Students Learn CS in Different Ways: Insights from an Empirical Study[1]

Anders Berglund
UpCERG, Uppsala Computing Education Research Group
Department of Information Technology
Uppsala University
Uppsala, Sweden
Anders.Berglund@it.uu.se

Mattias Wiggberg
UpCERG, Uppsala Computing Education Research Group
Department of Information Technology
Uppsala University
Uppsala, Sweden
Mattias.Wiggberg@it.uu.se

Abstract

This empirical study demonstrates that students' learning of computer science takes place in qualitatively different ways. The results consist of categories, where each category describes a certain way in which the students approach their learning. The paper demonstrates that some of the ways of tackling learning do better than others in producing a good learning outcome, and that they should therefore be encouraged. The data underlying these results were collected through interviews with third and fourth year students in two countries, and were analysed using a phenomenographic research approach.

Keywords

Computer science education research, the act of learning, phenomenography.

1 Introduction

Student's approaches to their studies partly explain the differences in results between individuals (Marton & Booth, 1997; Marton, Watkins, & Tang, 1997). Let us, for sake of the example, consider a student who is trying to learn formulas and standard solutions by heart. She/he is less likely to develop an advanced understanding of that which she/he tries to master (for example, a network protocol) than someone who is searching for an underlying meaning, and tries to integrate the newly learned concepts into what she/he already knows.

This study analyses the different ways advanced students in computer science approach their studies. The outcome consists of a set of categories that describe the different ways in which the students tackle their learning of computer science concepts.

The study is performed by computer scientists (with a strong practical and theoretical background in educational research) in a real teaching situation. Exactly here, in its origin within computer science, lies the key strength of the current work. Each of the categories, as well as those particular aspects of a category that distinguish it from another, arise from computer science, and are thus described in the terms of the field. It becomes easy to bring the results back into teaching situations, since the results 'talk computer science'.

The value lies on different levels. Firstly, a teacher who is aware of how her/his students approach their studies, and which ways of studying are better than others, is better prepared for her/his teaching. Secondly, the project is intended as a platform for further research into the teaching and learning of computer science. As a consequence of these aims, the paper offers general insights on how CS is learned by our students, but does not offer concrete advice on 'how to teach'.

The following section introduces the theoretical framework used in the study. In section 3 the landscape of related research is outlined. Section 4 describes the setting in which the study took place. The empirical results are presented in section 5, while the last section analyses the conclusions and looks further into new research projects.

2 The pedagogical research framework

For a researcher who aims to explore students' learning, a thorough research approach[2] is crucial for getting a reliable result. An approach offers guidance on both practical (procedural) issues, such as how to collect data or perform the analysis, and on theoretical issues, such as trustworthiness and reliability. The research approach comes to serve as a 'theoretical lens' that makes certain aspects of a research object or situation clearer, while other aspects come to reside in the background. Thus, the

[1] This paper was previously published in the proceedings of the Eleventh Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2006). Reprinted with the permission of ACM.

[2] *Research framework* and *research methodology* are other common terms for what we in this paper call a *research approach.*

researcher has to select an approach that is good for obtaining results of the desired kind, that is appropriate in the setting in which the project takes place, and with which the researcher her/himself is comfortable (Denzin & Lincoln, 1994).

2.1 Phenomenography

In this project, the aim is to describe the different ways in which the students themselves experience how they go about learning. For this purpose, phenomenography (Marton & Booth, 1997) is selected as a research approach. It has previously been successfully used in studies of how university students understand core concepts in computer science (see for example Berglund (2005) for computer network protocols; Booth (1992), Bruce et al. (2004), and Eckerdal and Berglund (2005) for programming; Eckerdal & Thuné (2005) for the concepts of class and object; Lister, Box, Morrison, Tenenberg and Westbrook (2004) for data structures).

Phenomenography is a qualitative, empirically based research approach that aims to interpret, describe, and categorise how a phenomenon (for example a network protocol, or the aims students have for taking a course) is experienced or understood within a group of students. Each of the resulting categories comes to describe a certain way in which the phenomenon under investigation is understood. Taken together, the categories describe the variety of understandings that can be found in a group. Still, they cannot serve to count individuals, since an individual can express positions that fit in many categories. Furthermore, students are selected not to be statistically representative, but to represent various backgrounds, earlier study results, gender, age, etc.

