# EDUCATIONAL CRITERIA FOR EVALUATING SIMPLE CLASS DIAGRAMS MADE BY NOVICES FOR CONCEPTUAL MODELING

Mizue Kayama, Shinpei Ogata, David K. Asano and Masami Hashimoto
*Faculty of Engineering, Shinshu University, Japan*

**ABSTRACT**

Conceptual modeling is one of the most important learning topics for higher education and secondary education. The goal of conceptual modeling in this research is to draw a class diagram using given notation to satisfy the given requirements. In this case, the subjects are asked to choose concepts to satisfy the given requirements and to correctly extract the relations between objects and services. In this paper, we aim to propose criteria for evaluating conceptual modeling errors made by university freshmen. To achieve this research goal, we quantitatively analyzed class diagrams made by novice learners. Based on the results of three types of experiments, we propose 12 criteria, which are divided into 4 types, for evaluating class diagrams made by novices.

**KEYWORDS**

Conceptual modeling, class diagram, criteria, quantitative analysis, novice education

## 1. INTRODUCTION

Not only in undergraduate or graduate schools but also in pre-university education, conceptual modeling is one of the most important learning topics. These days, educational methods or learning courses related to conceptual modeling have been explored in many educational institutes, academic conferences and academic journals [Bezivin 2009, Börstler 2004, Fuller 2010, Kramer 2007, Moisan 2010, Sendall 2003]. The goal of conceptual modeling in this research is to draw a class diagram using given notation to satisfy the given requirements. In this case, the subjects are asked to appropriately capture the given requirements. This means the students need to choose concepts to satisfy the given requirements and to correctly extract the relations between objects and services.

In this paper, we aim to propose criteria for evaluating conceptual modeling errors made by university freshmen. To achieve this research goal, we quantitatively analyzed class diagrams made by students. During this analysis, we asked ourselves “What kind of criteria are suitable for novice learners when they create conceptual models? ” and “Are there any differences between the scores of novice learners with and without programming knowledge? ” There are two differences between previous research and our research. The first difference is our subjects. We focus on university freshmen and pre-university students, who have not taken any kind of CS specific courses. The other difference is the empirical and continual research method. We have been engaged in this research since 2007 [Hara 2007, Masumoto 2012].

## 2. RESEARCH METHODS

We define conceptual modeling as a way of thinking to solve problems using engineering methodology. In the first semester for university freshmen, students have to take a conceptual modeling course to express services and objects in the real world using a class diagram. The modeling targets in this course are not limited to artificial systems or services which can be executed on computers. The reason for this rule is to have our novice learners concentrate on modeling without programming.

Figure 1 shows an example class diagram used in this research. A hotel is modeled based on the simplified class diagram notation. We use a class diagram which is a simplified standard class diagram defined using UML2.x. Our simple class diagram has the minimum essential elements for conceptual modeling. The minimum essential elements are classes and bi-directional associations. For each class, a name and some attributes are listed, while no attribute types, method names, arguments, return types or visibilities are used. For each association, two names and two multiplicities with four types (0..1, 1, 1..*, *) are used, while no role, inheritance, aggregation, composition or dependency are used. The only association used in this diagram is a simple association between two peer classes. This association represents a pure structural relationship between two peers.

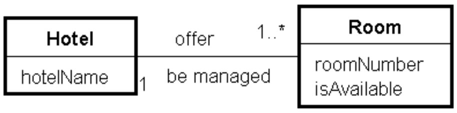


Figure 1. A class diagram with simplified notation

Our subjects were 174 university students who were novices at conceptual modeling. These subjects were divided into two groups based on their computer science knowledge. Table 1 shows their profiles. All subjects were required to answer the questions individually. They were not allowed to discuss the questions with each other or to solve the problems in groups.

Table 1. Subject Group Profiles

| **Group Name** | **11T** | **12T** |
|---|---|---|
| ***Number of subjects*** | 86 | 88 |
| ***Year*** | Sophomore | Freshmen |
| ***Pre-acquired knowledge about computer science (number of semesters)*** | Algorithmic thinking (0.5)<br>Conceptual Modeling (0.5)<br>Algorithm & data structure (1)<br>C language programming (2) | None |
| ***Experiment period*** | Second semestar in Second year | First semestat in first year |
| ***Course type*** | Elective | Required |

Our experiment expands on this learning method by using three tests: a model reading test, a model creation test and a model modification test. In the model reading test, students are given requirements and a diagram. The students answer questions related to the diagram to check their understanding of conceptual modeling. By doing this test, we can confirm the students' fundamental knowledge about conceptual modeling. In the model creation test, students are asked to draw a complete simple class diagram that satisfies given requirements to check their understanding of and appropriate abstraction. In the model modification test, students have to indicate some inadequate elements in the given samples. They are also required to draw a revised model to check their understanding of conceptual modeling, requirement analysis and appropriate abstraction.

Before these tests, the instructor asked his students the names of the essential elements in a simple class diagram to confirm their level of understanding. Two instructors were engaged in the course management. They planned the learning contents of this course and gave our subjects lectures. Then, we analyzed the subjects' answers and discussed the results.

