Softw Syst Model
DOI 10.1007/s10270-012-0312-6

SPECIAL SECTION PAPER

A fine-grained analysis of the support provided by UML class diagrams and ER diagrams during data model maintenance

Gabriele Bavota · Carmine Gravino · Rocco Oliveto · Andrea De Lucia · Genoveffa Tortora · Marcela Genero · José A. Cruz-Lemus

Received: 30 March 2012 / Revised: 30 October 2012 / Accepted: 19 December 2012

Abstract This paper presents the results of an empirical study aiming at comparing the support provided by ER and UML class diagrams during maintenance of data models. We performed one controlled experiment and two replications that focused on comprehension activities (the first activity in the maintenance process) and another controlled experiment on modification activities related to the implementation of given change requests. The results achieved were analyzed at a fine-grained level aiming at comparing the support given by each single building block of the two notations. Such an analysis is used to identify weaknesses (i.e., building blocks not easy to comprehend) in a notation and/or can justify the need of preferring ER or UML for data modeling. The analysis revealed that the UML class diagrams generally provided a better support for both comprehension and modification activities performed on data models as compared to ER diagrams. Nevertheless, the former has some weaknesses related to three building blocks, i.e., multi-value attribute, composite attribute, and weak entity. These findings suggest that an extension of UML class diagrams should be considered to overcome these weaknesses and improve the support provided by UML class diagrams during maintenance of data models.

Communicated by Prof. Tony Clark and Prof. Jon Whittle.

This paper is an extension of the work "Identifying the Weaknesses of UML Class Diagrams during Data Model Comprehension" appeared in the *Proceedings of the 14th International Conference on Model Driven Engineering Languages and Systems*, Wellington, New Zealand, pages 168–182, 2011. LNCS Press.

G. Bavota · C. Gravino · A. De Lucia · G. Tortora
University of Salerno, Fisciano, Salerno, Italy

R. Oliveto (✉)
University of Molise, Pesche, Isernia, Italy
e-mail: rocco.oliveto@unimol.it

M. Genero · J. A. Cruz-Lemus
University of Castilla, La Mancha, Spain

1 Introduction

Data modeling is a process used to define and analyze data requirements needed to support the business processes within the scope of corresponding information systems in organizations [19]. This process usually involves data modelers that work closely with business stakeholders and potential users of the information system.

The output of the data-modeling process is represented by a data model. The data model describes both the structure of and the operations on a database in a diagrammatic form aiming at facilitating the communication between the stakeholders involved in the data-modeling process [19].

Understanding and interpreting data models represents a fundamental activity from the earliest stages of software development, e.g., requirement analysis. Thus, a comprehensive notation is really desirable to avoid misunderstanding that can lead to the introduction of errors very expensive to remove in the later phases of the software development.

A comprehensive notation is also desirable to facilitate the maintenance of the produced data model. Indeed, data models should be considered living documents that will change in response to a changing business. Thus, a comprehensive notation can facilitate the comprehension activities that have to be performed to understand the data model before the analysis and the implementation of a change request.

Entity-relationship (ER) and its extensions are the most used notations for database conceptual modeling and still remains the *de facto* standard [19]. The success of the object-oriented (OO) approach for software development has

Published online: 17 January 2013

encouraged the use of this approach also for database modeling [23]. Specifically, UML class diagrams can be used to represent the conceptual schema of the whole software system, so the same notation can be used to model the functionality of the system as well as to represent its data. The structural constructs of the UML class diagram which represents the data structure is somewhat equivalent to extended ER (EER) representation (e.g., object classes considered equivalent to entity and relationship types). The functionality is represented through "Methods" that are attached to the object classes.

While UML is becoming a *de facto* standard for the analysis and design of software systems, it is not exploited with the same success for modeling databases. Indeed, nowadays ER remains the most used notation to model databases, and in some cases, it complements UML in the design of software systems. A recent survey also indicated that in some cases, both ER and UML class diagrams are employed to represent the same database [11]. Such behaviors might be the trigger for possible problems during the evolution of the data models. More effort is required to maintain the models and their implementation up-to-date, since out-of-data models can generate inconsistency and misunderstanding during software maintenance and evolution.

All these considerations lead researchers to empirically compare the ER and UML diagrams to show the actual benefits given by one notation as compared to the other [11]. The results achieved in all these studies indicate that the support given by UML class diagrams in comprehension and modification tasks is at least equal to (and in some cases, higher than) the support given by ER diagrams. However, a deeper analysis concerning the identification of the graphical elements of one notation that are more comprehensible than the corresponding elements in the other notation is still missing. To the best of our knowledge, such an analysis was only performed by Shoval et al. [23] during the comparison of EER and OO models.

A fine-grained analysis is vital to provide insight on why UML class diagrams are better than the ER diagrams or vice versa and to highlight strengths and limitations of the two notations. This kind of analysis can be used to (i) justify the need of preferring ER or UML class diagrams for data modeling, or (ii) identify weaknesses in a notation that could be overcome to improve its support for data modeling.

In this paper, we aim at bridging this gap presenting the results of a set of controlled experiments to deeply analyze the support given by ER and UML class diagrams during maintenance of data models. In particular, we conducted a controlled experiment and two replications aiming at analyzing the support given by the two notations during comprehension tasks. In addition, we conducted another controlled experiment to analyze the support given during modification activities. We focus the attention only on these two activities, since high comprehensibility and high modifiability are the two typical expectations of people using these two notations to represent conceptual data models. Indeed, an effective notation should help to understand a system, modify it, and avoid defects in the early stages of development [7].

The experiments aimed at performing a fine-grained analysis to (objectively and subjectively) compare the single building blocks, i.e., Entity, Primary Key/ID, Composite Attribute, Multi-value Attribute, Recursive relationship, Relationship cardinality, Ternary relationship, Generalization IS-A, Weak entity, M:N relationship, of the two notations. The experiments aimed at analyzing the comprehensibility of the two notations involved 156 students of the University of Salerno (Italy) with different academic background represented by fresher, bachelor, and master's students. The experiment conducted to analyze the modifiability of the two notations involved 28 master's students of the University of Salerno (Italy).

The results achieved indicated that UML class diagrams are characterized by three weaknesses related to the representation of Composite attribute, Multi-value attribute, and Weak entity, as compared to the ER diagrams. However, except for the three identified weaknesses, the UML class diagrams are generally more comprehensible and easier to modify than the ER diagrams.

The rest of the paper is organized as follows. Section 2 presents the related works. Section 3 provides details on the design of the experiments, while Sect. 4 presents the results achieved. Section 5 discusses the possible threats that could affect the validity of the results achieved in our study. Concluding remarks and directions for future work are given in Sect. 6.

2 Related work

In the last four decades, several papers have analyzed graphical notations supporting data modeling through controlled experiments, empirical studies, and surveys.

In the 1970s, there was a tendency for comparing logical models and focusing on the relational model versus hierarchical and network models. For example, Brosey and Shneiderman [8] found that the hierarchical model was significantly easier to use than the relational model, but only for the beginners group. Durding et al. [13] investigated how people could organize data without using specific data models. Results suggested that the ease of using a model depends on the inherent structure of the data in an application, and the results supported Brosey and Shneiderman's findings.

During the 1980s, empirical studies were performed to compare logical models with conceptual models. They largely emphasized the relational model versus conceptual models. Usually, the results favor one model or the other

based on the design task. Juhn and Naumann [14] compared logical data structure (LDS), entity-relationship model (ERM), data access diagram (DAD), and relational model (RM). They reported that in relationship and cardinality finding tasks, ERM and LDS were superior to RM and DAD. On the other hand, RM outperformed ERM and LDS on identifier comprehension tasks. Batra et al. [4] compared novice user's performances, using RM and Extended ER model (EERM), and the results of their study suggested that EERM led to significantly better user's performances in modeling binary and ternary relationships. Palvia et al. [21] reported end-user's experiences with hierarchical, network, relational, and object-oriented models (OOM) [12]. Their analysis revealed that the OOM and network model outperformed relational and hierarchical model in terms of comprehension, efficiency, and productivity. Liao and Shih [17] investigated the effects of data models and training on data representation. EERM resulted to be superior to RM in many areas. Furthermore, the high-degree training group outperformed the low-degree one in modeling identifiers, categories, and relationships.

In the 1990s, the research interest moved to empirically comparing different conceptual models, in particular, ERM and OOM. As the focus of the current paper falls in this category of studies, we will present a more detailed analysis of previous research on comparing ERM and OOM. Tables 1 and 2 summarize the analyzed papers. For a better extract of the main information when comparing the empirical studies, we designed Tables 1 and 2, according to the following columns:

1. *Ref.* contains the reference to the paper presenting the considered empirical study.
2. *Goal* describes the goal pursued by the empirical study.
3. *Subjects* presents the numbers of subjects, who participated in the empirical studies.
4. *Independent variables* describes the variables that are studied to ascertain their effect on the dependent variables. The values (treatments) of the independent variables are presented.
5. *Dependent variables* presents the outcome variables, which are the variables that are affected by the changes produced in the independent variables.
6. *Experiment design* contains the type of design selected, which can be:
   - *Between-subjects* each subject receives only one treatment.
   - *Within subjects* each subject receives all the treatments.
7. *Tasks* describes the tasks to be performed by the subjects as part of the empirical study.
8. *Results* reveals the main findings obtained.

