Applications of variation theory in computing education

Jarkko Suhonen

Department of Computer Science and Statistics
University of Joensuu
PO. Box 111
FI-80101 Joensuu
Finland

jsuhon@cs.joensuu.fi

Errol Thompson

2 Haven Grove,
Lower Hutt,
New Zealand

kiwiet@computer.org

Janet Davies

School of Curriculum and Pedagogy
Massey University
Private Bag 11 222,
Palmerston North
New Zealand

j.r.davies@massey.ac.nz

Kinshuk

Professor and Director
School of Computing and Information Systems
Athabasca University,
1 University Drive
Athabasca, Alberta T9S 3A3
Canada

Kinshuk@ieee.org

Abstract

There are various challenges in IT education. Students have difficulties in applying the key concepts, theories and techniques taught in introductory courses. At the same time, the backgrounds of students are becoming more and more diverse. Teachers work with heterogeneous student cohorts. Phenomenographic studies can be used to understand different perspectives of learners' understanding. Variation theory is a promising approach to improving the teaching of computing subjects. The theory can be applied to design variations in teaching that make explicit the different aspects of computing concepts. We discuss the role of variation theory in creating diversity in teaching practices to reach students with diverse backgrounds and expectations.

Keywords: variation theory, programming.

1 Introduction

Research has indicated that there are universal challenges in the teaching and learning of computing subjects. Learners are not reaching the desired performance levels and they have problems even with the basic concepts, especially after the first programming courses (McCracken et al., 2001, Lister et al., 2004a). Novice programmers, in particular, seem to have a wide range of difficulties and deficits (Robins et al., 2003). Students in computing subjects now have more varying backgrounds, skills, and motivations. Students might have been used to a range of teaching and learning approaches. Furthermore, students' behaviour patterns can also be different, which causes problems with communication, collaboration and interaction. Taking these factors into consideration, there is a need for novel pedagogical approaches in the CSE community to fulfil the needs of heterogeneous students. In this paper we discuss the applicability of variation theory as a course design tool to enrich teaching practices in computing subjects. We provide two examples of the use of variation theory and discuss the role of variation theory in assisting teachers to reach students with diverse backgrounds and expectations.

2 Application areas of variation theory in teaching and learning

2.1 What is variation theory?

The foundation of variation theory is based on the concept that people become aware of a phenomenon through the way that it varies from its environment (external horizon) or the way in which its internal parts vary in relation to one another (internal horizon) (Booth, 1992, Marton et al., 1993, Marton and Booth, 1997). When considering the nature of variations, there are those that on the surface seem obvious and those that are highlighting significant details. Variations help in identifying a phenomenon but there are also variations in the way that individuals recognise or are aware of a phenomenon. These variations in the ways in which people are aware of a phenomenon are used in a phenomenographic study to develop *categories of description*. It has been shown that there are a limited number of categories of description or ways in which people are aware of a phenomenon (Marton and Booth, 1997, Marton, 2000). These categories of description are usually placed in a hierarchy, where each higher layer incorporates or expands on the previous description. It is argued that learning occurs when a person becomes aware of a phenomenon in a different way.

Variation theory focuses on the way that a phenomenon is made visible in a teaching context (Marton and Tsui, 2003). Utilising knowledge of the variations with respect to how the phenomenon stands out from other things in the environment and with respect to the phenomenon's internal structure, it is possible to focus on the aspects that will help build the desired level of understanding. Marton et al. (2003) have defined the patterns of variations that are considered to be significant:

Contrast: "in order to experience something, a person must experience something else to compare it with" (p16). This may be a way of identifying critical aspects of the phenomenon with respect to other phenomena.

Generalisation is required with respect to the object of learning (p16). It isn't enough simply to see the object; we need to see variations in the use of the object to fully comprehend it. This involves recognising that some features are not critical to the identification of that phenomenon.

Separation of an aspect from other aspects is required. The object needs to be looked at from different angles. The aspect being examined "must vary while other aspects remain invariant" (p16).

Fusion is where "several critical aspects" need to be considered together. Those aspects must be experienced simultaneously (p16).

2.2 Variation theory in planning teaching

A core idea behind variation theory in the teaching context is that the teacher develops the teaching material with a perception of the content, that is, an "*intended object of learning*". Marton et al. (2003) argue that the object of learning is defined by "its critical features, that is, the features that must be discerned in order to constitute the meaning aimed for" (p22). A *critical feature* is a way of "distinguishing one way of thinking from another" (p24). In terms of an object of learning, variation theory differentiates effective and ineffective ways of experiencing that object. From these, it is then possible to look at the variations that can be used to 'enact' the phenomenon by using the patterns of variation or categories of description (Cope and Prosser, 2005).