Data for a phenomenographic study is normally collected through interviews, which are later transcribed. Normally the number of interviewees is rather low – ten to twelve are normal numbers – since the aim is to reveal variation in the students understanding, rather than to discuss the properties of a ‘sample’.

During the analysis the researcher takes quotes that illustrates the students’ experience of the phenomenon under investigation, and sorts them into ‘piles’ that represent the emerging categories. When doing so she/he goes back and forth between the interviews and the ‘piles’ in order to try different interpretations. This is a demanding, iterative process, full of failed attempts and incorrect interpretations, before the researcher reaches her/his final outcome.

Since the categories illustrate different aspects of the *same* phenomenon, they are logically related to each other. Were they not, they would describe aspects of different phenomena. In general, some categories offer a wider or richer perspective and often come to embrace others in an inclusive structure. As will be illustrated in the results section of the paper, the more embracing categories are generally more desirable.

There are different means to evaluate the reliability of the results. The logical structure can be examined; the process and the results can be discussed with, and be examined by, other researchers. Furthermore, through the use of a phenomenographic framework, a language and a theory are available that let the researcher compare her/his findings with the insights gained by other researchers. A deeper discussion of phenomenography, its theoretical underpinnings and applications, is given in Marton & Booth (1997), while Berglund (2005) offers details on the current project.

2.2 The act of learning

Learning, as it takes place in real situations, is complex and multi-faceted. In order to offer a framework for analysing learning, phenomenography introduces a distinction between two aspects of learning: (1) the *what* aspect of the learning, describing the content of the learning (for example a network protocol); and (2) the *how* aspect, describing how the students go about learning, or how they tackle their learning. While the first is normally referred to as the *object of learning*, the latter is labelled the *act of learning.* This distinction is, as Marton and Booth (1997) point out, purely analytical: the aspects can only be ‘thought apart’ for research purposes and do not represent different concepts.

It is sometimes tempting to draw parallels between phenomenography and psychologically based theories of learning. However, the differences are important: While psychology discusses what *learning is*, independent of what the learning is about and the context of what is learnt, phenomenography studies the *experience of learning something*, in a particular setting. Thus the phenomenographic concept of the *act of learning* always refers to how students go about learning something specific, such as a particular concept in computer science, mathematics or economics, while psychology has learning in itself as one of its primary objects of research.

3 Related research

Although the results of a phenomenographic study concerning the *act of learning* describe how students go about learning something specific, it *is* possible to draw generalised conclusions on how students go about learning.

Marton, Beaty & Dall’Alba (1993) have discerned six different ways of approaching studies among social science students at the Open University (OU) in the UK, while Marshall, Summers & Wollnough (1999), in their work on learning in an engineering context, presented five categories. The broad picture formed by these two studies are similar, and are confirmed by other researchers (Dahlin & Regni, 1997; Eklund-Myrskog, 1998; Marton, Watkins, & Tang, 1997): while the least advanced categories relate to words, syntax, texts etc. of that which is learnt about, the advanced categories see the big picture and focus on and contextualise the content of the learning. Marshall, Summers, & Woolnough (1999, p. 305) argue that the differences between the studies are fine-tuned and related to the various cultural and educational contexts, which make “different aspects of the learning experience be foregrounded or accentuated in different contexts”. They also express differences between different subject areas and the consequent different kinds of effort that are needed for good learning. These differences thus come to display what is characteristic for learning in a certain subject area from the students' point of view.

It is interesting to contrast these findings with the results of the only two studies on the act of learning computer science that have been made. Booth (1992) and Bruce et al. (2004) have both studied introductory courses, where they revealed fewer, and different, categories compared to the studies mentioned above. Although further research is needed to determine the reasons for these discrepancies, it could be assumed that the different student populations, beginners (Booth, Bruce et al.) and advanced students (other studies), are important. Only students who have a long experience in studying computer science can develop advanced understandings of concepts within the subject.