Table 1 presents the studies that compare the single building blocks (also called constructs), e.g., Entity, Primary Key/ID, Composite Attribute, Multi-value Attribute, while Table 2 summarizes empirical studies that compare ERM and OOM in overall terms, i.e., not focusing on each building blocks.

There are four empirical studies for comparing data models by carrying out a fine-grained analysis of each of the building blocks:

- Bock and Ryan [6] examined the correctness of the design for eight constructs (i.e., entities/objects, attribute/property identifiers, categories, unary one-to-one relationships, binary one-to-many relationships, binary many-to-many relationships, ternary one-to-many-to-many relationships, and ternary many-to-many-to-many relationships) in an empirical analysis comparing EER and OO models from a designer perspective. The analysis revealed significant differences only in four cases (i.e., representation of attribute identifiers, unary 1:1 and binary M:N relationships) and no difference was found concerning the time to complete the tasks.
- Shoval and Shiran [24] compared EER and OO data models from the point of view of design quality, where quality was measured in terms of correctness for the produced models, time to fully perform the design task, and designers' opinions. The comparison performed by Shoval and Shiran based on several buildings blocks, related to relationships (i.e., inheritance, unary 1:1, binary 1:1, binary 1:n, and binary M:N relationships, ternary m:n:1, and ternary m:n:p relationships) revealed that there was no significant difference between EER and OO data models, except for the use of ternary and unary relationships, since in this case, EER models provided better results. Furthermore, the designers preferred to work with the EER models.
- Shoval and Frumermann [23] also performed a comparison of EER and OO diagrams, focusing on three main building blocks (i.e., attribute, binary relationships, and ternary relationships), taking into account the user comprehension. As done by Shoval and Shiran [24], they separately examined the comprehension of various constructs of the analyzed models. Similar to Shoval and Shiran [24], their analysis revealed that the EER schemas are more comprehensible for ternary relationships while for the other constructs, no significant difference was found.
- Liao and Plavia [16] assessed the design effectiveness of some data models (i.e., RM, ERM, and OOM) from the end-user's perspective. They focused on the following buildings blocks: entities/objects, descriptors, identifiers, relationships and generalization hierarchies, and six facets of a relationship: unary one-to-one, unary one-ternary, binary one-to-one, binary one-to-many, binary

Table 1 Empirical studies focusing on the building blocks

References	Type of empirical study	Goal	Subjects (students)	Independent variables (treatments)	Dependent variables	Experiment design	Tasks	Results
[6]	1 experiment	Comparing data models in terms of the dependent variables, focusing on 8 building blocks	38	ERM (extended model)-OOM ([12])	Model correctness: errors found in the models based on the protocol proposed by [4]. Time	Between subjects (2 groups of 19 students)	Design a data model	ERM provides significantly improved correctness for attribute/property identified, unary one-to-one relationship and binary many-to-many relationship. There is no difference in time to complete the tasks
[24]	1 experiment	Comparing two data models in terms of the dependent variables focusing on several building blocks related to relationships	44	ERM (extended model)-OOM ([12])	Correctness of design Designer preference time	Between subjects (2 groups of 22 subjects)	Design a data model Fill a perception-based questionnaire	ERM surpasses OOM in the correctness of designing unary and ternary relationships. It takes less time to complete the task with ERM. Designers prefer working with the ERM
[23]	1 controlled experiment	Comparing the comprehension of two data models focusing on three building blocks	78	ERM (extended model)-OOM	Comprehension correctness	Between subjects (2 groups of 41 and 37 subjects)	Complete a comprehension questionnaire	ERM is more comprehensible for ternary relationships
[16]	1 controlled experiment	Investigating the design effectiveness of the data models from end users' perspective focusing on several building blocks	66	RM (relational model)-ERM (extended model)-OOM ([12])	Modeling correctness (number of correct modeling tasks) Efficiency (time required to complete the task satisfactorily) Perceived ease of use (subjects' perception)	Between subjects (3 groups of 23 subjects)	Design a data model Complete a perception-based questionnaire	ERM is generally superior in representing relationships. OOM requires less time. There is no significant difference in perceived ease of use

Table 2 Empirical studies focusing on the whole models

References	Type of empirical study	Goal	Subjects (students)	Independent variables (treatments)	Dependent variables	Experiment design	Tasks	Results
[22]	1 controlled experiment	Examining end-user comprehension, efficiency and productivity in using three conceptual data models	121	DSD-ERM-OOM	Comprehension correctness (number of correct answers) Efficiency (time spent in answering the questionnaire) Productivity (number of correct answer/time)	Between subjects (3 groups of 41 subjects)	Complete a comprehension questionnaire	Comprehension, efficiency and productivity: OOM is superior to DSD or ERM
	1 survey	Comparing each data model in terms of the dependent variables. Examining the relationships between the dependent variables and user attributes (work experience, computer experience, database experience)	36	DSD-ERM-OOM	A set of 17 characteristics measured via perception-based measures using a 1–5 Likert scale	Within subjects	Fill a perception-based questionnaire	There is no significant difference between the three models. The superior user's performances for OOM detected in the experiment diminished with increased computer and database experience
[11]	1 experiment and 2 replications	Comparing ERM and UML class diagrams in terms of comprehension support	40 (exp) 30 (replic. 1) 68 (replic. 2)	ERM (extended model)-OOM (UML class diagram)	Comprehension support (F-measure)	Within subjects	Complete a multiple-choice comprehension questionnaire	UML class diagrams provide better comprehension support
	1 experiment and 1 replication	Assessing whether UML class diagrams provide a better support than ER diagrams in the comprehension of the change to perform on the data to meet a change request	40 (exp) 30 (replic. 1)	ERM (extended model)-OOM (UML class diagram)	Maintenance support (F-measure)	Within subjects	Complete a multiple-choice questionnaire related to comprehension of the change to perform on the data model to meet a change request	Both notations give the same support
	1 experiment and 1 replication	Assessing whether UML class diagrams provide a better support than ER diagrams in the detections of defects in a data model	40 (exp) 30 (replic. 1)	ERM (extended model)-OOM (UML class diagram)	Verification support (F-measure)	Within subjects	Detect defects in conceptual data models	UML class diagrams provide a better verification support

many-to-many, and ternary many-to-many-to-many. The results of the study revealed that the ERM was generally superior in representing relationships, OOM required less time, and there was no significant difference with respect to perceived ease of use.

The remaining two studies (Table 2), which contain eight experiments in total (considering replications) and one survey, compared data models in overall terms, considering the model as a whole, not analyzing each building block:

- A comparison between OOM and ERM from an end-user's perspective was carried out by Palvia et al. [22], whose aim was to establish which was more comprehensible. The results of a controlled experiment suggested that the OOM was superior in this respect. They also carried out a survey pursuing to compare ERM and OOM with respect to 17 characteristics (i.e., data structure, data independence, data integrity, data duplication, level of detail, communication ability to users, communication ability to computer professionals, documentation, revision of design, better design production, early discovery of problem, flexibility of design produced, ease of use, ease of learning, efficiency, productivity, and overall quality). They also aimed at investigating the relationships between the 17 characteristics and user's attributes (i.e., work experience, computer experience, and database experience). The obtained findings revealed that the gap in the user's performances using OOM and EER highlighted by the experiment diminished when computer and database experience of subjects increased.
- De Lucia et al. [11] presented the largest empirical study, which consist of three sets of controlled experiments (7 experiments in total) aimed at analyzing whether UML class diagrams provided better support in comprehension, maintenance, and verification tasks, respectively. The results showed that the UML class diagrams provided better support in comprehension and verification activities. There was no difference regarding maintenance tasks.

3 Design of the experiments

We present in detail the design of the experiments we carried out to assess and compare the support provided by ER and UML class diagrams during maintenance of data models. We performed one controlled experiment (and two replications) that focused on comprehension activities. Subjects represented the only substantial difference among the experiment and the two replications. We also conducted another controlled experiment considering modification activities. In the context of this experiment, we considered different tasks and different subjects as compared to the experiments on comprehension activities.

According to the two-dimensional classification scheme by Basili et al. [3], we performed blocked subject-project studies, as we examined objects across a set of subjects and a set of projects. The description follows a template originating from the Goal–Question–Metric (GQM) paradigm [2] as described by Wohlin et al. [26]. The materials and the the raw data of our studies are publicly available online [5].

3.1 Experiment definition and context

The goal of our study was "analyzing *the effectiveness of UML class diagrams and ER diagrams* for the purpose of *understanding which provides better support* with respect to *the comprehension and modification of data models from* the point of view *of researchers, in* a context *represented by B.Sc. and M.Sc. students*".

The performed experiments involved students from the University of Salerno (Italy) having different academic backgrounds and, consequently, different levels of experience on ER and UML class diagrams. The experiments on comprehension activities were conducted in the 2009 with three categories of students:

- *fresher students* first-year B.Sc. students those were starting their academic career when the experiment was performed;
- *bachelor students* second-year B.Sc. students those attended Programming and Databases courses in the past and were attending the Software Engineering course when the experiment was performed;
- *master's students* first-year M.Sc. students those attended advanced courses of Programming and Software Engineering in the past and were attending an advanced Databases course when the experiment was performed.

It is worth to note that in the Software Engineering course, the design notation used is UML while in the Databases course, the design notation is ER.

The number of subjects involved in the original experiment (Com_1 in the following) were 37 bachelor students, while the first (Com_2 in the following) and second replications (Com_3 in the following) involved 52 master's students and 67 fresher students, respectively.

