Quality Assessment of UML Class Diagrams

- A Study Comparing Experts and Students –

Bilal Karasneh, Dave R. Stikkolorum, Enrique Larios

Leiden Institute of Advanced Computer Science
Leiden University
Leiden, the Netherlands
{b.h.a.karasneh, d.r.stikkolorum,
e.larios.vargas}@liacs.leidenuniv.nl

Michel R.V. Chaudron

Joint Dept. of Computer Science and Engineering
Chalmers Univ. of Technology & Gothenburg Univ.
Gothenburg, Sweden
chaudron@chalmers.se

***Abstract*.** In this paper, we present an experiment conducted for comparing how experts and students assess the quality of class diagrams. Six quality attributes were addressed: Understandability, Layout, Extensibility, Modifiability, Completeness and Correctness. From this study, we aim to find out how well students are capable of evaluating the quality of UML designs. Moreover, we aim to learn which features experts and students use for assessing the quality attributes of class diagrams. The study reveals that experts and students' assessment of the six quality attributes differ significantly. However, a qualitative analysis of experts and students' feedback suggests that students use similar features as experts use for assessing the quality of diagrams. Hence peer-feedback from students can be useful in educational settings.

Keywords: Empirical Studies; Software Engineering Education; Software Design Education; UML Class diagram; UML Quality.

1 Introduction

Quality is a multidimensional concept, and in practice people make different interpretations of the same concept. Nowadays UML [1] is the de-facto standard for modeling software systems. UML offers a rich set of symbols for describing software. Modern software designs contain many abstraction levels, and designing them is an iterative process [2]. The collection of design documents is an important part of the system documentation which will be used and maintained for a long time by a development organization. In the software engineering class, students should understand the importance of software models and their design process.

Software design is considered as a difficult task in comparison with programming for many students. One reason is that current Integrated Development Environments (IDEs) help students to improve the quality of their code, for example using code metrics such as maintainability index and cyclomatic complexity. On the other hand, students do not get much useful feedback on their design models. Current Computer-Aided Software Engineering (CASE) tools do not give any hints to improve models, except some layout algorithms and syntax. Although there are some proposed tools that give students some feedback about their design, these tools still suffer from many limitations, such as availability and connectivity [3, 4].

During programming courses, students are taught about the quality of the source code (including for example, naming and layout conventions and API design guidelines). In software engineering courses, students are taught to understand basic modeling concepts and modeling notations. For instance what UML diagrams are, when to use a class diagram, how to create a sequence diagram, what are elements of use case diagrams. Many teachers focus on teaching students the proper use of syntactical elements in creating UML diagrams. In both programming and modeling, the completeness and correctness are key attributes of the quality of a solution. However, there are no clear or specific rules or guidelines about assessing quality attributes of designs. This leaves students to self-learning on how to make a good design.

For proper learning of modeling and designing, students need to get feedback from their teachers, or from peers to evaluate their design. For example: this class should have more operations, this class name should be changed, and this operation should have more parameters. One way of providing feedback could be to use a method for assessing the quality of UML models. Unfortunately, currently no such method exists.

In this study, we explore the use of ISO standards for software quality as a basis for reviewing UML models. Software product quality models such as ISO/IEC 25010 [5] have categories of quality characteristics, and each characteristic is composed of a set of sub-characteristics. One difficulty with this standard is that there are many ways of interpreting every characteristic.

In this experiment, we want to measure the ability of students to evaluate their design and other students' designs. Also, we want to study whether the evaluations of students are consistent with (significantly correlated with) those of experts. In addition to the quantitative analysis, we do a qualitative comparison of the feedback provided by students and experts.

In this paper, we present results of our experiment on the evaluation of the quality of UML class diagram. We asked students to perform a modeling task, and then to evaluate their own models and to evaluate models of other students in terms of six quality attributes: Understandability, Layout, Extensibility, Modifiability, Completeness, and Correctness. Then we asked five experts to evaluate students' models in terms of the same quality attributes.

The aim of this study is to empirically investigate whether students' evaluations are different from experts' evaluation, and what are the differences and similarities between students' feedback and experts' feedback. The differences and similarities are useful to assess if the feedback of students can be useful for improving the quality of the design.