## 3. EXPERIMENTAL RESULTS

### 3.1 Model Reading Test

The goal of this test is to check the conceptual modeling capability of students. In this test, students point out the differences between a given diagram and the problem statements. Four problems (P) were given to students.

The trend of the percentage of problems answered correctly is the same for both groups. The students' level of understanding decreases as follows: class > association > multiplicity > attribute. The average total scores and in particular the attribute related problem scores of these two groups show a significant difference. So, the level of understanding about "attribute" is much lower than for other elements (class, association, multiplicity) for both groups.

## 3.2 Model Creation Test

The goal of this test is to check the requirement analysis capability and the appropriate abstraction capability.

Figure 3 shows the percentage of students who drew the class diagram correctly and incorrectly in this test for the two groups. The 12T group has a higher score than the 11T group. The average total scores and variances of these two groups show a significant difference.

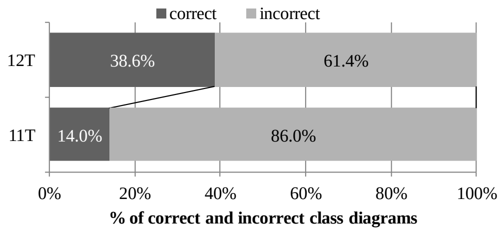


Figure 2. Percentage of students who drew the class diagram correctly and incorrectly in the model creation test for the two groups

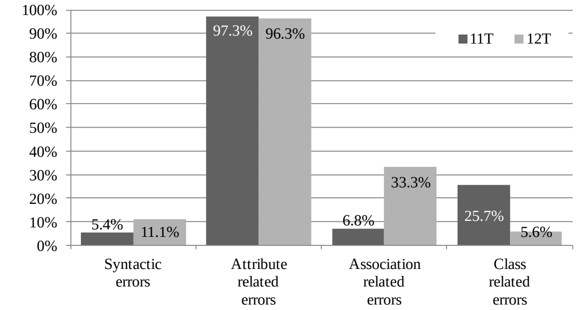


Figure 3. Percentage of the four error types that occurred in the model creation test

At first, we categorized the answers which had some errors based on the three error types from previous research. In total, we extracted four types of errors: syntactic errors, attribute related errors, association related errors and class related errors. Figure 3 shows the percentage of the four error types that occurred in the model creation test. For the 11T students, the number of incorrect answers was 74. For the 12T students, the number of incorrect answers was 54.

In this experiment, we found that attribute related errors are the most common type of error made by novice learners. In both groups, over 95% of the incorrect answers had this type of error. For the 11T group, which has programming knowledge, the percentage of class related errors is relatively higher. On the other hand, 12T group, which has no programming knowledge, shows a higher percentage of association related errors.

We analyzed these 4 types of errors in more detail. The class related error has two detailed subcategories:

1. There are some classes which have different abstraction levels in one diagram.
2 There are more than two classes whose names or attributes have the same meaning in one diagram.

The attribute related errors had six detailed error categories:

1. A class does not have any attributes (No attribute). This error also is included the syntactic error.
2. Two or more classes have the same set of attributes (Same attribute). "Same" means that each attribute has the same range of values.
3. An attribute is defined as not "name" but "value" (Value attribute).
4. Attributes which are actions or methods are listed (Behavioral attribute).
5. The meaning of both an attribute and the multiplicity of an association is overlapped (Overlapped property).
6. Duplicated attributes are used (Duplicated attribute).

The association related error type includes some class diagrams which have no association name or multiplicity and have inadequate association name or multiplicity. This type has four detailed error categories.

1. There are no association names. This error also is included the syntactic error.
2. Inadequate association name is given.
3. There are not two multiplicities for one association. This error also is included the syntactic error.
4. Inadequate multiplicity is given.

## 3.3 Model Modification Test

The goal of this test is to check the ability of conceptual modeling, requirement analysis and appropriate abstraction. In all five problems, students need to point out the mistakes in each class diagram and describe why they are incorrect. Then, they are asked to modify the class diagram to correct the mistakes. P1 has association related errors, which are inadequate multiplicity and duplicate association names. P2 has an attribute related error, where the attribute name is defined as a value instead of a property. P3 has an association related error, which is inadequate multiplicity. P4 has an association related error, which is the lack of association names. P5 has a syntactic error, which is redundant multiplicity.

The trend of the percentage of questions answered correctly is the same for both groups. The highest percentage of correctly corrected errors was for the syntactic error (P1) and the association related error (P4). The lowest percentage was for the association related error (P3). Their level of understanding decreases as follows: P1 > P4 > P2 > P5 > P3. Both P1 and P4 are lacking necessary elements in the diagram. P2, P5 and P3 have inadequate elements in the given diagrams. This means that the "inadequate description" error is more difficult to modify than the "lack of necessary element" error. The average total scores and variances of these two groups are statistically the same. Only the P3 scores of these two groups show a significant difference.

# 4. DISCUSSION

Here we want to discuss our research questions.

## 4.1 Question 1

What kinds of criteria are suitable for novice learners when they create conceptual models with simple class diagrams?

