Novice Java Programmers' Conceptions of "Object" and "Class", and Variation Theory

Anna Eckerdal
Department of Information Technology
Uppsala University
P.O. Box 337, 751 05 Uppsala, Sweden
Anna.Eckerdal@it.uu.se

Michael Thuné
Department of Information Technology
Uppsala University
P.O. Box 337, 751 05 Uppsala, Sweden
Michael.Thune@it.uu.se

ABSTRACT

Problems with understanding concepts, so called misconceptions, have been investigated and reported in a number of studies regarding object-oriented programming [4], [3]. In a first programming course using an object-oriented language, it is of great importance that students get a good understanding of central concepts like *object* and *class* at an early stage of their education. We have, with a phenomenographic research approach, performed a study with first year university students, investigating what an understanding of the concepts *object* and *class* includes from a student perspective. By applying variation theory [8] to our results we are able to pin-point what the students need to be able to discern in order to gain a "rich" understanding of these concepts.

Categories and Subject Descriptors

K.3.2 [**COMPUTERS AND EDUCATION**]: Computer and Information Science Education—*Computer science education*; D.1.5 [**PROGRAMMING TECHNIQUES**]: Object-oriented Programming—*Java*
; D.3.3 [**PROGRAMMING LANGUAGES**]: Language Constructs and Features—*Cla-sses and objects*

General Terms

Human Factors, Theory

Keywords

Conceptions, misconceptions, phenomenography, variation theory

1. INTRODUCTION

Java is an often used first programming language in introductory programming courses for university students. There are many reports on problems with teaching Java [6], pointing out difficulties to understand central concepts in the object-oriented paradigm [3]. The study reported in this paper has a focus on students' different understandings of some central concepts in object-oriented programming. We identify different understandings of the concepts expressed by the students in the group. These understandings are critical from the students' perspective [10]. They also cover most of an expert understanding [9]. We claim that it is possible to establish general guidelines on how to organize the teaching and learning environment in such a way that students can get a good understanding of the concepts in question, and thus avoid misconceptions. We first give a theoretical background for the study and the analysis performed. The study and the results will then be presented, and after that we discuss implications for teaching following from the results from the study. The general implications for teaching are well in line with the results from other studies, and give a theoretical basis for explaining these results and how to generalize them.

ITiCSE'05, June 27–29, 2005, Monte de Caparica, Portugal.

2. PREVIOUS RESEARCH

There are many studies on *mis*conceptions of object-oriented concepts. The studies by Fleury [3] and Holland, Griffiths, and Woodman [4] are good example of this line of research. Fleury performed a study on student-constructed rules in beginning programming courses, where Java was taught, pointing out misconceptions among students. Holland, Griffiths and Woodman reported on misconceptions observed among students in a distance course where Smalltalk was taught. Our study differs from these, since it focuses on students' *conceptions* rather than misconceptions. We have found very few studies of this kind in the literature on how students learn to program. Fleury's constructivistically based study of students' understandings of objet-oriented programming [2], is an example. Another is Booth's phenomenographical investigation of how students' experience functional programming [1]. Our study, as Fleury's, has its focus on object-oriented concepts. However, like Booth, we use a phenomenographic approach (see below). The rationale for considering conceptions instead of misconceptions is the following. The conceptions found among students typically correspond to different aspects of a correct understanding of the concepts of interest. Since these are the conceptions actually formed by the students, they reveal ways to understand the concepts, that are of decisive importance from the students' perspective. We argue that focusing on these crucial aspects in the teaching can help the students to gain a good understanding, and thus avoid different kinds of misconceptions.