4 The study

The findings discussed in this paper stem from phenomenographically analysed interviews with seventeen students who took part in an international project-based course in computer systems. They were in their third and fourth years and were aiming for a Masters degree in Computer Science (Berglund, 2005; Daniels, 1999). In the course they worked in cross-national teams of six, with three students located at Uppsala University in Sweden and three at Grand Valley State University in Michigan, USA. In their projects, the students developed a software system to control a motorised toy from a web browser. The task was demanding for the students; to succeed, the team members had to collaborate. They interacted using different ICT-based tools, such as IRC and e-mail. In the year from which the data in this study stems, the students were asked to select a program package created by a student team during the previous year and to further develop this code package.

This project forms the background against which the interview data have been analysed. The project and the project teams were often referred to by the students during the interviews.

5 Empirical results

The results primarily consist of seven categories, each of which describes a way in which students act to learn computer science. They differ in their 'richness': the more advanced categories (with a higher number) describe a more complex or sophisticated way of acting to learn. Table 1 summarises the categories. Each row corresponds to a category, while the columns A – F indicate the aspects that build each of the categories. For example, category 2 only contains one aspect, namely *Learning in computer science as an academic discipline*, while the last category, number 7, contain all aspects.

Some categories are illustrated by interview scripts. Certainly, the full interviews lie behind each of the categories, but space does not allow more than selected examples. The extracts serve to 'flesh out' some categories and to give the reader a 'feel' for what the students say and how we have interpreted their statements.

Cat 1. Learning CS through learning to use application programs

What it means to learn computer science is described in the first category as learning to deploy application programs. The category focuses on the tools that are used, neglecting other aspects of the field. This category differs from the remaining six in that it does not focus on computer science as it is known to and taught by the computer science community. For this reason it comes to stand outside the hierarchical structure.

Cat 2. Learning CS through learning about isolated concepts

The second category expresses learning of computer science as learning about isolated concepts of computer science. The interview excerpt with Staffan[3] illustrates:

Interviewer: Have you learnt any computer science through working with the camera then [...]?

Staffan: Yes you bet, I have found out a bit more about Linux, how you install things, and download new sources and compile them. That sort of thing, it was quite a lot of fun but at the start we had a lot of problems with the computers as well, both Magnus' and Michael's computers crashed a few times so we had to reinstall everything, yeah, otherwise, well yeah it was a lot of stuff like that as we learned the technical stuff.

After this statement, the theme is continued, and Staffan named more topics, such as compiling and installing the operating system Linux. His statements never touched upon what learning meant, or what was done with that which was learnt. Only the catalogue of computer science concepts was offered.

What is new in this category, compared to the first, is that computer science is seen as an academic discipline with a certain content.

Cat 3. Learning CS through consolidating what is already known

In the third category, the computer science concepts are still seen in isolation. Learning is here understood as getting deeper insights, or consolidating what is already known.

As an example from one of the many quotes that underlie this category, we can listen to a few words by Samuel. He argued that "some of the knowledge I had [...] became a lot clearer for me". The underlying meaning is that the insights become clearer by working in this project.

In contrast to what was present in the previous category, the students here 'do' something with what they learn. They 'go deeper' or consolidate their previous knowledge. Thus, since a particular way of learning is pointed out, there are different ways to learn.

[3] The Swedish students are assigned names that start with an "S", while the names of American students start with an "A". Students that are referred to in the interview scripts are assigned names starting with an "M".

	Label	A	B	C	D	E	F
1.	Learning CS through learning to use application programs						
2.	Learning CS through learning about isolated concepts	Learning in computer science as an academic discipline					
3.	Learning CS through consolidating what is already known		There are different ways in which CS concepts can be understood				
4.	Learning CS through analysing systems			Focus on a whole – parts Relationship			
5.	Learning CS through integrating systems				Integrating or trans-forming something		
6.	Learning CS through giving meaning to concepts					Searching for personal meaning	
7.	Learning CS through developing as a professional						Professional development

Table 1. How computer science students act to learn their subject. For each category (labelled 1 - 7), the aspect of which it is constituted can be found in the columns (labelled A – F). The table is further discussed in the text.

Cat 4. Learning CS through analysing systems

This category introduces the idea of a whole, of which different concepts are parts. To study such a whole makes it possible to explore the role of the parts, and thereby to learn. The interview extract below is taken from a longer discussion, in which Abraham discussed what his team did in order to get started with their project and how he valued the different steps the team took.