This paper is organized as follows: Section two describes the related work. Section three describes the method used in the experiment. The evaluation of models is in section four. The results and the analysis are presented in section five. The results are discussed in section six. Threats to validity are in section seven. The conclusion and future work are in section eight.

2 Related work

We follow the general guideline for experimental design and analysis from [6, 7]. Tichy [8] shows that there are good reasons for conducting experiments with students, for testing experimental design and initial hypotheses, or for educational purposes. Depending on the actual experiment, students may also be representative of inexperienced professionals [9]. Boustedt [10] made an empirical study on how students understand class diagram using phenomenographic investigation. He found that the purpose of class diagrams and various elements of the UML notation are understood in a varied way. He recommended that teachers should put more effort in assessing skills in proper usage of the basic symbols and models, and students should have opportunities to practice collaborative design. Our experiment is different from [10] as we ask the students to evaluate class diagram directly in terms of six quality attributes, and we ask them to give feedback and explanation about their evaluation. Ali et. al. [11] presented the UML class diagram assessor (UCDA) that evaluates class diagrams automatically based on their structure, correctness and language used. The aim of the proposed assessor is to guide students to represent class diagram correctly. The results of our experiment are useful for the kind of assessors in [11] because from the information collected we know which kind of feedback experts and students use for describing violation of modeling conventions and/or models improvement. Hoggart et. al. [12] found that students understand the theory in classroom settings but find it hard to apply in exercises and tasks. They proposed a tool that gives students feedback about their diagrams by comparing it with the model answer. Generating feedback based on model answers is a bit difficult because in modeling there is typically more than one solution. Because it is difficult to find sufficient number of experts, we decide to explore whether the feedback from peer students can help improve model quality.

Kaneda et. al. [13] show that class diagrams reflect the cognitive structure of English based cognitive linguistic. They found that there is impedance mismatch for understanding of class diagram from students who are not native English speakers. In our experiment, our students are a mix of various nationalities. Before admission, our students have to pass an English (TOEFL) test, and therefore we consider their English language skills will not influence our study.

Aguilera et. al. [14] show that names of elements in UML diagram have a strong influence on the understandability. They proposed guidelines for naming various kinds of elements in UML.

Selic [15] shows that understandability is the most important characteristics of models. In our experiments, we show in Table 4 the features that experts and students take focus on for assessing understandability of models.

3 Method

In this section, we explain our approach, the participants of the experiment and the evaluation forum that the participants use for assessing class diagrams.

3.1 Approach

We conducted an experiment in Leiden University in which both experts and students participated. We gave the students a modeling task and asked them to use the StarUML CASE tool [16] to create their models. Upon completion of the modeling task, they had to upload their models to the Models Repository[1] and evaluate their models based on six quality attributes. Also, they had to mention their background: academia, industry or both, and their experience in UML modeling (less than one year, <1,2>, <2,5> or expert). Subsequently, they had to evaluate other students' design based on the same six quality attributes. We asked students to explain their evaluations through feedback comments for each quality attribute. Students had a trial assignment two weeks before the experiment with another modeling task. This trial is important for the students to be prepared for the experiment. It helped them in getting acquainted with the type of assignment, the tools and thereby limits the learning effects.

We also asked the experts to evaluate students' models based on the six quality attributes and to give a feedback of the models and describe their evaluations.

Participants had to register to the UML Repository to access the evaluation form. From their profile page, they can visualize the number of models evaluated, a link to the modeling task and another link to the evaluation form.

3.2 Participant

The participants are: experts and students.

1. Experts

Five experts joined this experiment. Each expert has at least five years of experience in UML modeling and software design. Two experts are teachers of software modelling and software engineering for at least three years. One of those two experts also worked in industry. The other three experts are PhD students in the area of software engineering since 2011.