3. RESEARCH APPROACH

3.1 Phenomenography

Variation theory is a tool to support and give guidelines in different educational settings. Variation theory has developed from Phenomenography [8]. Phenomenography aims at describing the variation of understandings of a certain phenomenon found in a group of people. Marton and Booth discuss the idea of phenomenography:

> The unit of phenomenographic research is *a way of experiencing something,* [...], and the object of the research is the *variation* in ways of experiencing phenomena. At the root of phenomenography lies an interest in describing the phenomena in the world as other see them, and in revealing and describing the variation therein, especially in an educational context [...]. This implies an interest in the variation and change in capabilities for experiencing the world, or rather in capabilities for experiencing particular phenomena in the world in certain ways. These capabilities can, as a rule, be hierarchically ordered. Some capabilities can, from a point of view adopted in each case, be seen as more advanced, more complex, or more powerful than other capabilities. Differences between them are educationally critical differences, and changes between them we consider to be the most important kind of learning. [8, p. 111]

And later:

> [...] the *variation* in ways people experience phenomena in their world is a prime interest for phenomenographic studies, and phenomenographers aim to describe that variation. They seek the totality of ways in which people experience, or are capable of experiencing, the object of interest and interpret it in terms of distinctly different categories that capture the essence of the variation, a set of categories of description [...] [8, p. 121-122]

The object of interest in a phenomenographic study is thus how a certain phenomenon is experienced by a certain group of people. A fundamental assumption in phenomenography is that there exists only a limited number of qualitatively different ways in which a certain phenomenon can be understood.

Phenomenography is an empirical, qualitative research approach. It is often used in educational settings. Data can, like in the present study, be gathered in the form of interviews. The interviews are transcribed and analysed. Researchers, often more than one, analyse the data in order to find qualitatively different ways to understand the phenomenon expressed in the data. The researcher formulates the essence of the understandings found with his or her own words as categories of description. It is important to state that the analysis is on a group level, not aiming at presenting individual students' understandings, but the different understandings found in the group. This is done by reading and rereading the interviews, in context, but also by decontextualising excerpts and comparing them and grouping them together in different categories of understandings. The resulting description of qualitatively different categories of understanding constitutes the *outcome space* of the phenomenographic analysis.

3.2 Variation Theory

According to the phenomenographic tradition, the learning process is a question of *discerning* new aspects of phenomena. A specific aspect cannot however be discerned without experiencing *variation* in a "dimension" corresponding to that aspect. These dimensions are characteristic for the specific aspects, and the variations make central features of these aspects visible [10, p. 146].

With the phenomenographic outcome space as the starting point, the results can be applied in education, by using variation theory. When the empirical data is studied it is possible to discern the focus of each understanding expressed in the categories of description. In the phenomenographic analysis we identify aspects of the understanding of the phenomenon, critical for the understanding from the students' perspective. Learning requires discernment of new aspects of the phenomenon, and the teacher can create the conditions for such discernment with the judicious use of variation. By varying examples and problems and holding the critical aspect of the phenomenon invariant, that critical aspect is lifted out of the surrounding "noise". We speak of opening a dimension of variation, in which taken-for-granted ways of understanding are now brought into focus. Identifying these dimensions of variation corresponding to the critical aspects, gives a basis for finding implications for teaching.

4. THE STUDY

4.1 The Interviews

A study has been performed where 14 first year university students were interviewed on their understandings of the concepts *object* and *class*. The students had just finished their first programming course, a compulsory course giving 4 credit points. (At Swedish universities one credit point represents one week's full-time study.) The programming language used in the course was Java.

The interviews were semi-structured [7] with the aim to encourage the students to demonstrate as much as possible of their understandings and experiences within the theme of the interview. The interviewer had prepared a small number of questions, intended to approach the phenomena of object and class in different ways, to give the opportunity for the students to express as much of their understanding as possible.

Let us emphasize once again that our study is qualitative. Thus we are not aiming for statistically significant results. The objective in selecting persons to interview was to get as broad a coverage as possible of different conceptions. For that reason, most students taking part in the course filled in a questionnaire about previous programming knowledge, education, work experiences and gender. On the basis of these answers, we selected interviewees that represented as broad a coverage as possible of the factors mentioned.

4.2 The Phenomenographic Analysis

The interviews were transcribed and analysed. Two researchers independently read and analysed the interviews, looking for qualitatively different ways to understand the concepts *object* and *class* expressed in the data. Our results were very similar. We agreed upon three different ways to understand the concepts found in the data.

The different understandings of the concepts *object* and *class* found in the data are presented in Table 1 and Table 2 respectively. The understandings are inclusive. This means that an understanding expressed in one of the latter categories includes the understandings expressed in the former categories. Below, the categories in Table 1 and Table 2 are illustrated by excerpts from interviews. In the quotes, the interviewer is labeled I, and the students A, B, C etc.