Interviewer: Are there advantages of that?
Abraham: [...] look at the code a lot, mostly 'cause we were trying to figure out what was wrong and it did make us examine how everything was working. So I guess if the code had worked right away in a way we might not have looked at it so closely, I think.

Abraham had earlier told the interviewer that the team had been frustrated concerning the code they had selected. The code did not work when the students initially got it (see section 4), forcing them to analyse it in detail. Abraham mentioned this as an advantage, when explicitly asked by the interviewer. The direct answer, presented without hesitation, indicates the importance he placed upon reading code.

Reading the code in order to learn means that one needs to see both the parts and the whole that the code constitute. This category adds an aspect of a whole-parts relationship.

Cat 5. Learning CS through integrating systems

Contrasting this category with the previous one reveals an interesting difference. The whole software system is something that 'already exists' or is taken for granted in category 4: a way to learn is, in the fourth category, to split the unit into pieces and analyse the parts. In this category the process goes in the opposite direction. Putting the components together is a way to learn about and to create a system.

Adam: [...] The interaction between the motor daemon and the video daemon and the game server, you know, just having separate components that were totally independently written. It was very interesting and sort of a new experience to see that kind of thing and how they ended up working together so well.

A software system is built up of different components – each with a different function in the final system – that have to be integrated, Adam argued.

Thus, the new aspect in this category is the creation or integration of a whole.

Cat 6. Learning CS through giving meaning to concepts

The sixth category introduces a way of experiencing learning where personal insights concerning the computer science concepts studied and a personal understanding of computer science are sought for. Let us listen to a short quote by Alec:

Alec: [...] But as far as a learning factor, I learned an immense amount about RMI as far as I am concerned because I didn't even know it existed before. It is very interesting to me.

Alec both related the discussions to himself (by stating that he found it "very interesting" and by pointing out that he "didn't even know it existed before"), and put the learning in different contexts

As illustrated in other parts of Alec's interview, the focus of his statements was not on the CS topics, but instead on the experience of learning these topics.

In short, Alec was searching for a personal meaning of CS concepts, and thus illustrated the aspect that is new in this category.

Cat 7. Learning CS through developing as a professional

Alec answered a question in the following way:

Interviewer: Did you learn anything from that?

Alec: Yes, uh just basic design, how other people think, how you should approach such a large amount of code just thrown at you saying I want to know what it does and I want to know how it works. You can't just start picking it up and reading through it and saying Ok, I got it. So just a strategy I guess as to how I would do it in the future.

Alec discussed how "other people think", and how to relate himself to a large software system. He stressed the need to take personal decisions and contrasted this to a superficial way of approaching the system. Finally he referred this learning to his own future.

The new aspect in this category, worded by Alec in this quote, introduces an experience of learning that differs from what was described in the previous categories. A conscious professional development is now in focus.

6 Conclusions and implications

The current study investigates how advanced students, majoring in computer science, think about learning their subject area. Seven categories have been identified. While the first concerns the learning of using application programs, the remaining six describe how the students go about learning computer science.

The richness of the learning differs between the categories: The least advanced is the second, where only isolated concepts are learnt. The seventh, at the other extreme, shows a picture of learning as a professional development. Such learning includes components such as a search for personal meaning, an intention to integrate parts to a whole etc. The other categories fill the space between these two.

Clearly, the more advanced categories are the most desirable from a teachers' point of view. A student who strives to develop by studying, and who is searching for different aspects of what she/he is learning, is well placed to become a better computer scientist than someone who sees different concepts in isolation, and who does not try to integrate the parts or find the meaning of what she/he learns about.

Studying the full picture, a qualitative difference can be found between categories 2-4, on the one hand, and 5-7 on the other. The former take the computer science concepts for granted, whether they are isolated or parts of a program. The concepts exist, and the learning aims are restricted to understanding these concepts. In the latter categories, the computer science concepts are used to create something that is outside the original concepts, such as for example a program or personal development. We use the term *concepts-for-granted* approach to denote the first set of categories, and *concepts-in-context* approach for the second.

