hence it was not a surprise to us that fewer students addressed correctly that hierarchy. However, we did not expect such a huge rift. These results are in line with the findings of [20], in which students made a lot of modelling errors regarding hierarchies of classes.

Most of the students (83%) failed to provide an aggregation around the participation relationship to connect it to the 'Person Award'. Most of them provided some variation of the solution depicted in Fig. 6. This solution is erroneous since the model can only tell which movies (one relationship) and which persons (one relationship) received awards, but it cannot tell for a person who receives an award, the movie in which the person participated for which the award was given to. Surprisingly though, no one suggested an n-ary solution in which 'Movie', 'Industry Person', 'Role', and 'Award' are connected via 4-ary relationship, although such a solution makes sense (though duplicates the participation relationship).

Generally, we can conclude that the students had even more difficulties to identify and model the elements included in category C than the elements of the two previous categories. Most of the students did not identify correctly weak entities, hierarchies, and aggregations. Indeed, according to our interpretation of the SOLO's taxonomy, these complex aspects of the model were not used properly by many of the students, as they require high abstraction skills to create a complete and coherent view of the model. Novice designers are usually not able yet to use advanced model components and make the simplistic assumption to disregard them. We may say that abstract thinking at that level necessitates a relational level of understanding [12] that only a small percentage of novice designers possess.

5. INSTRUCTIONAL IMPLICATIONS

Mastery of the ERD design model enables one to design a robust, clear, and normalized database schema. Unfortunately, many students in our study demonstrated only basic abilities concerning the designing of the ERD model and failed to apply more advanced concepts of the model such as non-binary relationships, entity-hierarchies, and aggregations. To improve the students' abilities, we recommend more practice of these advanced concepts. The teacher should present more cases like the problem discussed in this paper and discuss it in detail while presenting various solutions to each concept presented in the problem and show its merits and flaws. The teacher should also point out erroneous solutions, and discuss the problems concerned with the faulty solutions.

Also, instructors should include activities in which students are requested to evaluate the design solution of other classmates that will be followed by a class discussion in which the students' insights will be emphasized. The evaluation process of other classmate solution may cause students to reflect on their solution and improve it and help them develop reflection skills. To complete the whole process the students will be also required to build the derived relational schema and address several queries that will reflect the consequences of their faulty ERD design.

6. CONCLUDING REMARKS

In this research, we focused on the difficulties related to the designing of a data model using ERD. The results which have been obtained reveal that most students have difficulties in designing an appropriate data model using ERD. Although the students had been taught and had used data model design via ERD, they did not properly use the advanced concepts of the model and most of them remained at a basic level of understanding. Most of the students correctly identified most of the entities and attributes stemming from the problem text, as well as the binary relationships among them. However, the students had difficulties in exhibiting high levels of understanding concerning the more complex constituents such as non-binary relationships, aggregations, hierarchies, and week entities. These elements were not used properly. Some students skipped them or bypassed them partially or completely, while others used them in a faulty way. These difficulties might be attributed to their basic level of abstraction abilities [16]. The results are also in line with previous research regarding difficulties in understanding object-oriented design capabilities of novice programmers [28, 29]. The SOLO model [18] provides a theoretical infrastructure to map the students' difficulties and the underlying reasons.

We believe that providing students with more practice including integrated examples that require the design of complex data models, with reflexive and n-ary relationships, hierarchies, and weak entities might help them to develop their abstraction abilities in general, and data modeling capabilities in particular.

7. LIMITATIONS AND FUTURE PLANS

We intend to repeat the above study with a larger student group and add two additional phases. Following the ERD design, the students will be asked to add the derived relational schema and then address several queries and examine the effects of these phases on their ERD design.

REFERENCES

[1] Moody, D. L. & Shanks, G. G. (1998). "What Makes a Good Data Model? A Framework for Evaluating and Improving the Quality of Entity Relationship Models," Australian Computer Journal, vol. 30, pp. 97-110.