4.2.1 The Concept of "object"

The different comprehensions of the concept *object* found in this study, can be formulated in three categories of description presented in Table 1.

Object is experienced as a piece of code.
As above, and in addition object is experienced as something that is active in the program.
As above, and in addition object is experienced as a model of some real world phenomenon.

Table 1: Categories describing the different ways to understand the phenomenon *object* found in the group.

In the first category, the understanding of the concept is limited to focus on the code as text. Student C says about objects:

> I imagine that it is a piece of code with all the variables piled under

When the interviewer asks the student how he/she would explain to a friend, who does not know anything about programming, what an object is, student N answers:

> I'd just say that it is a part of the program.

In the second category the comprehension is extended to include the results of the program execution, and the task of the object. It can be illustrated by the following answers.
Student H says:

> the object is a kind of, what is doing something [..] because it is all about that something is going to happen.

Student J says:

> If you think of the Java program, that it is built of different objects and it is the objects we modify so that we can get what we want from it.

The third category describes an understanding that an object is a model of some real world phenomenon. This is expressed in the following quotes:

> C: Yes an object, you can have a rather physical image of it....
> I: What did you say, physical?
> C: Kind of, you can think of a car and then it has one variable for how many wheels it has, one variable for the size of the engine like that.

The three categories express an increasing complexity. The first category shows an understanding that all students express in one way or the other, objects as they appear in the code. A few students express only this understanding. This category expresses a poor understanding, while the last one shows a rich understanding including fundamental ideas behind the object-oriented paradigm.

4.2.2 The Concept of "class"

When looking for the different understandings of the concept *class* expressed in the study, a pattern similar to the understandings of the concept *object* is found. There are comprehensions focusing on the code and the task of the programmer, but there are also comprehensions where the reality the program is supposed to model is present. The categories of description are presented in Table 2 and illustrated by quotes below.

Class is experienced as an entity in the program, contributing to the structure of the code.
As above, and in addition class is experienced as a description of properties and behaviour of the object.
As above, and in addition class is experienced as a description of properties and behaviour of the object, as a model of some real world phenomenon.

Table 2: Categories describing the different ways to understand the phenomenon *class* found in the group.

Many of the students express an understanding belonging to the first category. Student H says:

> A class is, well I figure a class is like a small program, that's how I think of it, a small program inside the whole big program, if you say that the big program is the main program, then the class is like a small program doing certain things.

The understanding has its focus on the program structure and the programmers task and describes the class-concept as a help for the programmer when structuring the code. It deals with the code and the programming task, and the description of the class reminds of a description of modules, even if no student explicitly uses this formulation. Some students emphasize this module aspect:

> C: Then the class should be something reasonable, containing what you detach [...] But the class I suppose, is only a diffuse collection of, what I belive belong together in some way.

The second category is the most common understanding expressed in the group. Even if none of the students explicitly uses the expression "abstract data type", the descriptions point in this direction.
Student O says:

> Eh, when you write a class [...], you write what you want the objects to look like, and that's how I understand a class, that you are able to create an object and something about what you want to do with this object in the different methods [...]

In the third category in Table 2, the close relationship between the class definition and the reality the class depicts is pronounced. This category includes the understanding expressed in category two. Only a few students express this kind of understanding.

> I: I mentioned class. How do you understand classes?
> C: It's a bit more diffuse actually. Class, it is I can imagine that a class contains, can contain a number of objects or only one object and different operations you can do in an object or between objects. So you

can also imagine what it would represent in the reality.
I: Okay.
C: Yes well, you can think of a workplace and a person working there, then you have two objects and then they can kind of interplay with each other through different operations sort of, what do I know. The person gets coffee and then the coffee variable decreases in the workplace like that.

4.3 Dimensions of Variation

The phenomenographic analysis of the data has revealed understandings found among the students. These are critical understandings from a student's perspective. Each aspect of the concept, expressed as categories of description in Table 1 and Table 2, require focal awareness of a specific dimension of relevance for the understanding.