2. Students

46 master students of the ICT in business M.Sc. program[2] in Leiden University participated in the experiment during their course on software engineering. All of them

---

1 http://Models-db.com

2 This is a degree in the Science faculty of the University of Leiden. This degree is a mix of topics from Information System, Software Engineering and Management and Business Administration.

have less than one year experience in UML modeling. Some of them have some (mostly short) background in industry, but most of them have an academic background (just finished their B.Sc. degree).

3.3 Evaluation Form

The form for evaluating class diagrams was implemented through an online system. This system showed a form that contains:

(1) The number of models that were evaluated by the participant (out of 46 models).

(2) An image of a student's class diagram. The image is created by other students and has not been evaluated by the participant before.

(3) A list of radio-buttons for entering assessments for 6 quality attributes. Each quality attribute can be rated on a scale ranging from 1 to 8:

- For Understandability, Extensibility and Modifiability: (1) is difficult, (8) is easy.
- For Layout: (1) is complex, (8) is simple.
- For Completeness: (1) is not complete, (8) is complete.
- For Correctness: (1) is not correct, (8) is correct.

(4) A comment box. For each quality attribute participants can submit details about their evaluation using a text box. We perform qualitative analysis of the comments provided by experts and students to figure out which features they focus on when they assess the quality of a model.

A submit button. Stores the assessment and navigates participants to evaluate another design.

4 Models Evaluation

For comparing the evaluation between experts and students, we use Multivariate General Linear Model (MGLM). This model is used because it considers multiple dependent variables and multiple independent variables. We also use bootstrapping [17], which is a method that approximates the sampling distribution of the sample mean. In our experiment, the dependent variables are the six quality attributes, and the independent variables are the assessors (experts and students). We use IBM SPSS [18] as statistics tool.

4.1 Experts evaluation and students evaluation for their models (self-evaluation)

When students uploaded their models, they evaluated them. Figures 1 and 2 show the average evaluations of experts and students for understandability and layout respectively.

Each class diagram was evaluated by at least three experts and one student (each student evaluated his/her model). For making this comparison, we do resampling (bootstrapping) of 1000 times of size 121 for the experts evaluation and independently the same resampling time of size 46 for the students evaluation.

From Figures 1 and 2, we show that experts and students differ in the most cases (good diagrams and bad diagrams).