The experiment on modification activities (Mod in the following) was conducted at the University of Salerno (Italy) involving 28 master's students having almost the same academic background and the level of experience on ER diagrams and UML class diagrams of the students involved in Com_2.

For all the experiments, we employed the data models of the following systems:

Table 3 Data models used in each controlled experiment

System	# Entities	# Attributes	# Relationships
Company	7	17	5
EasyClinic	6	18	5

- *Company* a software system implementing all the operations required to manage the projects conducted by a company;
- *EasyClinic* a software system implementing all the operations required to manage a medical doctor's office.

We used two different data models represented in terms of ER and UML class diagrams.[1] Table 3 shows the characteristics of the data models used in the experiments. The selection of the objects for each experiment was performed ensuring that the data models had a comparable level of complexity. For this reason, we extracted sub-diagrams of comparable size from the original data models according to "the rule of seven" given by Miller [18] to build comprehensible graphical diagrams.[2] In the context of our experimentation, we applied such a rule to select data models easy to comprehend. This was necessary because (i) each experiment was designed to be performed in a limited amount of time and (ii) a simple data model is preferred to a more complex data model, since the latter might negatively influence the subject's performances. To obey to such a rule, for EasyClinic, we extracted from the original data model only the entities and relationships related to the booking management.

3.2 Variable selection and experiment design

We employed a single factor within-subjects design, where the independent variable (main factor) is represented by the design notation used to represent a data model. This variable is denoted as "Method", that can assume two values, ER diagram (ER) or UML class diagram (CD).

For the experiments Com_1, Com_2, and Com_3, the dependent variable is "Comprehension Level", which denotes the comprehension level achieved by the subjects using the two methods (i.e., ER and CD). Turning to the experiment Mod, the dependent variable is "Modification Level", related to the ability of the subjects to comprehend which modification should be applied to a data model to implement a change request.

To evaluate the "Comprehension Level" and "Modification Level" values achieved with the two methods, we asked the subjects to answer a questionnaire. Specifically, for Com_1, Com_2, and Com_3, we employed the same questionnaire consisting of ten multiple-choice questions where each question has three possible answers and, among them, one or more are correct. The questionnaire used for Mod also included ten multiple-choice questions, however each question admitted only one correct answer.

The questions cover all the building blocks B_i of the two notations exploited to model a database, where $B_i \in$ {Entity, Primary Key/ID, Composite Attribute, Multi-value Attribute, Recursive relationship, Relationship cardinality, Ternary relationship, Generalization IS-A, Weak entity, M:N relationship}. Figure 1 shows a sample comprehension task on the system Company, while Fig. 2 reports an example on modification activity on the system EasyClinic.

The structure of the questionnaires allowed us to assess the answers using the well-known information-retrieval (IR) metrics, namely recall and precision [1]. Indeed, since the questionnaire is composed of multiple-choice questions, we could compute recall and precision for each question as given below.

Q4 Let us focus on the classes Project and Company. Which of the following statements is true:

[] A company has a unique office
[] A project has a unique office
[] A company may have multiple offices

Fig. 1 An example of comprehension activity

Q6 We want to specify the information about the equipments present in each laboratory. A laboratory can contain several equipments and each equipment is identified by an inventory code. A description and a name have also to be specified for an equipment. Indicate the part of the class diagram interested by this modification activity:

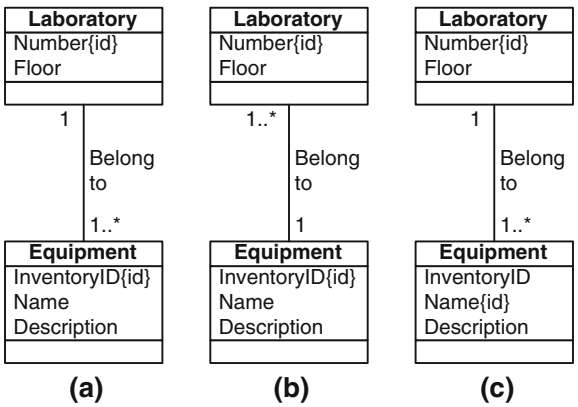

Fig. 2 An example of modification activity

[1] A representation of the EasyClinic system using both ER and UML is reported in the appendix.

[2] The rule of seven is the generally accepted claim that people can hold approximately seven (plus or minus two) chunks or units of information in their short-term memory at a time [18].

$$\text{recall}_{s,i} = \frac{|\text{answer}_{s,i} \cap \text{correct}_i|}{|\text{correct}_i|} \%$$

$$\text{precision}_{s,i} = \frac{|\text{answer}_{s,i} \cap \text{correct}_i|}{|\text{answer}_{s,i}|} \%$$

where $\text{answer}_{s,i}$ is the set of answers given by subject s to question i and correct_i is the set of correct answers expected for question i.

It is worth noting that the recall and precision measure two different concepts. Thus, we decided to use an aggregate measure (i.e., F-measure [1]) to obtain a balance between them:

$$F\text{-measure}_{s,i} = 2 \times \frac{\text{precision}_{s,i} \times \text{recall}_{s,i}}{\text{precision}_{s,i} + \text{recall}_{s,i}} \%$$

Thus, in the context of our study, we computed both the variables "Comprehension Level" and the "Modification Level" using the F-measure.

Besides an objective evaluation of the support given by the different building blocks, we performed a subjective evaluation aiming at capturing the subjects' preferences between the two considered notations. Specifically, subjects filled-in another questionnaire where for each building block B_i, they expressed a preference among ER diagram, No preference, and UML class diagram.

The experiments were performed in a controlled laboratory and organized in two sessions where subjects used the two different methods, i.e., ER and CD, to fill-in two questionnaires on two different data models. Such an organization required to control other factors (called co-factors) that may impact the results achieved by the subjects and be confounded with the effect of the main factor. In the context of our study, we identified the following co-factors:

- *System* in the context of the experiments students had to perform comprehension or modification activities on two different systems (Sect. 3.1). Even if we tried to select software systems of a comparable size and to balance the complexity of the data models by using the Miller's rule, there is still the risk that the system complexity may have a confounding effect with "Method". For this reason, we considered the modeled system as an experimental co-factor.
- *Lab* the experiments were organized in two laboratory sessions and in each session, the subjects experimented one of the two methods (i.e., CD and ER). This requires to analyze whether subjects perform differently across subsequent sessions to verify the presence of any learning or tiring effect.

The organization of each group of subjects[3] in each experimental lab session (*Lab1* and *Lab2*) followed the design shown in Table 4. In particular, the rows represent the four experimental groups, whereas the columns refer to the design notation used to represent the data model (i.e., ER and CD). Such an experimental design ensured that each subject worked on different systems in the two laboratory sessions, using a different design method each time. The chosen design also permitted to consider different combinations of "System" and "Method" in different order across laboratory sessions. It is important to note that all the experiments (i.e., Com_1, Com_2, Com_3, and Mod) followed the same balanced design (Table 4).

Table 4 Experimental design

Group	Method	
	ER	CD
A	EasyClinic, Lab1	Company, Lab2
B	Company, Lab2	EasyClinic, Lab1
C	Company, Lab1	EasyClinic, Lab2
D	EasyClinic, Lab2	Company, Lab1

3.3 Experimental procedure and data analysis

Subjects performed the assigned tasks individually. Before the experiments, subjects were trained on both ER and UML class diagrams. To avoid bias, (i) the training was performed on a data model not related to the systems selected for the experimentation and (ii) its duration was exactly the same for all the experiments. Right before the experiments, the students attended a 30-min presentation where detailed instructions concerning the tasks to be performed were illustrated.

According to the experimental design (Table 4), each subject was involved in two laboratory sessions, where subjects had a fixed amount of time to complete the required tasks. In particular, in the experiments Com_1, Com_2, and Com_3, subjects had 30 min to complete the task on the data models of the assigned systems documented with ER and CD. In the experiment Mod, subjects had 45 min to complete the assigned task.

Moreover, at the end of each laboratory session, a survey questionnaire was proposed to the subjects. This survey aimed at assessing the overall quality of the provided material as well as the clearness and difficulty of comprehension and modification tasks. In particular, the subjects provided answers to the following questions (one choice for each question).

- S_1: I had enough time to perform the tasks
- S_2: The task objectives were perfectly clear to me
- S_3: The tasks I performed was perfectly clear to me
- S_4: Judging the difficulty of the assigned task

[3] The students were assigned to the four groups in a randomly balanced way.

where S1, S2, and S3 expected closed answers according to the Likert scale [20] going from 1 (strongly disagree) to 5 (strongly agree), while for S4 the answer ranges from 1 (very low) to 5 (very high).

After the execution of each experiment, we collected the questionnaires filled-in by each subject in each laboratory session. The results of the questionnaire were reported by one of the authors in spreadsheets to ease data analysis. To reduce the risk of human errors in reporting the results, another author double-checked the inserted data. Once the data were validated, the F-measures achieved by the subject were calculated.

The results achieved were statistically analyzed. Specifically, we used a paired Wilcoxon one-tailed test [10] to analyse the differences exhibited by each subject for the two methods and test the following null hypotheses:

- H_{0_c}: there is no difference in the comprehension level of subjects using the ER or UML class diagrams.
- H_{0_m}: there is no difference in the modification level of subjects using the ER or UML class diagrams.