We propose 12 criteria, which are divided into 4 types, for evaluating simple class diagrams made by novices for conceptual modeling based on the results we mentioned above. Table 2 shows the proposed criteria. The frequency of occurrence is different for each item. However, by using these items we can check the level of understanding for conceptual modeling of novice learners. Therefore, conceptual modeling instructors can develop their course for novices with these criteria. An example of each criterion is shown in Figure A1 in the Appendix. These examples were made based on our students' answers. To explain each criterion exactly in these Figures, the authors simplified the students' answers.

Table 2. Criteria for Evaluating Simple Class Diagram made by Novice for Conceptual Modeling

| Error types | Criteria |
|---|---|
| *Syntactic error* | Inadequate notation. |
| | Lack of necessary element [association and multiplicity]. |
| | Lack of name of necessary element [class, attribute and association]. |
| *Class related error* | There are some classes which have different abstraction levels in one diagram. |
| | There are two or more classes whose names or attributes have the same meaning in one diagram. |
| *Attribute related error* | The same attributes are used in more than two classes. |
| | The attribute name is defined as a value instead of a property. |
| | An attribute that describes an action is used. |
| | An attribute that has the meaning of multiplicity is used. |
| | Duplicated attributes are used. |
| *Association related error* | Inadequate association name. |
| | Inadequate multiplicity. |

## 4.2 Question 2

Are there any differences between the programming-known group and the not-known group in terms of their level of understanding of conceptual modeling?

In the model reading test, there are no significant differences between the 11T group and the 12T group in terms of the percentage of questions answered correctly. Especially, the trend of the percentage of problems answered correctly is statistically the same for both groups. The average of the percentage of questions answered correctly by the 12T group is higher than the average of the 11T group. However, there are no statistically significant differences between any scores.

In the model creation test, there is a statistically significant difference between scores of the 11T group and the 12T group. Firstly, The percentage of problems answered correctly by the 12T group which has no programming knowledge is higher than the 11T group which has programming knowledge. Though these two groups were given the same contents and the same length of lectures about conceptual modeling, the 12T group which has no programming knowledge showed a higher score in creating models based on the given requirements. Secondly, about the percentage of the four error types that occurred in this test, for the 11T group the percentage of class related errors is high. On the other hand, the 12T group shows a high percentage of association related errors. Whereas the association related errors are relatively superficial mistakes, the class related errors are quite essential mistakes in conceptual modeling using class diagrams. These types of errors are concerned with abstraction level control. This fact means that programming knowledge has no effect on the ability to control abstraction levels. And finally, about the percentage of attribute related error types in this test, for the 11T group, the percentage of no attribute errors and behavioral attribute errors is about 20%. Duplicated attribute errors occurred only in the 12T group. The behavioral attribute errors occurred only in the 11T group. We think this fact is caused by structured programming knowledge which includes functions. If they draw a class diagram with methods, students in the 11T group would get a higher score on this test. Therefore, it is better to teach conceptual modeling with this notation before programming. However, the total trend of our 12 criteria seems to be the same for both groups.

## 5. CONCLUSION

Our research questions are “What kind of criteria are suitable for novice learners when they create conceptual models?” and “Are there any differences between the scores of novice learners with and without programming knowledge?”

In this paper, we propose criteria for evaluating conceptual modeling errors made by novices based on the results of three experiments. Overall, based on our experiments, programming knowledge seems to not directly affect conceptual modeling ability. If so, conceptual modeling education in this notation for university freshmen is reasonable. In this case, the instructors should consider our 12 criteria listed above. The effects of these matters for the proposed conclusions need to be considered in future work. Also, we need to discuss the relation between diagram notation and education timing more carefully.

## REFERENCES

Bezivin, J. et al, 2009. “Teaching Modeling: Why, When, What?”, *Proc. of the MODELS 2009*, pp.55-62.

Börstler, J., Michiels, I. and Fjuk, A., 2004. “ECOOP 2004 Workshop Report: Eighth Workshop on Pedagogies and Tools for the Teaching and Learning Object-Oriented Concepts”, *ECOOP 2004 Workshop Reader*, pp.36-48.

Fuller, R.M., Murthy, U. and Schafer, B.A., 2010. “The effects of data model representation method on task performance”, *Information & Management*, Vol.47, No.4, pp.208-218.

Hara, Y. et al, 2007. “A Basic Study of an Educational Environment for Modeling and Abstraction”, *Proc. of the ESS 2011*, pp.16-1-16-5, 2011.

Kramer, J., 2007. “Is Abstraction The Key To Computing?”, *In CACM*, Vol.50, No.4, pp.37-4.

Masumoto, K. et al, 2012. “A Basic Study of Teaching Methods for Model-Based Design with Error Analyses of Beginners - in the case of the class diagrams–”, *Proc. of the ESS 2012*, pp.101-109.

Moisan, S. and Rigault, J.P., 2010. “Teaching Object-Oriented Modeling and UML to Various Audiences”, *Proc. of the Educators' Symposium at MODELS 2009*, pp.40-54.

Sendall, S., 2003. “Model Transformation: The Heart and Soul of Model Driven Software Development”, *In IEEE SOFTWARE*, Vol.20, No.5, pp.42-45.