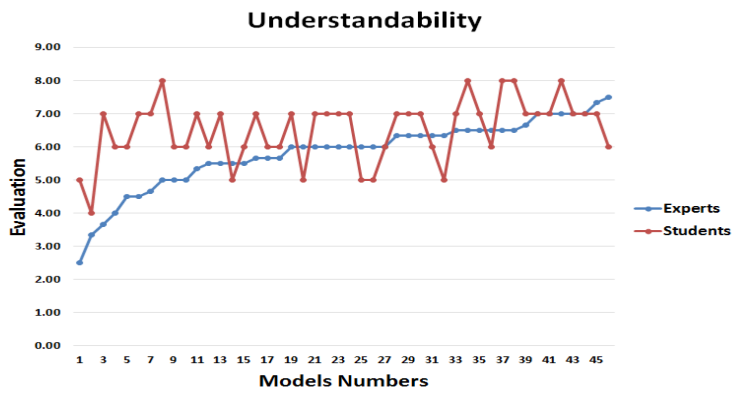

Fig. 1. Experts and students evaluation (self-evaluation) for Understandability

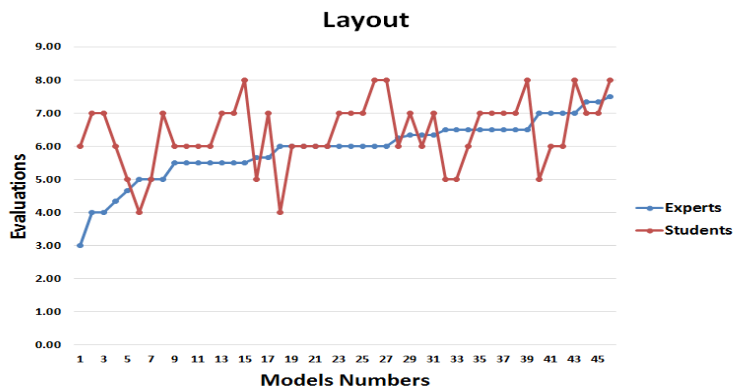

Fig. 2. Experts and students evaluation (self-evaluation) for Layout

4.2 Experts evaluation and students (peer) evaluation

From the set of evaluations, we leave out seven class diagrams because the standard deviation of student's evaluation is high. This leaves a total of 39 class diagrams. Each expert was asked to evaluate at least 20 class diagrams. We have 95 model-evaluations from experts per each quality attribute. Moreover, each student evaluated at least 10 class diagrams. We have 435 evaluations in total from students for each quality attribute. Figures 3 and 4 show the average evaluations of experts and students for understandability and layout respectively. The evaluation is sorted in ascending order based on experts' evaluation. From Figures 3 and 4, students assessment is mostly higher than the experts' for understandability and layout, and sometimes they are close, especially when the model is good, but they differ much when models are not good. Each class diagram was evaluated by at least three experts and ten students. For making this comparison between experts and students evaluations, we do resampling of a 1000 times of size 95 for the experts evaluation and independently the same resampling time of size 435 for students evaluation.

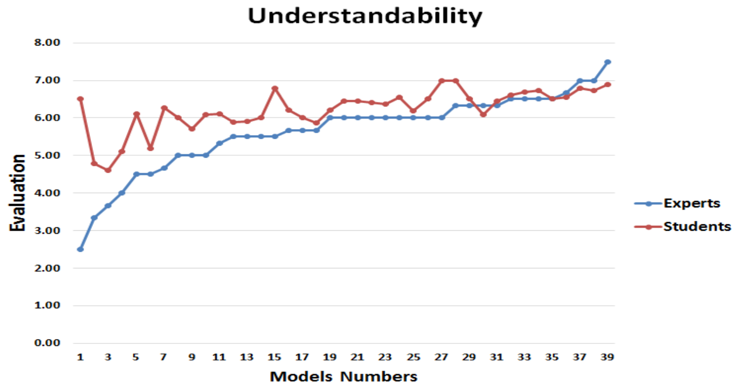

Fig. 3. Experts and students evaluation (peer-evaluation) for Understandability

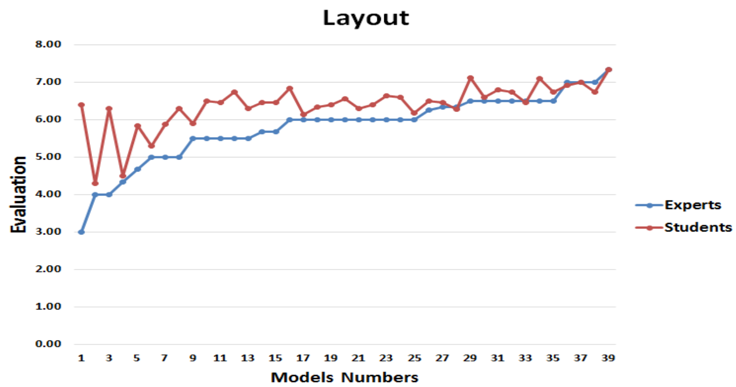

Fig. 4. Experts and students evaluation (peer-evaluation) for Layout

5 Results and Analysis

Table 1 shows the results of MGLM. Table 1 shows that there is significant difference between expert evaluation and students self-evaluation. From Table 1, also we see there is significant difference between experts evaluations and students peer-evaluations. Table 2 shows the description of experts and students evaluations. From Table 2, all means of experts evaluations are less than the means of students evaluations in both cases (self/peer evaluation).

For analyzing the evaluations, we show the correlation metrics between experts and students evaluations in Table 3. From Table 3, it is possible to see many high correlations between quality attributes. First, the correlation between experts evaluation, understandability has a high correlation with all quality attributes. Second, on the student side, understandability also has a high correlation with most other quality attributes. We notice that the correlation between understandability and layout for students evaluations is higher than experts evaluations. Third, regarding the correlation between experts' and students evaluations, the highest correlation is between experts understandability and students understandability. The second highest correlation is between experts and students evaluations for layout.