A one-tailed paired t test [10] can be used as an alternative to the Wilcoxon test. However, we decided to use the Wilcoxon test, since it is resilient to strong departures from the t test assumptions [7].

When the null hypothesis can be rejected, it is possible to accept an alternative hypothesis highlighting the positive effect of one of the two involved notations on the comprehension/modification level of the subjects. The achieved results were intended as statistically significant at $\alpha = 0.05$. This means that if the derived p value is less than 0.05, the null hypothesis can be rejected and it can be concluded that there is a significant difference between the support given by the treatments when performing comprehension and/or modification tasks.

Other than testing the hypotheses formulated above, it is of practical interest to estimate the magnitude of the difference between performances achieved with different notations (e.g., ER vs. CD). To this aim, we used the Cohen d effect size [10], which indicates the magnitude of the effect of the main treatment on the dependent variables ("whereas the p values reveal whether a finding is statistically significant, effect size indicates practical significance" [15]). For dependent samples (to be used in the context of paired analyses), it is defined as the difference between the means, divided by the standard deviation of the (paired) differences between samples. The effect size is considered small for $0.2 \leq d < 0.5$, medium for $0.5 \leq d < 0.8$ and large for $d \geq 0.8$ [9]. We chose the Cohen d effect size as it is appropriate for our variables (in ratio scale) and given the different levels (small, medium, large) defined for it, it is easy to be interpreted.

The chosen design also permitted to statistically analyze the effects of co-factors and their interaction with the main factor. For this, we used the three-way analysis of variance (ANOVA) [10] to analyze the interaction between the main factor, i.e., "Method", and the two co-factors, i.e., "System" and "Lab". We decided to use ANOVA, because, in contrast to its non-parametric alternatives such as the Friedman test [10] that could have been considered in this case, ANOVA allows to test for the presence of interactions between factors. ANOVA is also quite robust to deviations from normality. In addition, we can relax the normality assumption applying the law of large numbers. According to [25] having a population higher than 30 it is possible to relax the normality assumption. Since in our experimentation subjects performed two tasks, our population is 312 in the experiments on comprehension activities and 56 in the experiment on modification activities.

Finally, we also analyzed the students' preferences about the single building blocks of the two notations using histograms, while the answers provided by subjects to the survey questionnaire were analyzed using boxplots.

4 Analysis of the results

In this section, we report the results achieved in our experiments. We discuss (i) the results of the Com_1, Com_2, and Com_3 experiments, aimed at evaluating the comprehension level of the subjects using the two different notations, i.e., ER and UML class diagrams, and (ii) the results of the Mod experiment, targeted at analyzing the influence of the two notations on the ability of subjects in performing changes on data models.

4.1 Support to comprehension activities

Table 5 reports the descriptive statistics of the comprehension level (measured through F-measure) achieved by the subjects in our experimentation. The results highlighted that the two notations provided comparable support when performing comprehension activities on data models. The higher difference between the two notations in terms of F-measure is 1 % achieved in the experiment with the master's students (see Table 5).

Table 5 Comprehension activities: descriptive statistics

Subjects	ER			CD		
	Mean	Median	SD	Mean	Median	SD
Fresher	0.801	1.000	0.307	0.816	1.000	0.280
Bachelor	0.849	1.000	0.242	0.845	1.000	0.278
Master	0.849	1.000	0.277	0.838	1.000	0.272

Table 6 Comprehension activities: Wilcoxon test

Subjects	CD versus ER			p value	Effect size
	Mean	Median	SD		
Fresher	0.014	0.000	0.404	0.343	0.037
Bachelor	0.003	0.000	0.330	0.420	−0.011
Master	−0.012	0.000	0.383	0.817	−0.030

As designed, we performed the Wilcoxon test to analyze whether the difference between the results obtained using the two notations is statistically significant. Table 6 reports the achieved results that highlight no significant difference between the two notations when used to comprehend data models (p value always higher than 0.05). Thus, we cannot reject the null hypothesis H_{0_c}.

Our finding surprisingly contrasts with the results achieved in a previous experimentation where the analysis highlighted the benefits provided by the UML class diagrams with respect to the ER diagrams during the comprehension of data models [11]. To further investigate this discrepancy, we analyzed the support given by the two notations at a fine-grained level, i.e., on each building block used in the definitions of data models.

To statistically analyze the weaknesses of CD, Table 8 shows the results of the Wilcoxon test executed for each building block to verify where the ER performances are statistically better than those of CD. The achieved results revealed that the ER has a comprehension level significantly higher than the comprehension level of CD for three building blocks, i.e., Composite attribute, Multi-value attribute, and Weak entity. These results generally hold for all the groups of subjects, i.e., Fresher, Bachelor, and Master, involved in the experimentation. The only exception is given by bachelor students when analyzing the Multi-value attribute building block. However, Table 7 shows that bachelor students also achieved better results in terms of descriptive statistics with ER when answering the questions related to the Multi-value attribute.

It is worth noting that the previous controlled experiments [11] did not consider these three building blocks to determine the comprehension level provided by the two notations, i.e., the questionnaires used by the authors did not include questions related to composite attribute, multi-value attribute, and weak entity. To verify whether the different findings between our experiment and previous experiments [11] were due to these three building blocks, we also performed the comparison between ER and UML class diagrams without considering the answers of the students related to Composite attribute, multi-value attribute, and weak entity. Specifically, we re-executed the Wilcoxon test to analyze if CD provided a significant higher comprehension level than ER. The results in Table 9 highlight that the CD achieved statistically significant higher comprehension level than ER for the Fresher and Bachelor students. Thus, in these two experiments we can reject the null hypothesis H_{0_c} in favor of CD. Moreover, CD provided better results than ER also for master's students even if this is not statistically significant (p value 0.096). This results are in line with the results achieved in the previous experiments [11].

Besides an objective analysis, we also conducted a subjective comparison of the support given by the building blocks of the two notations. Figures 3, 4, and 5 report the preferences expressed by the Fresher, Bachelor, and Master, respectively. The analysis of the results confirmed the results of the objective analysis. Indeed, students preferred ER diagrams to represent the three building blocks identified as weaknesses of the UML class diagrams, i.e., Multi-value attribute, Composite attribute, and Weak entity. Concerning the remaining building blocks, the students preferred UML class diagrams to represent the Entity, the Relationship cardinality, and the Generalization relationship, while they did not provide a clear preference for the Primary key/ID, Recursive relationship, Ternary relationship, and M:N relationship.

Summarizing, the achieved (objective and subjective) results highlighted that the UML class diagrams are characterized by three weaknesses related to the representation of composite attribute, multi-value attribute, and weak entity, with respect to the ER diagrams, when performing comprehension activity. However, except for the three identified weaknesses, the UML class diagrams are generally more comprehensible than the ER diagrams, confirming the findings of previous experiments [11].

4.2 Support to modification activities

Table 10 reports the descriptive statistics of the results (in terms of F-measure) achieved by the subjects. As observed for the comprehension level, the overall modification level achieved by the subjects using the two notations is almost the same. Specifically, the difference in terms of F-Measure is less than 2 % in favor of CD (0.750 vs. 0.732).

As expected, this difference does not result to be statistically significant (Table 11). This means that also for the modification level achieved by the subjects the two notations are almost equivalent considering all the building blocks involved in data modeling (i.e., we cannot reject the null hypothesis H_{0_m}). However, the goal of our study was to perform a fine-grained analysis of the two notations, and thus, of the support provided by each of their building blocks.

Table 12 shows the descriptive statistics of the results (in terms of F-measure) achieved by the subjects considering the answers to questions related to each building block. Also in this case for both CD and ER, strengths and limitations are highlighted. In particular, CD performed better than ER on six building blocks: Entity, Primary Key/ID, Relationship

Table 7 Fine-grained analysis of comprehension activities: descriptive statistics

Method	Element	Fresher			Bachelor			Master		
		Mean	Median	SD	Mean	Median	SD	Mean	Median	SD
ER	Entity	0.887	1.000	0.260	0.936	1.000	0.125	0.872	1.000	0.281
	Primary Key/ID	0.784	1.000	0.406	0.955	1.000	0.179	0.907	1.000	0.277
	Composite attribute	0.883	1.000	0.159	0.897	1.000	0.146	0.920	1.000	0.140
	Multi-value attribute	0.859	1.000	0.195	0.847	1.000	0.168	0.862	1.000	0.213
	Recursive relationship	0.779	1.000	0.301	0.757	0.667	0.224	0.817	1.000	0.243
	Relationship cardinality	0.875	1.000	0.240	0.892	1.000	0.158	0.929	1.000	0.179
	Ternary relationship	0.741	1.000	0.347	0.828	1.000	0.220	0.804	1.000	0.321
	Generalization IS-A	0.684	0.667	0.369	0.734	1.000	0.363	0.712	1.000	0.379
	Weak entity	0.725	0.800	0.266	0.767	1.000	0.305	0.747	0.900	0.329
	M:N relationship	0.789	1.000	0.368	0.865	1.000	0.319	0.923	1.000	0.244
CD	Entity	0.961	1.000	0.108	0.937	1.000	0.234	0.926	1.000	0.145
	Primary Key/ID	0.875	1.000	0.296	0.937	1.000	0.234	0.926	1.000	0.246
	Composite attribute	0.742	0.667	0.255	0.781	0.800	0.251	0.815	1.000	0.308
	Multi-value attribute	0.775	0.667	0.259	0.788	0.667	0.257	0.801	0.667	0.209
	Recursive relationship	0.767	1.000	0.323	0.856	1.000	0.226	0.806	0.800	0.210
	Relationship cardinality	0.865	1.000	0.261	0.856	1.000	0.320	0.906	1.000	0.150
	Ternary relationship	0.827	1.000	0.265	0.888	1.000	0.150	0.855	1.000	0.162
	Generalization IS-A	0.828	1.000	0.225	0.838	1.000	0.290	0.804	1.000	0.328
	Weak entity	0.629	0.667	0.407	0.611	0.667	0.407	0.608	0.733	0.447
	M:N relationship	0.890	1.000	0.162	0.955	1.000	0.179	0.929	1.000	0.212

