

Quantitative Conceptual Model Analysis for Evaluating Simple Class Diagrams made by Novices

Mizue Kayama, Shinpei Ogata, David K. Asano & Masami Hashimoto

Shinshu University

{kayama, ogata, david, hasimoto}@cs.shinshu-u.ac.jp

Abstract. In this paper, we aim to propose criteria for evaluating conceptual modeling errors made by university freshmen. 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.

1 Introduction

These days, educational methods or learning courses related to conceptual modeling have been explored in many educational institutes, academic conferences and academic journals [1-3]. The learners who were subjects of previous research were mainly third or fourth-year undergraduate students, graduate students and/or young engineers in computer science (CS). They already had finished many specialized classes related to programming, object-oriented analysis and design, databases and so on. Especially, almost all of the students were in a CS program as their major [4,5]. A few studies whose subjects had little prerequisite knowledge, for example, high school students (pre-university), undergraduate students in non-CS programs and CS freshmen in their first semester, also reported their teaching experience or teaching methods related to modeling education [6-8]. However, quantitative evaluation results were not shown in their reports. Especially for students in pre-university, Non-CS and CS freshmen, there are no quantitative studies about model based thinking and subsequent curricula of conceptual modeling.

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 2010 [9,10].

2 Research Methods

2.1 Overview

We define conceptual modeling as a way of thinking to solve problems using engineering methodology. The learning objectives of this course are to develop 3 types of capabilities [11]:

1. Conceptual modeling: The ability to sketch a model diagram correctly according to a certain notation.
2. Requirements analysis: The ability to create the diagram so as to satisfy the requirements represented as sentences.
3. Appropriate abstraction: The ability to avoid defining unnecessary or inadequate classes and attributes for a target domain.

Capability 1 is concerned with the ability to form concepts for designing visual models. If learners lack this ability, they cannot read the given models correctly based on requirements or cannot appropriately detect the differences between the given model and the requirements. Capability 2 is related to the ability to capture the essentials of software requirements. If learners lack this ability, they cannot create suitable models for the requirements. Capability 3 is the same as the ability to abstract fundamental features and/or significant entities from an object or service. If learners lack this ability, they cannot control the abstraction level in model reading and creation.

2.2 Simple Class Diagram

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. 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. Both classes are conceptually at the same level, neither being more important than the other.

In general modeling using object oriented methodology, classes in different levels are used in one diagram. However, novice learners tend not to control abstraction level appropriately. They often assign a system name to a class name or the name of a concrete value to an attribute name. Therefore, we only used a subset of the notation of class diagrams from the original UML2.x notation.

2.3 Subjects

Our subjects were 174 university students who were novices at conceptual modeling. They were divided into two groups based on their computer science knowledge. The members of the 11T group were 86 sophomores. They already had some com-

puter science knowledge. Our experiment was held in their second semester in second year during one of their elective courses. On the other hand, the subjects in the 12T group were 88 freshmen. They had not taken any CS related courses. Our experiment for this group was held in their first semester in first year during one of their required courses.

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.

2.4 Experimental Procedure

When humans acquire a new notation or concept, the typical first step is to read or observe some appropriate samples. By doing this, some features of the notation or concept can be captured. In the next step, the given notation or concept can be used to draw or describe some product. Our experiment expands on this method by using three tests: a model reading test, a model creation test and a model modification test.

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 (P) statements. This test includes four problems. Each problem is related to classes, attributes, associations and multiplicities. Among the choices, some statements are not true for the given class diagram.

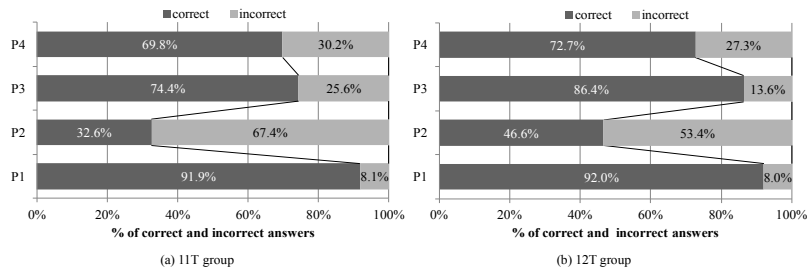


Fig. 1. % of problems answered correctly and incorrectly in the model reading test

Figure 1 shows the percentage of problems answered correctly and incorrectly in this test for the two groups. 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. The 12T group (38.6%) has a higher score than the 11T group (14.0%). However, the average total scores and variances of these two groups show a significant difference. At first, we categorized the answers which had some errors based on the three error types from previous research. As a result, we found a new error type: class related error. In total, we extracted four types of errors: syntactic errors, attribute related errors, association related errors and class related errors. Figure 2 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.

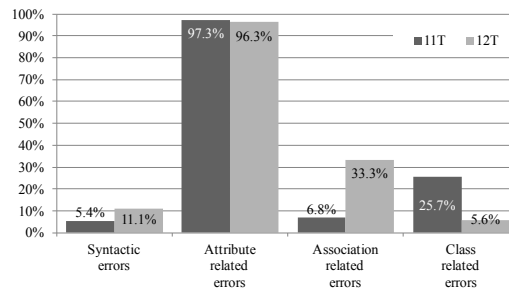


Fig. 2. % of the four error types that occurred in the model creation test.

In this experiment, we found that attribute related errors are the most common type of error made. 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 three in four types of errors in more detail.