This distinction corresponds well with earlier research findings presented in section 3, where a dichotomy between *surface* and *deep* learning is discussed in Marton, Dall'Alba and Beaty (1993) and in later literature. While the former focuses on the text (the 'sign'), the latter is directed towards the content of the learning (the 'signed'). A strong correlation has been demonstrated in many studies (see Ramsden (2003) for an overview) between a deep approach to learning and a good learning outcome. Thus a similar good learning outcome can be expected for the *concepts-in-context* approach.

Results of this kind clearly point towards the need for educators to design courses that encourage the forms of learning that are described in the advanced categories. Different approaches have been proposed to encourage students to experience a phenomenon in advanced ways, corresponding to the higher categories. Ramsden (2003) argues, based on a meta-analyses, that clear goals, appropriate workload and appropriate assessment are among the factors in the teaching that encourage students to search for meaning-related (as opposed to superficial) constituents of their learning. It remains to be determined, however, to what extent these suggestions can be implemented to influence how students go about learning computer science.

The results of this study contribute to computer science education and to research in computer science education in the following ways. (1) They demonstrate that students tackle their learning in different ways. Some of these ways are more desirable than others, since they describe a more multi-faceted learning. (2) They offer guidance for teachers who want to improve their teaching. (3) They also contribute to research in computer science education by presenting results concerning the students' learning that constitute a starting point for future research, on both the teaching and the learning of computer science. (4) They offer an example of how data-driven, explorative, qualitative research, when performed according to theory-based guidelines, can constitute a basis for understanding how our students learn computer science.

7 Acknowledgement

ACE2008 and the author gratefully acknowledge the financial support of ACM SIGCSE in the presentation of this paper as an invited talk.

8 References

Berglund, A. (2005). *Learning computer systems in a distributed project course: The what, why, how and where* (Vol. 62). Uppsala, Sweden: Acta Universitatis Upsaliensis.

Booth, S. (1992). *Learning to Program. A phenomenographic perspective*. Gothenburg, Sweden: Acta Universitatis Gothoburgensis.

Bruce, C., Buckingham, L., Hynd, J., McMahon, C., Roggenkamp, M., & Stoodly, I. (2004). Ways of experiencing the act of learning to program: A phenomenographic study of introductory programming students at university. *Journal of Information Technology Education, 3*, 143 - 160.

Dahlin, B., & Regni, M. (1997). Conceptions of learning among Nepalese students. *Higher Education, 33*, 471 - 493.

Daniels, M. (1999). *Runestone, an International Student Collaboration Project* (No. 11). Linköping: NyIng report.

Denzin, N. K., & Lincoln, Y. S. (1994). *Handbook of Qualitative Research*: SAGE.

Eckerdal, A., & Berglund, A. (2005). *What Does It Take to Learn 'Programming Thinking'?* In Proceedings of the 1st International Computing Education Research (ICER) Workshop, Seattle, WA, USA, 135 -143.

Eckerdal, A., & Thuné, M. (2005). Novice Java programmers' conceptions of "object" and "class", and variation theory. *SIGCSE Bulletin inroads, 37*(3), 89–93.

Eklund-Myrskog, G. (1998). Students' conceptions of learning in different educational contexts. *Higher Education*, *35*, 299 - 316.

Lister, R., Box, I., Morrison, B., Tenenberg, J., & Westbrook, D. S. (2004). *The dimensions of variation in the teaching of data structures.* In Proceedings of the 9th annual ACM SIGCSE Conference. ITiCSE: Innovation and Technology in Computer Science Education, Leeds, United Kingdom, 92-96.

Marshall, D., Summers, M., & Woolnough, B. (1999). Students' conceptions of learning in an engineering context. *Higher Education*.

Marton, F., & Booth, S. (1997). *Learning and awareness*. Mahwah, New Jersey, USA: Lawrence Erlbaum Associates.

Marton, F., Dall'Alba, G., & Beaty, E. (1993). Conceptions of learning. *International Journal of Educational Research, 19*(3), 277–300.

Marton, F., Watkins, D., & Tang, C. (1997). Discontinuities and continuities in the experience of learning: an interview study of high-school students in Hong Kong. *Learning and Instruction, 7*(1), 21 - 48.

Ramsden, P. (2003). *Learning to teach in higher education* (2nd ed.). London, UK; New York, NY, USA: Routledge.