There is a close relationship between the concepts *object* and *class*, and Table 1 and Table 2 show similar patterns for the understandings of the concepts. In the empirical data collected for the present study, most students express understandings of the concepts in corresponding categories. If a student for example expresses an understanding of *object* corresponding to the second category in Table 1, he or she also expresses an understanding of *class* corresponding to the second category in Table 2. There are few, if any examples where students show an advanced understanding of one concept, and a poor understanding of the other concept. The understandings found in the three categories in the two tables, will now be grouped together and discussed in terms of focal awareness found in the empirical data. The focal awareness of the understandings are then analysed in order to find dimensions of variations necessary for discernment of these aspects of the concepts. In this way we have identified the variation necessary for learning to take place.

In the first categories in Table 1 and Table 2 the students have experienced class as 'an entity of the program, contributing to the structure of the code', and object as 'a piece of code'. The focal awareness of this understanding of a class is the appearance of the structure of the program text. The focal awareness of the understanding of objects is on the program text. To be able to focus on this aspect, students need to discern that in different programs objects and classes appear in different ways. In that sense, the textual representation of programs constitutes a dimension of relevance for the understanding of *object* and *class*. Different, specific program texts constitute values along this dimension.

In the second category, class is experienced as 'a description of properties and behaviour of the object', where object is understood as 'something that is active in the program'. The focal awareness in these categories is on what happens during execution of the program, in particular on the objects created and how they contribute to different events at run-time. The objects are the active parts of the program, accomplishing the task given. To be able to discern the understanding expressed in the second categories, the students need to focus on the objects the program creates and events happening at execution of the program. Here, the relation between class description, object action, and resulting events during the execution of the program constitutes a dimension. Different specific cases of such relations provide values along this dimension. The variation between these values can enhance an awareness of object and class corresponding to the second category of understanding, according to Table 1 and Table 2.

In the last categories in Table 1 and Table 2, class is experienced as 'a description of properties and behaviour of the object', where object is understood as 'a model of some real world phenomenon'. The focal awareness is still on the class' description of the active objects, but now with an emphasis on the reality aspect of the class description. In this case, the relation between class, object and real-life phenomena constitute a dimension. Different specific cases of such relations constitute values along this dimension.

5. IMPLICATIONS FOR EDUCATION

Our results can shed new light upon and give explanation to other research and discussions in the field. The following paragraphs show some examples of this. Holland, Griffiths and Woodman list some misconceptions noticed at distance courses where Smalltalk was taught, in one introductory undergraduate course, and one postgraduate course [4]. One misconception mentioned is "object as a kind of variable". Students with previous experience of procedural programming may, if the examples they first come across have only one instance variable, develop the misconception that objects are in some sense mere wrappers for variables. It is trivially easy to avoid this misconception by ensuring that all the classes showed as an introduction, have more than one instance variable. Another misconception that can appear is if the data aspect of objects is overemphasized at the expense of the behavioural aspect. This misconception can be avoided by using introductory object examples where the response to a message is substantially altered depending on the state of the object. Both the misconception "object as a kind of variable" and the overemphasizing of the object's data aspect is an indication of the importance to attain a conception according to the second categories in Table 1 and Table 2. The second category in Table 1 emphasizes that objects are active during execution of the program. This points to the behavioral aspect of objects. The second category in Table 2 explains classes as a description of both data about the object, and methods explaining the behaviour of the object. As explained in section 4.3, the relation between class description, object action, and resulting events during program execution constitutes a dimension where variation is needed. This implies, e.g., variation in values of several instance variables, caused by several method calls. This is according to the recommendations from Holland et al.

A common problem among novice programmers, also mentioned by Holland et al, is to understand the difference between *class* and *object*. This is obviously a problem if several examples are presented in which only a single instance of each class is used. To avoid this, good practice is always to work with several instances of each class. As explained in section 4.3, the textual representation of programs constitutes a dimension of variation. This implies variation in the sense of presenting more than one instance of the class in the code, as recommended by Holland et al.

In the light of the present study, the recommendations from Holland et al can be summarized as: *variation in dimensions corresponding to critical aspects of the understanding is of great importance*. These dimensions of variation are not only pinpointed here, but also explained in the theory of phenomenography and in the analysis of the data by applying variation theory on the results of the study.