Table 8 Fine-grained analysis of Comprehension activities: Wilcoxon test

Element	Fresher ER versus CD					Bachelor ER versus CD					Master ER versus CD				
	Mean	Median	SD	*p* value	Effect size	Mean	Median	SD	*p* value	Effect size	Mean	Median	SD	*p* value	Effect size
Entity	−0.059	0.000	0.262	0.983	−0.257	−0.036	0.000	0.153	0.599	−0.032	−0.054	0.000	0.309	0.796	−0.161
Primary Key/ID	−0.091	0.000	0.517	0.927	−0.166	−0.027	0.000	0.198	0.415	0.059	−0.019	0.000	0.388	0.660	−0.049
Composite attribute	0.141	0.000	0.303	**0.000**	0.490	0.116	0.000	0.306	**0.022**	0.380	0.105	0.000	0.304	**0.012**	0.343
Multi-value attribute	0.085	0.000	0.316	**0.014**	0.269	0.059	0.000	0.324	0.141	0.180	0.061	0.000	0.311	**0.008**	0.196
Recursive relationship	0.012	0.000	0.401	0.455	0.024	−0.010	0.000	0.287	0.983	−0.345	0.011	0.000	0.308	0.536	0.037
Relationship cardinality	0.009	0.000	0.358	0.439	0.028	−0.009	0.000	0.200	0.446	0.094	0.023	0.000	0.224	0.258	0.103
Ternary relationship	−0.086	0.000	0.471	0.897	−0.184	−0.042	0.000	0.266	0.869	−0.221	−0.050	0.000	0.368	0.720	−0.135
Generalization IS-A	−0.145	0.000	0.421	0.999	−0.388	−0.104	0.000	0.476	0.905	−0.217	−0.093	0.000	0.526	0.903	−0.177
Weak entity	0.096	0.000	0.457	**0.027**	0.211	0.156	0.000	0.504	**0.045**	0.309	0.139	0.000	0.590	**0.049**	0.234
M:N relationship	−0.105	0.000	0.379	0.972	−0.249	−0.045	0.000	0.334	0.942	−0.252	−0.006	0.000	0.313	0.562	−0.020

The values is in *bold* when the ER comprehension level is statistically higher than CD comprehension level

Table 9 Comprehension activities (without the identified CD weaknesses): Wilcoxon test

Subjects	CD versus ER			*p* value	Effect size
	Mean	Median	SD		
Fresher	0.066	0.000	0.410	**0.000**	0.161
Bachelor	0.052	0.000	0.290	**0.010**	0.120
Master	0.027	0.000	0.358	0.096	0.074

The value is in *bold* if the comprehension levels achieved with CD are statistically higher than those achieved with ER

cardinality, Ternary relationship, Generalization IS-A, and M:N relationship. In contrast, ER provided a better support for the Composite attribute, Multi-value attribute, Recursive relationship, and Weak entity building blocks. Thus, on six out of the ten investigated building blocks the support provided by CD was superior to that provided by ER. Note that these results confirm in part those achieved for the comprehension level. Only one difference leaps out: when performing modification activities, the weaknesses of CD seem to be four, adding the Recursive relationship to the three weakness identified during comprehension activities.

Table 13 shows the results of the Wilcoxon test executed for each building block to verify whether the ER performances are statistically better than those of CD. The results show that only for two building blocks, i.e., Multi-value attribute and Weak entity, the modification level of ER is statistically higher than the one achieved by CD. Note that these two weaknesses have also been identified for the comprehension task in the family of experiments reported in Sect. 4.1. However, in the comprehension task, also the Composite attribute has been identified as a weakness of the UML class diagrams. In this case, even if the difference is not statistically significant, the students achieved better results in terms of descriptive statistics with ER when answering the questions related to the Composite attribute (Table 12).

Concerning the recursive relationship, also in this case, there is no statistically significant difference, but only better descriptive statistics achieved by ER. In these cases, the subjects' preferences could help in understanding whether (i) the composite attribute could be considered a weaknesses also for the modification task, and (ii) whether it is the case to include also the recursive relationship in the list of the possible CD weaknesses.

Figure 6 reports the preferences expressed by the students for each of the analyzed building blocks. The results achieved highlighted the following.

– The ER representation of Composite attribute, Multi-value attribute, and Weak entity is clearly preferred by the students also for modification tasks. In fact, on these three building blocks on average 18 students preferred the

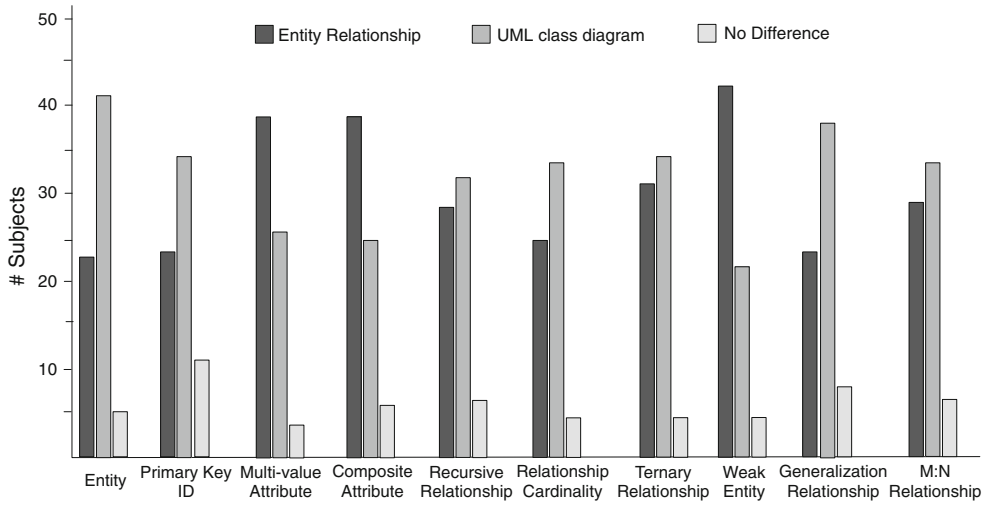

Fig. 3 Comprehension activities: preferences expressed by fresher students

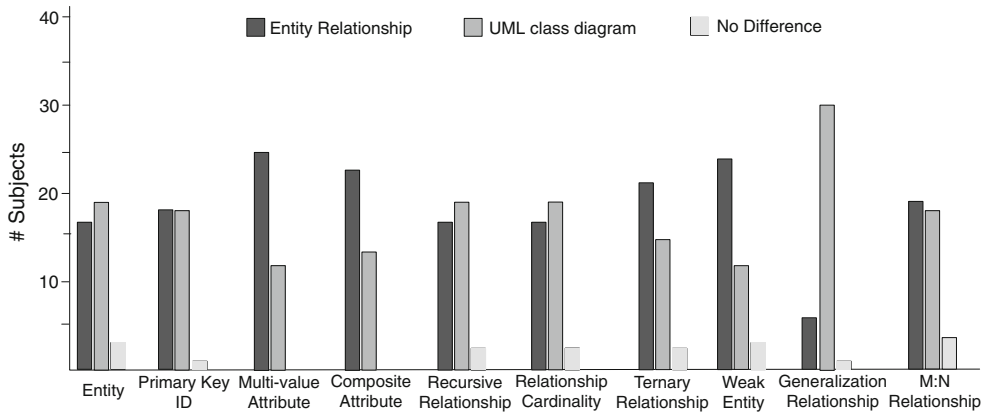

Fig. 4 Comprehension activities: preferences expressed by bachelor students

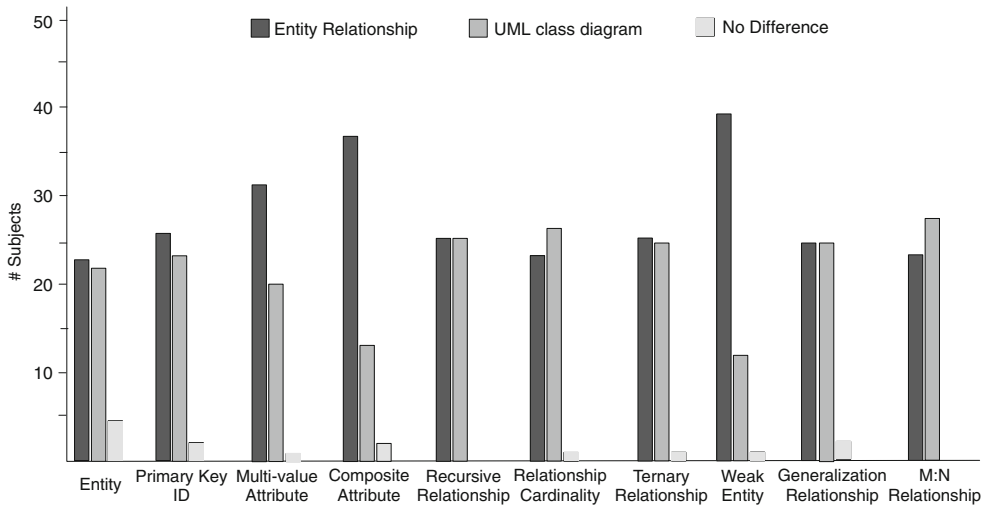

Fig. 5 Comprehension activities: preferences expressed by master's students

Table 10 Modification activities: descriptive statistics

ER			CD		
Mean	Median	SD	Mean	Median	SD
0.732	1.000	0.444	0.750	1.000	0.434

Table 11 Modification activities: Wilcoxon test

CD versus ER			p value	Effect size
Mean	Median	SD		
0.018	0.000	0.613	0.313	0.029