[2] Codd, E. F. (1970). A relational model of data for large shared data banks. Communications of the ACM, 13(6), 377-387.
[3] Codd, E. F. (1979). Extending the database relational model to capture more meaning. ACM Transactions on Database Systems (TODS), 4(4), 397-434.
[4] Frederiks, P. J., & Van der Weide, T. P. (2006). Information modeling: The process and the required competencies of its participants. Data & Knowledge Engineering, 58(1), 4-20.
[5] Chen, P. P. S. (1976). The entity-relationship model—toward a unified view of data. ACM Transactions on Database Systems (TODS), 1(1), 9-36.
[6] Teorey, T.J., Yang, D., and Fry, J.F. (1986). A logical design methodology for relational databases using the extended entity-relationship model. Computing Surveys, Vol. 18, No. 2, pp. 197-222.
[7] Ram, S. (1995). "Deriving Functional Dependencies from the Entity Relationship Model," Communications of the ACM. Vol. 38, No. 9, pp. 95-107.
[8] Batra, D. and Antony, S (1994). Novice errors in database design. European Journal of Information Systems, Vol. 3, No. 1, pp. 57-69.
[9] Antony, S. R., & Batra, D. (2002). CODASYS: a consulting tool for novice database designers. ACM Sigmis Database, 33(3), 54-68.
[10] Batra, D. (2007). Cognitive complexity in data modeling: Causes and recommendations. Requirements Engineering 12(4), 231–244.
[11] Anderson, L. W., Krathwohl, D. R., & Bloom, B. S. (2001). A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives. Allyn & Bacon.
[12] Biggs, J. B., & Collis, K. F. (1982). Evaluation the quality of learning: the SOLO taxonomy (structure of the observed learning outcome). Academic Press.
[13] Bloom, B. S. (1956). Taxonomy of educational objectives. Vol. 1: Cognitive domain. New York: McKay, 20, 24.
[14] Rashkovits, R. & Lavy, I. (2020). Students difficulties in identifying the use of ternary relationships in data modeling. The International Journal of Information and Communication Technology Education (IJICTE), Vol. 16, Issue 2, 47-58.
[15] Balaban, M., & Shoval, P. (1999, November). Resolving the “weak status” of weak entity types in entity-relationship schemas. In International Conference on Conceptual Modeling (pp. 369-383). Springer Berlin Heidelberg.
[16] Or-Bach, R., & Lavy, I. (2004). Cognitive activities of abstraction in object-orientation: An empirical study. The SIGCSE Bulletin, 36(2), 82-85.
[17] Liberman, N., Beeri, C., Ben-David Kolikant, Y., 2011). Difficulties in Learning Inheritance and Polymorphism. ACM Transactions on Computing Education, 11, (1), Article 4, 1-23.
[18] Chick, H. (1998). Cognition in the formal modes: Research mathematics and the SOLO taxonomy. Mathematics Education Research Journal, 10(2), 4-26.
[19] Huang, I. L. (2012). An empirical analysis of students' difficulties on learning conceptual data modeling. Journal of Management Information and Decision Sciences, 15(2), 73.
[20] Leung, F., & Bolloju, N. (2005). Analyzing the quality of domain models developed by novice systems analysts. In System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on (pp. 188b-188b). IEEE.
[21] Kesh, S. (1995). Evaluating the quality of entity relationship models. Information and Software Technology, 37(12), 681-689.
[22] Topi, H., Valacich, J. S., Wright, R. T., Kaiser, K., Nunamaker Jr, J. F., Sipior, J. C., & de Vreede, G. J. (2010). IS (2010): Curriculum guidelines for undergraduate degree programs in information systems. Communications of the Association for Information Systems, 26(1), 18.
[23] Lindland, O. I., Sindre G., and Solvberg A., (1994). "Understanding quality in conceptual modeling," IEEE Software, vol. 11, pp. 42-49.
[24] Taylor, S.J. & Bogdan, R. (1998). Introduction to Qualitative Research Methods. New York: John Wiley & Sons.
[25] Neuendorf, K. A.(2002). The Content Analysis Guidebook. Thousand Oaks, CA: Sage Publications.
[26] Batra, D., & Davis, J. G. (1992). Conceptual data modelling in database design: similarities and differences between expert and novice designers. International journal of man-machine studies, 37(1), 83-101.

[27] Sim, E. R., & Wright G. (2001). The difficulties of learning object-oriented analysis and design: An exploratory study. Journal of Computer Information Systems, 42(4), 95–100.
[28] Lavy, I., Rashkovits, R., & Kouris, R. (2009). Coping with abstraction in object orientation with special focus on interface class. The Journal of Computer Science Education, 19(3), 155-177.
[29] Rashkovits, R., & Lavy, I. (2011). Students' strategies for exception handling. Journal of Information Technology Education, 10, 183-207.

AUTHORS

Dr Rami Rashkovits is a senior lecturer at Peres Academic Center, head of the Department of Management Information Systems. His PhD dissertation (in the Technion – Israel Institute of Technology) focused on content management in wide-area networks using profiles concerning users' expectations. His research interests are in the fields of distributed content management as well as computer sciences education. He has published over thirty papers and research reports.

Professor Ilana Lavy is an associate professor with tenure at the Academic College of Yezreel Valley. Her PhD dissertation (in the Technion – Israel Institute of Technology) focused on the understanding of basic concepts in elementary number theory. After finishing a doctorate, she was a post-doctoral research fellow at the Education faculty of Haifa University. Her research interests are in the field of pre-service and mathematics teachers' professional development as well as the acquisition and understanding of mathematical and computer science concepts. She has published over a hundred papers and research reports (part of them is in Hebrew).