The class related error has two detailed subcategories:

- There are some classes which have different abstraction levels in one diagram.
- 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:

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

(f) 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.

- (a) There are no association names. This error also is included the syntactic error.
- (b) Inadequate association name is given.
- (c) There are not two multiplicities for one association. This error also is included the syntactic error.
- (d) Inadequate multiplicity is given.

Figure 3 shows the percentage of attribute related error types in the model creation test. For the 11T students, the number of students who made attribute related errors was 72. For the 12T students, the number of students who made attribute related errors was 52.

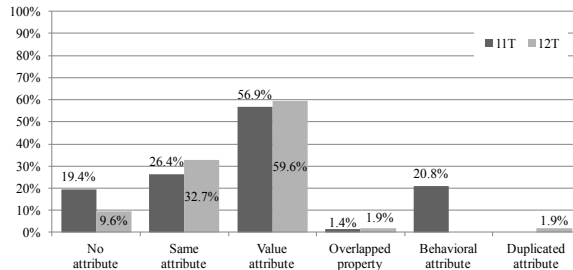


Fig. 3. % of attribute related error type in the model creation test

In the 6 subcategories of the attribute related error, the “Value attribute” errors and the “Same attribute” errors have relatively high occurrence percentages for both groups. For the 11T, which has programming knowledge, the percentage of the “No attribute” errors and the “Behavioral attribute” errors is about 20%. The “Duplicated attribute” errors occurred only in the 12T, which has no programming knowledge.

3.3 Model Modification Test

Overview. 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.

Results. Figure 4 shows the percentage of questions answered correctly and incorrectly in the model modification test for the two groups. 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.

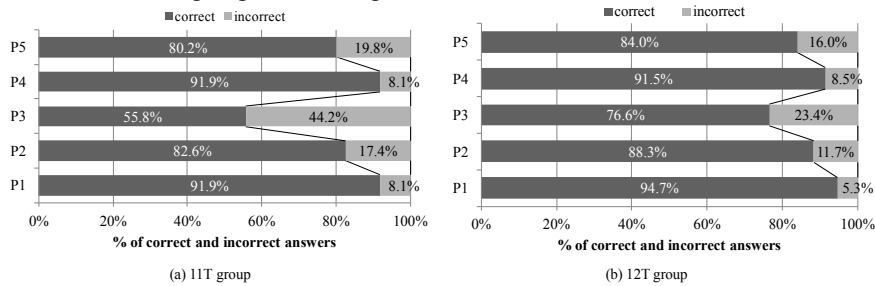


Fig. 4. % of questions answered correctly and incorrectly in the model modification test.

4 Discussion

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 1 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.

Table 1. Criteria for evaluating simple class diagram made by novice for conceptual modeling

Error types	Criteria
<i>Syntactic error</i>	Inadequate notation.
	Lack of necessary element [association and multiplicity].
	Lack of name of necessary element [class, attribute and association].
<i>Class related error</i>	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.
<i>Attribute related error</i>	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.
<i>Association related error</i>	Duplicated attributes are used.
	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.

- (a) 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.
- (b) 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.
- (c) 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.

About (a), 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. About (b), 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. About (c), 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 13 criteria seems to be the same for both groups.

The results of the model modification test are the same as 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. 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.

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. We found that there is no relation between programming knowledge and conceptual modeling ability for the notation used in our experiments. We used real world objects in our models, not abstract objects in this study. Moreover, we asked students to solve each problem individually, without discussion with other students.

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.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number 22300286 & 16H03074.

References

1. J.Bezivin, et al, “Teaching Modeling: Why, When, What?”, MODELS 2009, 55-62 (2009)
2. J.Börstler, I.Michiels, and A.Fjuk, “ECOOP 2004 Workshop Report: Eighth Workshop on Pedagogies and Tools for the Teaching and Learning Object-Oriented Concepts”, ECOOP 2004 Workshop Reader, 36-48 (2004)
3. J.Kramer,J. “Is Abstraction The Key To Computing?”, CACM, 50(4), 37-42 (2007)
4. N.Bolloju and F.S.K.Leung, “Assisting Novice Analysts in Developing Quality Conceptual Models with UML”, CACM, 49(7), 108-112 (2006)
5. H.C.Cham, H.H.Teo and X.H.Zeng, “An evaluation of Novice End-User Computing Performance : Data Modeling, Query Writing and Comprehension”, J. of the American Society for Information Science and Technology, 56(8), 843-853 (2005)
6. J.Niere and C.Schulte, “Avoiding anecdotal evidence : An experience report about evaluating an object-oriented modeling course”, MODELS 2005 Educator's Symposium, 63-70 (2005)
7. C.Schulte and J.Niere, “Thinking in Object Structures: Teaching Modeling in Secondary Schools”, 6th ECOOP Workshop on PTLOOC (2002)
8. S.Akayama et al, “Development of a Modeling Education Program for Novices using Model-Driven Development”, Workshop on Embedded and Cyber-Physical Systems Education (2012).
9. Y.Hara. et al, “A Basic Study of an Educational Environment for Modeling and Abstraction”, ESS 2011, pp.16-1-16-5, 2011.
10. M.kayama et al, “A Practical Conceptual Modeling Teaching Method Based on Quantitative Error Analyses for Novices Learning to Create Error-Free Simple Class Diagrams”, 3rd ICAAI, 616 - 622 (2014)
11. S.J.Merror and M.C.Balcer, “Executable UML: A Foundation for Model Driven Architecture”, Addison-Wesley Professional (2002)