Holmboe [5] performed a study where a few people of different background were asked to describe in their own words what object-oriented programming is. He asked students who had just finished an introductory course on object-oriented programming, senior students tutoring the same course and professors of Computer Science or System Engineering. He made a qualitative analysis of the answers, and comments that some types of knowledge are more suitable as a basis for further knowledge construction than others. When analysing the results from the study he writes about understandings which include the world outside the computer itself: "A person with holistic knowledge relates the implementation and design of a computer program to the real world being simulated." Holmboe emphasizes the importance that "[...] more students will experience the connection between reality, model and implemented program, and thus reach holistic knowledge of object-orientation sooner in their learning process." The third categories in Table 1 and Table 2 capture an understanding of classes and objects that includes the world outside the computer itself. The dimensions of variation found and discussed in section 4.3 are valuable as a basis for teachers to facilitate for the students to reach this understanding.

In Fleury's study on students' constructed rules [3], she stresses, with a reference to Holland, Griffiths and Woodman [4], the importance of carefully constructed sample programs to avoid misconceptions of concepts. Our study stresses the importance of designing the education so that the students can discern the critical aspects of the understanding. Carefully constructed sample programs in this sense means variation of dimensions corresponding to these critical aspects. This is applicable not only on sample programs, but in all different aspects of the learning environment.

6. CONCLUSIONS

For the Java educator, one challenge is to construct an educational environment which facilitates for students to reach a rich understanding of the concepts *object* and *class*. To this end it is important to know the different ways in which students (as opposed to experts) typically experience these concepts. Our phenomenographic study has given such insight. Next, the educator needs to identify what variation the students have to discern in order to become aware of aspects belonging to a rich understanding of these concepts. Here, variation theory can be a useful tool, as demonstrated in the previous discussion.

By using dimensions of variation, discussed in the previous section, implications for teaching are found. Teaching is here defined in a wide sense. By teaching we mean everything that supplies resources for learning. Examples of such resources could be programming assignments, software tools, lectures, Internet and fellow students, anything the students choose to use in their learning. The whole organisation of the learning environment is in this sense teaching.

A general implication for teaching is to make resources in the learning environment available that help students to discern the aspects mentioned Table 1 and Table 2 and developed in the previous section. These are resources that point out the corresponding dimensions of variation of the aspects.

The results in Table 1 and Table 2 can be implemented in the teaching and learning environment offered to the students, in a number of ways. There is a great freedom and possibility to adapt the results to the preferences of each teacher and student group.

7. REFERENCES

[1] S. A. Booth. *Learning to Program. A phenomenographic perspective.* Number 89 in Göteborg Studies in Educational Science. Acta Universitatis Gothoburgensis, Göteborg, Sweden, 1992.

[2] A. E. Fleury. Student conceptions of object-oriented programming in java. *The Journal of Computing in Small Colleges*, 15(1), November 1999.

[3] A. E. Fleury. Programming in java: Student-constructed rules. In *Proceedings of the thirty-first SIGCSE technical symposium on Computer science education*, 2000.

[4] S. Holland, R. Griffiths, and M. Woodman. Avoiding object misconceptions. In *ACM SIGCSE Bulletin , Proceedings of the twenty-eighth SIGCSE technical symposium on Computer science education, Volume 29 Issue 1*, 1997.

[5] C. A. Holmboe. Cognitive framework for knowledge in informatics: The case of object-orientation. In *Proceedings of the 4th annual SIGCSE/SIGCUE ITiCSE conference on Innovation and technology in computer science education*, 1999.

[6] M. Kölling. The problem of teaching object-oriented programming, part i: Languages. *JOURNAL OF OBJECT-ORIENTED PROGRAMMING*, January 1999.

[7] S. Kvale. *InterViews: An introduction to qualitative research interviewing.* Sage, 1996.

[8] F. Marton and S. Booth. *Learning and Awareness.* Lawrence Erlbaum Ass., Mahwah, NJ, 1997.

[9] B. Meyer. *Object-oriented Software Construction.* International series in Computer Science. Prentice Hall, 1988.

[10] M. F. Pang. Two faces of variation: on continuity in the phenomenographic movement [1]. *Scandinavian Journal of Educational Research*, 47(2), 2003.