Table 1. Results of Multivariate General Linear Model

Dependent Variable	*Assessors*		*95% Sig.*
Understandability	*Experts*	*Self-Evaluation*	0.000
		Peer-Evaluation	0.021
Layout	*Experts*	*Self-Evaluation*	0.003
		Peer-Evaluation	0.006
Extensibility	*Experts*	*Self-Evaluation*	0.003
		Peer-Evaluation	0.009
Modifiability	*Experts*	*Self-Evaluation*	0.001
		Peer-Evaluation	0.011
Completeness	*Experts*	*Self-Evaluation*	0.001
		Peer-Evaluation	0.053
Correctness	*Experts*	*Self-Evaluation*	0.000
		Peer-Evaluation	0.004

We qualitatively analyze experts and students (peer-evaluation) comments/feed-back for their evaluation. This analysis is important to see the features that experts and students use for assessing the quality of class diagrams. We discuss the feedback of three of the quality attributes: understandability, layout, and completeness. From Table 3, understandability and layout are the highest correlated quality attributes between experts and students peer-evaluation (0.70 and 0.67) respectively. Understandability is also the most correlated quality attributes with others quality attributes. We choose completeness because from Table 1, experts and students almost differ with significant 0.053.

Table 2. Description of Experts and students Evaluation

Dependent Variables	Experts			Self-Evaluation			Peer-Evaluation		
	Mean	***95% Confidence Interval***		***Mean***	***95% Confidence Interval***		***Mean***	***95% Confidence Interval***	
		Lower Bound	***Upper Bound***		***Lower Bound***	***Upper Bound***		***Lower Bound***	***Upper Bound***
Understandability	5.71	5.44	5.97	6.51	6.27	6.76	6.22	5.96	6.49
Layout	5.74	5.45	6.02	6.40	6.14	6.67	6.38	6.09	6.66
Extensibility	5.59	5.33	5.84	6.18	5.94	6.42	6.14	5.89	6.39
Modifiability	5.52	5.26	5.78	6.20	5.96	6.44	6.06	5.80	6.32
Completeness	5.77	5.47	6.08	6.53	6.25	6.82	6.30	5.99	6.60
Correctness	5.07	4.74	5.41	6.31	6.00	6.62	5.86	5.53	6.20

Table 3. Correlation of Experts and Students Evaluation

	E_Unders.	***E_Layout***	***E_Extens.***	***E_Modif.***	***E_Compl.***	***E_Correct.***	***S_Unders.***	***S_Layout***	***S_Extens.***	***S_Modif.***	***S_Compl.***	***S_Correct.***
E_Unders.	1											
E_Layout	**0.74**	1										
E_Extens.	**0.77**	0.63	1									
E_Modif.	**0.84**	**0.71**	**0.88**	1								
E_Compl.	0.67	0.65	**0.70**	0.68	1							
E_Correct.	**0.71**	0.57	**0.83**	**0.85**	**0.74**	1						
S_Unders.	**0.70**	0.62	0.59	0.52	0.44	0.41	1					
S_Layout	0.57	0.67	0.56	0.53	0.47	0.47	**0.81**	1				
S_Extens.	0.61	0.57	0.62	0.53	0.47	0.45	**0.86**	**0.83**	1			
S_Modif.	0.60	0.62	0.57	0.56	0.51	0.37	**0.85**	**0.81**	**0.88**	1		
S_Compl.	0.54	0.37	0.49	0.46	0.58	0.51	0.62	0.56	0.59	0.56	1	
S_Correct.	0.59	0.45	0.58	0.56	0.66	0.61	0.62	0.59	0.60	0.61	**0.89**	1

We use NVivo10[3] for qualitatively analysing experts and students comments. In their comments, they explain how they evaluated models quality attributes, and features they used for evaluation. Table 4 shows the 11 features of models that experts and students used for assessing understandability, where: (i) 64% of the features are used by both experts and students. (ii) 27% of the features are used by students and are not used by experts. (iii) 9% are used by experts and not used by students.