Table 12 Fine-grained analysis of the modification activities: descriptive statistics

Method	Element	Mean	Median	SD
ER	Entity	0.750	1.000	0.441
	Primary Key/ID	0.821	1.000	0.390
	Composite attribute	0.714	1.000	0.460
	Multi-value attribute	0.679	1.000	0.476
	Recursive relationship	0.714	1.000	0.460
	Relationship cardinality	0.857	1.000	0.356
	Ternary relationship	0.607	1.000	0.497
	Generalization IS-A	0.643	1.000	0.488
	Weak entity	0.821	1.000	0.390
	M:N relationship	0.714	1.000	0.460
CD	Entity	0.929	1.000	0.262
	Primary Key/ID	0.893	1.000	0.315
	Composite attribute	0.678	1.000	0.476
	Multi-value attribute	0.464	1.000	0.508
	Recursive relationship	0.571	1.000	0.504
	Relationship cardinality	0.929	1.000	0.262
	Ternary relationship	0.893	1.000	0.315
	Generalization IS-A	0.786	1.000	0.418
	Weak entity	0.536	1.000	0.508
	M:N relationship	0.821	1.000	0.390

ER diagrams, 4 the UML class diagrams, and 6 answered with "No preference".

- The Recursive relationship is not considered a true weaknesses of CD for the students, in fact 13 students preferred ER against the 11 of CD and 4 students had "No preference".
- There is a substantial equilibrium on the remaining building blocks. The highest difference of preferences is achieved with the Primary Key/ID building block, on which ER got three preferences more than CD.

Summarizing, we can conclude that (i) the objective analysis have highlighted two main weaknesses of CD when performing modification tasks, i.e., Multi-value attribute and Weak entity, and (ii) in the subjective analysis, the students preferences confirmed as CD weaknesses the three identified in the comprehension task, adding the Composite attribute to the two objectively identified. Finally, it is worth noting that the modification support provided by CD resulted to be statistically higher than that provided by ER when removing the three identified weaknesses from the dataset (p value 0.006). Thus, in this case, we can reject the null hypothesis H_{0_m} in favor of CD. This confirms that the removal of these three weaknesses can strongly increase the modification support of CD when working on data models.

Table 13 Fine-grained analysis of the modification activities: Wilcoxon test results

Element	ER versus CD			*p* value	Effect size
	Mean	Median	SD		
Entity	0.179	0.000	0.476	0.976	0.375
Primary Key/ID	0.071	0.000	0.539	0.784	0.132
Composite attribute	−0.036	0.000	0.693	0.406	−0.052
Multi-value attribute	−0.214	0.000	0.630	**0.045**	−0.340
Recursive relationship	−0.143	0.000	0.705	0.151	−0.203
Relationship cardinality	0.071	0.000	0.378	0.885	0.189
Ternary relationship	0.286	0.000	0.600	0.991	0.476
Generalization IS-A	0.143	0.000	0.651	0.885	0.220
Weak entity	−0.286	0.000	0.600	**0.012**	−0.476
M:N relationship	0.107	0.000	0.629	0.830	0.170

The value is in *bold* when the ER comprehension level is statistically higher than CD comprehension level

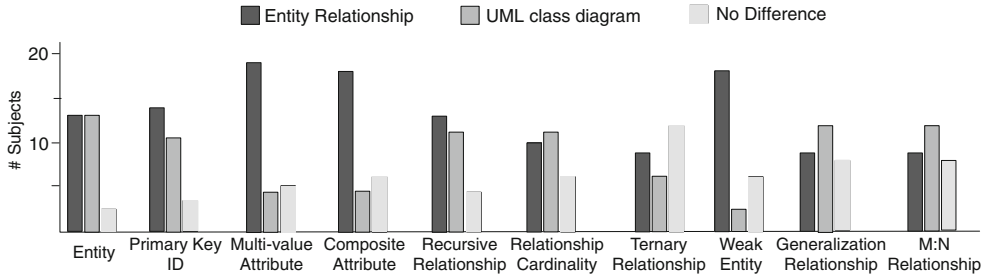

Fig. 6 Modification activities: preferences expressed by students

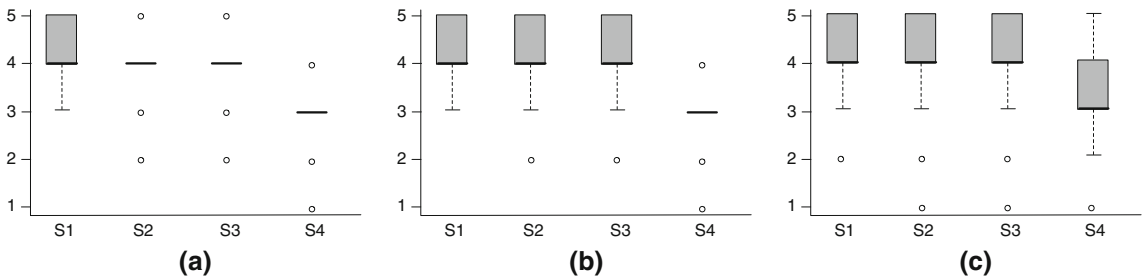

Fig. 7 Comprehension activities: answers of subjects to the post-experiment survey questionnaire

5 Validity evaluation

In this section, we discuss the threats to the validity that could affect the validity of our results, focusing the attention on construct, internal, external, and conclusion validity threats.

Construct validity threats that may be present in this experiment, i.e., interactions between different treatments, were mitigated by a proper design that allowed to separate the analysis of the different factors and of their interactions. To avoid social threats due to evaluation apprehension, students were not evaluated on their performances achieved in the experiments. Moreover, subjects were not aware of the experimental hypotheses. In addition there was no abandonment during the experiments, and the analysis of the post-experiment survey (Figs. 7, 8) revealed that the students had enough time to perform the assigned tasks (S1) and had clear the task (S2) as well as the lab objectives (S3). It is worth noting that the subjects experienced no particular difficulties (S4) when performing the comprehension tasks, while they experienced some difficulties in the modification task. This is also clear by the average modification level achieved by the subjects that is considerably lower than the comprehension level.

The experiments have been carried out to evaluate the value of ER and CD in supporting comprehension and modifi-

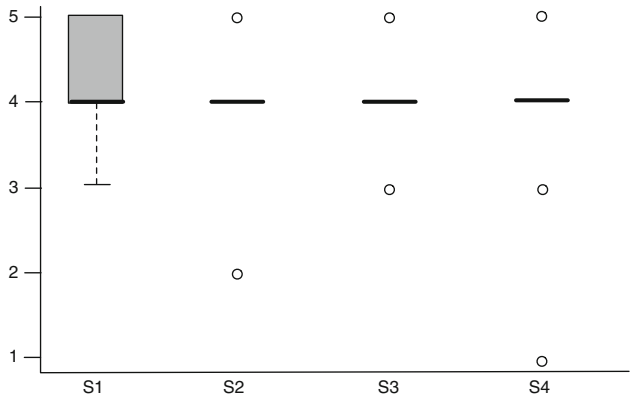

Fig. 8 Modification activities: answers of subjects to the post-experiment survey questionnaire

cation of data models. Thus, ease of comprehension and modification were the only criteria examined, since they represent the key issues for a graphical notation. However, especially, where the design of performance-critical, data-intensive software like databases is concerned, there are other key considerations as well, e.g., analyzability. One may choose to sacrifice expressiveness for analyzability or other properties. For this reason, future work will be devoted to evaluate other properties of the two notations.

Internal validity threats can be due to the learning (or tiring) effect experienced by subjects between labs. We tried to mitigate these issues through the experiment design: subjects worked, over the two labs, on different tasks and using two different design methods (i.e., ER and CD). Nevertheless, there is still the risk that, during labs, subjects might have learned how to improve their comprehension/modification performances. We tried to limit this effect by means of a preliminary training phase. In addition, as highlighted by Briand et al. [7], one possible issue related to the chosen experiment design concerns the possible information exchange among the subjects between the laboratories. To mitigate such a threat, the experimenters monitored all the students during the experiment execution to avoid collaboration and communication between them. Finally, subjects worked on two different diagrams and, even if we tried to select diagrams having comparable size, there is still the risk that one diagram might be easier than another.

We statistically analyzed the effect of co-factors (Lab and System) on the dependent variables as well as the interaction between the main factors and the co-factors. Tables 14 and 15 report the results achieved by the ANOVA test in the analysis of the interaction between the main factor, i.e., Method and the two aforementioned co-factors. The results highlight no effects of both lab and system on the subject's performances in all the experiments, as well as no interaction with the main factor. This statistically confirms the goodnesses of the used experimental design.