[3] http://www.qsrinternational.com/

Table 4. Features that experts and students focus on when they evaluate Understandability

Features	***Experts***	***Students***
Easy to read	-	X
Completeness	X	X
Extra information	X	X
Complexity	X	X
Correctness	X	X
Data type	-	X
implementation	X	-
Layout	X	X
Class, attributes and operations name	X	X
Relationships name	-	X
Number of classes, operation, and attributes	X	X

Table 5 shows that experts and students used 12 features for assessing the layout, where: (i) 58% of the features are used by both experts and students. (ii) 9% of the features are used by students, but not used by experts. (iii) 33% used by experts, but not used by students.

Table 5. Features that experts and students focus on when they evaluate Layout

Features	***Experts***	***Students***
Classes Hierarchy, alignment	X	X
Classes with similar size	X	-
Complexity	X	X
Number of classes, attributes and operations	-	X
Distance between Classes	X	X
Rectilinear edges and diagonal edges	X	-
Line Style(overlapping, crossing, bend)	X	X
Good Class name	X	-
Neat and chaotic structure	X	X
Easy to read	X	X
Same Layout for same/All relationships	X	-
Extra information	X	X

Table 6 shows that experts and students used 10 features for assessing completeness, where: (i) 60% of the features are used by both experts and students. (ii) 30% of the features are used only by students. (iii) 10% are used only by experts.

Although we see experts focus on more features in Table 5, it may be that students are interested in many of the same features – yet they do not mention them clearly in their feedback.

Table 6. Features that experts and students focus on when they evaluate Completeness

FEATURES	*EXPERTS*	*STUDENTS*
MODEL ABSTRACTION	-	X
FUNCTIONALITY	X	X
STRANGE RELATIONSHIPS	X	X
MISSING CLASSES, ATTRIBUTES, AND OPERATIONS	X	X
DATA TYPES	-	X
MULTIPLICITY	X	X
FUNCTIONS PARAMETERS	-	X
RELATIONSHIPS NAME	X	X
REQUIREMENTS	X	X
MODEL SEMANTICS	X	-

6 Discussion

From the MGLM results in Table 1, we conclude that there is a significant difference between the evaluation of experts and students. The results also show that peer-evaluation of students is closer to the evaluation of experts than self-evaluation (because the mean difference is bigger between experts and self-evaluation than with peer-evaluation for all quality attributes as shown in Table 2). We explain this by the different viewpoints in the peer-evaluation. Different points of view may have caused different evaluations that on average became more reliable, or at least better than the self-evaluation.

The qualitative analysis of experts' and students' comments shows that students use most features that experts use for assessing the quality of class diagrams. In Tables 4, 5 and 6 we summarize the features that experts and students use for assessing understandability, layout and completeness respectively. In the qualitative analysis, we only take into account the issues that can be clearly identified in the feedback. We notice that feedback from experts is more specific than that from students. For example, some students mentioned they did not like a class, but they did not mention what was the problem with this class: name, size, position, etc. However, this general feedback can still be useful because it can be considered as general hints that directs students to a particular area where they still themselves need to find out what needs to be improved.

We conclude that due to students use similar features for assessing the quality of class diagram as experts use, their feedback is useful for improving their models. So we expect that if students exchange their feedback about their models, this will be a valuable source of feedback for learning and improving their models. Also, we expect that students can make a better evaluation if they do this in a group because they can then discuss their different viewpoints and improve their evaluation.

7 Threats To Validity

In this section, we discuss the threats to validity of our study.

7.1 Internal Validity

We ensured that students are familiar with class diagrams. The experiment was conducted at the end of the Software Engineering course where they had a trial two weeks before the experiment. The participants did not know the aim of our experiments, nor the measures that we are looking for, in order to avoid their expectations from biasing the results.

7.2 External Validity

There were 46 students participants in the experiment. To mitigate their representativeness, we only address their experience level with UML, and their background (academic, industry or both). About the modeling task, we chose a system from an application domain that should be familiar to students. The choice of tasks may not surely reflect what students do with the models, but they reflect the degree to which students comprehend the models and how they assess them.

8 Conclusion and Future Work

In this paper, we presented an experiment that investigates the difference between experts and students in assessing the quality of UML class diagram empirically. We made two comparisons: first between experts' and students' self-assessment. Second, between experts' and students' peer- assessment. We use the Multivariate General Linear Model as a statistical method for making those comparisons. The results show that experts' and students' (self- assessment) are different in terms of means (95% significance). The students' self- assessments are higher than experts' assessments in terms of mean for the quality attributes used in the experiment. The results also show that experts and students (peer-evaluation) are different in terms of mean (95% significance). The students peer- assessments are higher than experts assessments in terms of mean for all quality attributes used in the experiment. Our interpretation is that especially for poor quality models, students are reluctant to give fellow students below-passing grades.

Analyzing the correlation between experts' assessments and students' peer- assessments shows that understandability is the highest correlated quality attribute, and that layout is the second highest. The correlation also shows that understandability is correlated with most of the other quality attributes based on both experts' assessments and students' assessments.

We did a qualitative analysis of experts' feedback and students' feedback in peer-assessments. From this, we observe that students mostly use similar features as experts for their assessments. So we conclude that feedback from students is valuable and can be useful for other students for improving their designs.

In the future, we are planning to replicate the experiment and ask students to assess the quality of class diagram in groups. We believe that having an online community for students where they can exchange their models, and their feedback is very useful for improving modeling education. So we are establishing this community with the collaboration of some experts. From this community, students and experts can upload their models and exchange their feedback.

9 References

1. Miliev, D.: On the semantics of associations and association ends in uml. Software Engineering, IEEE Transactions on. 33, 238–251 (2007).
2. Craig, L.: Applying UML and patterns. Tredje upplagan, Prentice Hall. (2002).
3. Hasker, R.W.: UMLGrader: An Automated Class Diagram Grader. J. Comput. Sci. Coll. 27, 47–54 (2011).
4. Hasker, R.W., Rowe, M.: UMLint: Identifying Defects in UML Diagrams. In: 2011 Annual Conference & Exposition. ASEE Conferences, Vancouver, BC (2011).
5. ISO, I.: ISO/IEC 25010". 2011. Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—System and software quality models. (2011).
6. Christensen, L.B., Johnson, B., Turner, L.A.: Research methods, design, and analysis. Allyn & Bacon (2011).
7. Montgomery, D.C.: Design and analysis of experiments. John Wiley & Sons (2008).
8. Tichy, W.F.: Hints for reviewing empirical work in software engineering. Empirical Software Engineering. 5, 309–312 (2000).
9. Sjøberg, D.I., Hannay, J.E., Hansen, O., Kampenes, V.B., Karahasanovic, A., Liborg, N.-K., Rekdal, A.C.: A survey of controlled experiments in software engineering. Software Engineering, IEEE Transactions on. 31, 733–753 (2005).
10. Boustedt, J.: Students' different understandings of class diagrams. Computer Science Education. 22, 29–62 (2012).
11. Ali, N.H., Shukur, Z., Idris, S.: A design of an assessment system for UML class diagram. In: Computational Science and its Applications, 2007. ICCSA 2007. International Conference on. pp. 539–546. IEEE (2007).
12. Hoggarth, G., Lockyer, M.: An automated student diagram assessment system. In: ACM SIGCSE Bulletin. pp. 122–124. ACM (1998).
13. Kaneda, S., Ida, A., Sakai, T.: Understanding of Class Diagrams Based on Cognitive Linguistics for Japanese Students. In: Knowledge-Based Software Engineering. pp. 77–86. Springer (2014).
14. Aguilera, D., Gómez, C., Olivé, A.: A complete set of guidelines for naming UML conceptual schema elements. Data & Knowledge Engineering. 88, 60–74 (2013).
15. Selic, B.: The pragmatics of model-driven development. IEEE software. 20, 19–25 (2003).
16. Lee, M., Kim, H., Kim, J., Lee, J.: StarUML 5.0 Developer Guide. The Open Source UML/MDA Platform.
17. Efron, B., Tibshirani, R.J.: An introduction to the bootstrap. CRC press (1994).
18. Spss, I.: IBM SPSS statistics version 21. Boston, Mass: International Business Machines Corp. (2012).