External validity threats concern the generalization of the results and are always present when experimenting with students. All the students have an acceptable analysis, development, and programming experience, and they are not far from junior industry analysts. Moreover, in the context of the Software Engineering course, both master's and bachelor's students had participated to software projects, where they experienced software development and documentation, including database design. Nevertheless, there are several differences between industrial and academic contexts. Thus, replications in industrial contexts would be desirable even if, given the number of subjects involved in our study (i.e., more than 150), we are confident about the reliability of the achieved results. Finally, the complexity and size of data models used to perform the comprehension tasks are comparable to those of the small/medium industrial projects.

Conclusion validity is the most important of the four validity types because it is relevant whenever someone is trying to decide if there exists a relationship in the considered observations. A definition of conclusion validity could be the degree to which conclusions we reach about relationships in our data are reasonable. Regarding our experiment, proper tests were performed to statistically analyze the results. Moreover, survey questionnaires, mainly intended to get qualitative insights, were designed using standard ways and scales [20].

6 Conclusion and future work

In this paper, we presented a comparison of ER diagrams and UML class diagrams in supporting comprehension and modification activities on data models. In particular, we involved

Table 14 Comprehension: influence of co-factors

Factor	Interaction			
	Fresher	Bachelor	Master	All
Lab	No (0.787)	No (0.163)	No (0.175)	No (0.216)
System	No (0.793)	No (0.636)	No (0.113)	No (0.229)
Method versus lab	No (0.817)	No (0.833)	No (0.305)	No (0.439)
Method versus system	No (0.793)	No (0.817)	No (0.618)	No (0.679)

Table 15 Modification: influence of co-factors

Factor	Interaction
Lab	No (0.576)
System	No (0.720)
Method versus Lab	No (0.173)
Method versus System	No (0.720)

156 subjects in three experiments aimed at analyzing the comprehension support, while one experiment with 28 subjects was performed to analyze the modification support.

The results achieved can be summarized as given below.

- *Comprehension activity* the UML class diagrams are characterized by three weaknesses related to the representation of Composite attribute, Multi-value attribute, and Weak entity, as compared to the ER diagrams. However, except for the three identified weaknesses, the UML class diagrams are generally more comprehensible than the ER diagrams.
- *Modification activity* through an objective analysis, two weaknesses of the UML class diagrams were statistically identified, i.e., Multi-value attribute and Weak entity. However, a subjective analysis confirmed also the third weakness identified for the comprehension activities, i.e., the Composite attribute. Moreover, the UML class diagrams generally provide a better support than the ER diagrams also for the modification activities, because with UML class diagrams, on six out of the ten building blocks, students achieve better results.

Comparing our results with those achieved in the previous work assessing the comprehensibility of ER diagrams and OO data models, our findings confirm those reported De Lucia et al. [11] and Palvia et al. [22] highlighting the overall better support provided by the OO representation when performing comprehension activities. Focusing on the single building blocks, the only study in the literature analyzing at a fine-grained level the comprehension support provided by the two notations is the study by Shoval and Frumermann [23], where the authors compared three building blocks of EER and OO diagrams, i.e., attribute, binary relationships, and ternary relationships. In contrast to our results, they found that the EER diagram representation of ternary relationships provides a better support when performing comprehension activities with respect to the OO diagram representation. As for the results achieved in the experiment focused on modification activities, our findings confirm those reported by De Lucia et al. [11], since, even ifUML class diagrams achieved slightly better results, the modification support provided by the two notations is not significantly different.

It is worth noting that the overall superiority of UML highlighted by De Lucia et al. [11] and Palvia et al. [22] and confirmed by our studies is somewhat surprising. In fact, one could expect that a more domain-specific language like ER, designed to represent data models, should ensure a better support during their comprehension and modification as compared to UML, which is a general-purpose modeling language. However, we also identified three weaknesses of the UML class diagrams, that are likely due to a lack of detail of this notation in representing specific concepts of database modeling (e.g., weak entities). Thus, an extension of UML class diagrams focused on providing a clearer representation of the building blocks composite attribute, multi-value attribute, and weak entity should be considered to overcome the highlighted weaknesses and improve the maintainability of data models given in terms of UML class diagrams.

As it always happens with empirical studies, replications in different contexts, with different subjects and objects, is the only way to corroborate our findings. It would be interesting to consider alternative experimental settings in several respects, but maybe the most important one is the profile of the involved subjects. Replicating this study with students/professionals having a different background would be extremely important to understand how much our findings can be generalized.

Acknowledgments We would like to thank all the students participated as subjects to the controlled experiments. We would also like to thank the anonymous reviewers for their detailed, constructive, and thoughtful comments that helped us to improve the presentation of the results in this paper. This research has been partially funded by the following projects: ORIGIN (CDTI-MICINN and FEDER,IDI-2010043(1-5)) and GEODAS-BC (Ministerio de Econom'a y Competitividad and FEDER, TIN2012-37493-C03-01).

Appendix: EasyClinic data models

Figures 9 and 10 show the ER and UML models, respectively, of one of the object system used in our study, i.e., EasyClinic.

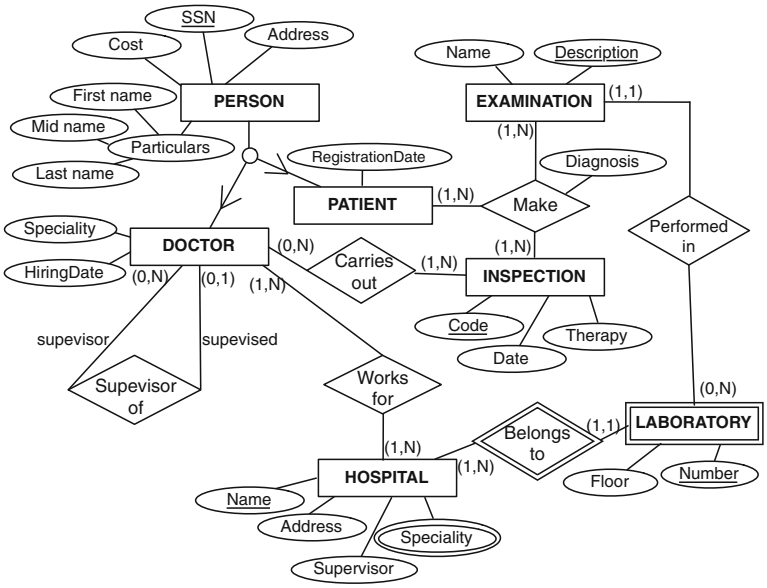

Fig. 9 ER diagram modeling the EasyClinic system

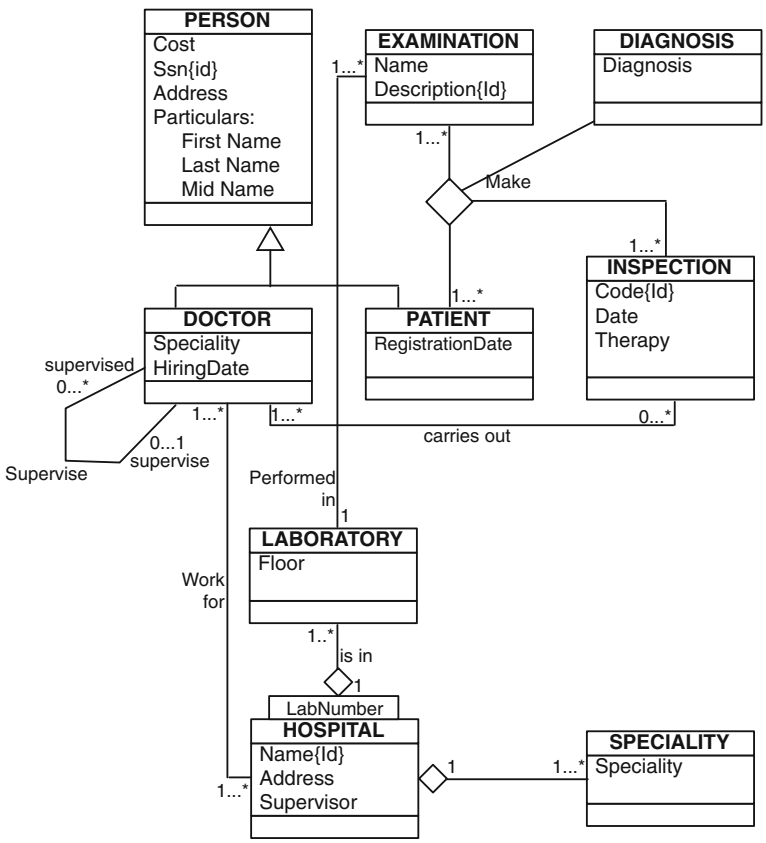

Fig. 10 CD diagram modeling the EasyClinic system

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: Modern information retrieval. Addison-Wesley, UK (1999)
2. Basili, V., Caldiera, G.: Rombach. The goal question metric paradigm. Wiley, New York (1994)
3. Basili, V.R., Selby, R.W., Hutchens, D.H.: Experimentation in software engineering. IEEE Transact. Softw. Eng. **12**(7), 758–773 (1986)
4. Batra, D., Hoffer, J., Bostrom, R.: Comparing representations with relational and eer model. Commun. ACM **33**(2), 128–139 (1990)

5. Bavota, G., Gravino, C., Oliveto, R., De Lucia, A., Tortora, G., Genero, M., Cruz-Lemus, J.: UML vs ER - experimental material. http://distat.unimol.it/reports/ERvsUML/ (2012)
6. Bock, D., Ryan, T.: Accuracy in modeling with extended entity relationship and object oriented data models. J. Database Manag. **4**(4), 30–39 (1993)
7. Briand, L., Labiche, Y., Di Penta, M., Yan-Bondoc, H.: An experimental investigation of formality in UML-based development. IEEE Transact. Softw. Eng. **31**(10), 833–849 (2005)
8. Brosey, M., Shneiderman, B.: Two experimental comparisons of relational and hierarchical database models. Int. J. Man-Mach. Stud. **10**, 625–637 (1978)
9. Cohen, J.: Statistical power analysis for the behavioral sciences, 2nd edn. Lawrence Earlbaum Associates, London (1988)
10. Conover, W.J.: Practical nonparametric statistics, 3rd edn. Wiley, New York (1998)
11. De Lucia, A., Gravino, C., Oliveto, R., Tortora, G.: An experimental comparison of ER and UML class diagrams for data modeling. Empirical Softw. Eng. **15**(5), 455–492 (2010)
12. Decorte, G., Eiger, A., Kroenke, D., Kyte, T.: An object-oriented model for capturing data semantics. In: Proceedings of the Eighth International Conference on Data, Engineering, pp. 126–135 (1992)
13. Durding, B., Becker, C., Gould, J.: Data org. Human Fact. **19**, 1–14 (1977)
14. Juhn, S., Naumann, J.: The effectiveness of data representation characteristics on user validation. In: Proceedings of the 6th International Conference on, Information Systems, pp. 212–226 (1985)
15. Kampenes, V.B., Dybå, T., Hannay, J.E., Sjøberg, D.I.K.: A systematic review of effect size in software engineering experiments. Inf. Softw. Technol. **49**(11–12), 1073–1086 (2007)
16. Liao, C., Palvia, P.: The impact of data models and task complexity on end-user performance: an experimental investigation. Int. J. Human Comput. Stud. **52**, 831–845 (2000)
17. Liao, C., Shih, M.: The effects of data models and training degrees on end users' data representations. MIS Review **8**, 1–20 (1998)
18. Miller, G.A.: The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychol. Rev. **63**(2), 81–97 (1956)
19. Navathe, S.B.: Evolution of data modeling for databases. Commun. ACM. **35**(9), 112–123 (1992)
20. Oppenheim, A.N.: Questionnaire design. Pinter Publishers, Interviewing and Attitude Measurement (1992)
21. Palvia, P.: On end-user computing productivity. Inform. Manag. **21**, 217–224 (1991)
22. Palvia, P., Lio, C., To, P.: The impact of conceptual data models on end-user performance. J. Database Manag. **3**(4), 4–15 (1992)
23. Shoval, P., Frumermann, I.: OO and EER conceptual schemas: a comparison of user comprehension. J. Database Manag. **5**(4), 28–38 (1994)
24. Shoval, P., Shiran, S.: Entity-relationship and object-oriented data modeling–an experimental comparison of design quality. Data Knowl. Eng. **21**(3), 297–315 (1997)
25. Sirkin, R.M.: Statistics for the social sciences. Sage Publications, California (2005)
26. Wohlin, C., Runeson, P., Host, M., Ohlsson, M.C., Regnell, B., Wesslen, A.: Experimentation in Software Engineering–an introduction. Kluwer, Dordrecht (2000)

Author Biographies

Gabriele Bavota was born in Napoli (Italy) on November, 19th, 1985. He received (cum laude) the Laurea in Computer Science from the University of Salerno (Italy) in 2009 defending a thesis on Traceability Management advised by Prof. Andrea De Lucia and Dr. Rocco Oliveto. He is currently a PhD student at the Department of Mathematics and Informatics of the University of Salerno under the supervision of Prof. Andrea De Lucia. His research interests include refactoring of software systems, traceability management, information retrieval, software maintenance and empirical software engineering. He serves and has served on in the organizing and program committees of international conferences in the field of software engineering. In particular, he was the web chair of WCRE 2012 and he will be publicity co-chair of ICPC 2013. He is student member of IEEE.

Carmine Gravino received the Laurea degree in Computer Science (cum laude) in 1999, and his PhD in Computer Science from the University of Salerno (Italy) in 2003. Since march 2006 he is assistant professor at the University of Salerno. His research interests include software metrics and techniques to estimate web application development effort, software-development environments, design pattern recovery from object-oriented code, evaluation and comparison of notations, methods, and tools supporting software development and maintenance. He has published more than 80 papers on these topics in international journals, books, and conference proceedings.

Rocco Oliveto received (cum laude) the Laurea in Computer Science from the University of Salerno (Italy) in 2004. He received the PhD in Computer Science from the University of Salerno (Italy) in 2008. From 2008 to 2010 he was research fellow at the Department of Mathematics and Informatics of the University of Salerno. In 2011 he joined the STAT Department of the University of Molise where he is currently assistant professor. His research interests include

traceability management, information retrieval, software maintenance and evolution, and empirical software engineering. He has published more than 50 papers on these topics in international journals, books, and conference proceedings. He serves on the editorial board of the Advances in Software Engineering. He serves and has served as organizing and program committee member of international conferences in the field of software engineering. In particular, he was the program co-chair of TEFSE 2009, the Traceability Challenge Chair of TEFSE 2011, the Industrial Track Chair of WCRE 2011, the Tool Demo Co-chair of ICSM 2011 and he will be the program co-chair of WCRE 2012. Dr. Oliveto is member of IEEE, ACM, and IEEE-CS Awards and Recognition Committee.

Andrea De Lucia received the laurea degree in computer science from the University of Salerno, Italy, in 1991, the MSc degree in computer science from the University of Durham, UK, in 1996, and the PhD degree in electronic engineering and computer science from the University of Naples "Federico II", Italy, in 1996. He is a full professor of software engineering and the Director of the International Summer School on Software Engineering at the University of Salerno. Previously, he was with the Department of Engineering and the Research Centre on Software Technology (RCOST) at the University of Sannio. He is actively consulting in industry and has been involved in several research and technology transfer projects conducted in cooperation with industrial partners. His research interests include software maintenance, program comprehension, reverse engineering, reengineering, migration, global software engineering, software configuration management, workflow management, document management, empirical software engineering, visual languages, web engineering, and e-learning. He has published more than 150 papers on these topics in international journals, books, and conference proceedings and has edited books and journal special issues. He serves on the editorial board of Journal of Software: Evolution and Process and other international journals and on the organizing and program committees of several international conferences in the field of software engineering. Prof. De Lucia is a senior member of the IEEE and the IEEE Computer Society. He was also at-large member of the executive committee of the IEEE Technical Council on Software Engineering (TCSE) and committee member of the IEEE Real World Engineering Project (RWEP) Program.

Genoveffa Tortora is a full professor in Computer Science at the University of Salerno, since 1990, where she has been Department Chair, and then Dean of the Faculty of Sciences. She has been (1993–1999) the Vice-President of GRIN (Gruppo di Informatica), the Italian Association of University Professors of Computer Science. She is IEEE senior member and member of ACM and IAPR. She has been General chair, Program chair and program committee member of several international conferences and serves on the Editorial Board of several scientific journals. She has been responsible of several national research projects, evaluator and scientific committee member of European and national projects. She has co-authored more than 200 papers published in scientific journals or proceedings of refereed conferences, and has co-edited three books. At the University of Salerno, she founded and directed the Department of Mathematics and Informatics and the software engineering, GIS, VR, and visual computing labs. Her research interests are in the Software Engineering and Information Systems areas, and include software development environments, human-computer interaction, visual languages, databases and geographic information systems, image processing and biometric systems.

Marcela Genero is Associate Professor at the Department of Information Systems and Technologies at the University of Castilla-La Mancha, Ciudad Real, Spain. She received her MSc degree in Computer Science in the Department of Computer Science of the University of South, Argentine in 1989, and her PhD at the University of Castilla-La Mancha, Ciudad Real, Spain in 2002. Her research interests are: empirical software engineering, research methods, software metrics, conceptual models quality, quality in MDD, evaluation of serious games, benefits of using UML benefits in software development, etc. Marcela Genero has published in prestigious journals (Information and Software Technology, Journal of Software Maintenance and Evolution: Research and Practice, Data and Knowledge Engineering, Empirical Software Engineering, Software Quality Journal, Information Sciences, Systema ans Software modelling, etc.), and conferences (CAiSE, ER, MODELS/UML, ISESE, METRICS, ESEM, EASE, etc). She edited with Mario Piattini and Coral Calero the books titled "Data and Information Quality" (Kluwer, 2001) and "Metrics for Software Conceptual Models" (Imperial College, 2005). She is member of the International Software Engineering Research Network (ISERN).

José A. Cruz-Lemus is an Associate Professor at the Department of Information Systems and Technologies at the University of Castilla-La Mancha, Ciudad Real, Spain. He is PhD in Computer Science from the same university. His main research interests are empirical software engineering, software metrics and UML models quality. He has published his works in several journals (Empirical Software Engineering, Information Sciences, Information and Software Technologies, Journal of Systems and Software, etc.), conferences (MoDELS, E/R, ESEM, etc.), and several book chapters.