The teacher can use appropriate variations within the identified space of learning to enact the object of learning (Marton et al., 2003). To create a space of learning there is a need to open up a dimension of variation. This variation may be through contrast with other objects in the learning environment or through variations in the frame of reference or ways of looking at the object of learning. The range of what can be learnt is defined by the space of learning and the range of variations presented in the learning space. The space of learning equates to an experience space in that it opens up ways of experiencing the object of learning (p25). One perspective of creating appropriate spaces of learning is to recognise that learners' previous experiences and knowledge influence their learning approaches and outcomes (Ramsden, 2003). If a teacher is aware of these variations, she or he can use examples, exercises or teaching approaches that are culturally relevant and familiar to the students. This increases the opportunity for the students to become aware of alternative ways of seeing the phenomenon. The linking with the learner's background is especially important with multi-cultural learner groups. Bates (1999) reports, for instance, that examples, idioms and writing styles may not be easily transferred between cultures. Learners can also have very different expectations of interaction patterns between instructors and students. For instance, in a western educational culture teachers often emphasise critical thinking skills, debate and discussion, while students from other cultures might have greater respect for the teacher and written text. In multi-cultural student cohorts it could be useful to raise the awareness of what learning is about and have some discussion about various ways of understanding the object of the study from different perspectives. For instance, the students' current conceptions and thinking processes can be uncovered through counter-examples or variations. As a result, learners' prior knowledge structures may change radically, when possibly existing naïve ideas and concepts have to be exchanged for viable knowledge (Bolhuis, 2003).

Important aspects of a learning domain are 'critical aspects'. When designing a course, it is the variations with respect to the critical aspects that help the learner to become aware of the phenomenon in the desired way (Cope, 2000). Critical aspects are identified as topics that are conceptually difficult, alien and/or counter-intuitive for students. Recent research results indicate that the critical concepts in the programming domain are object-oriented structures, pointers, recursion and procedural abstraction (Jenkins, 2001). Failing in the critical aspects will likely block learners' future progress and eventually lead to frustration and even to abandonment of studies.

What the student takes away from a learning situation as an understanding is the "lived object of learning" (Marton et al., 2003). Studies among computing themes have explored student conceptions of the phenomenon being studied (Booth, 1992, Berglund, 2002, Lister et al., 2004b, Eckerdal & Thuné, 2005). These studies help highlight the variations in perception that the students have as a result of the learning or the lived object of learning. In analysing the effectiveness of teaching, we can endeavour to assess how the intended, enacted, and lived objects of learning compare (Boyer, 1990, Glassick, 2000). The intended object of learning can be compared to the categories of description relevant to the phenomenon in order to determine both the level of awareness being focused on and the appropriateness of the intended object of learning. This could be considered an initial assessment of whether the teaching is intended to target the appropriate level of learning. The intended and enacted objects of learning can be compared to determine whether what is being taught matches what was intended to be taught. The student's lived object of learning can be compared against categories of description as a means of assessing the level of learning achieved or against the enacted level of learning to determine whether the enacted object of learning is being transferred to the lived object of learning as expected.

3 Examples of variation theory in computing education

3.1 Programming concepts

In teaching programming, there are at least two dimensions. There are the constructs or concepts that are to be introduced and the problems that those constructs or concepts are used to solve. By using variations in problems for a given construct or concept, we can emphasise the characteristics of that construct or concept within the context of the program. The construct or concept remains unchanging but the context varies. An alternative approach is to focus on a consistent problem but apply variations in the use of a construct or concept, or apply different constructs or concepts to the solving of that problem.

A fundamental concept in learning programming is the concept of a program. To start exploring the question 'What is a program(me)[2]?', a teacher might present several different examples of programmes that are familiar to the student. This provides a link to prior knowledge. These might include TV programmes, computer programs, recipes, building instructions, and instructions for roles in a restaurant (Thompson, 2006). The student would then discuss what the characteristics are that make these examples of programmes. The initial conception of a programme envisaged by the teacher is that of algorithm plus data (Wirth, 1976). Later examples, such as the restaurant example, challenge this thinking to focus on the idea of interacting entities or entities that respond to specific requests (Stein, 2003)[3].

The conceptual understanding of students can be further strengthened by comparing programme examples with non-programme examples and/or by letting students come up with their own examples of programmes and non-programmes. Further discussion can be held to investigate the characteristics of programmes. A goal of this activity is to discuss the key aspects and to construct a conceptual understanding of a programme and how it might be created. The examples are chosen to highlight the characteristics of a computer programme represented by the teacher's 'intended object of learning'. They are also chosen because they represent variations of the concept within the context familiar to the students (background knowledge).

3.2 Software development process

An approach to address the issues related to teaching of software development is to model the software development strategy. For example a strategy such as 'divide and conquer' or 'programming by intent' could be applied to a range of problems and applied consistently through the teaching of concepts and ideas. In this case, the process remains static but the problems vary, thus revealing that the process is not problem-dependent. If the opportunity were available, it might also be desirable to vary the programming language or development environment so that the learner saw that these strategies for programming are not dependent on the programming language or programming paradigm. They may be adapted but still utilised.

An alternative variation strategy is to use different software development strategies to solve the same problem. For example, the strategy of 'functional decomposition' might be contrasted with 'programming by intent', thus allowing the learner to see that alternative strategies can be used to address the same problem and possibly come to similar solutions.

With respect to the learning processes of software development, instead of modelling a software development strategy it would be necessary to model a process of discovery and exploration of domain concepts and of programming language constructs. This might include strategies for exploring framework features; for example, developing a series of automated tests to build an understanding of a framework object or feature. Again the modelled process could be applied to a range of learning issues within software development, including explorations for understanding the domain and for understanding the tools being used in software development.

4 Discussion and Conclusion

In this paper we have discussed the applicability of variation theory to the design of teaching with specific emphasis on computing subjects. We have endeavoured to emphasise how variation theory can be used to strengthen the understanding of a phenomenon by using variations relating to its critical aspects to form the space of learning. It has been suggested that knowledge of these critical aspects can assist in the planning process and that knowledge of possible variations can assist the teacher in helping students with their learning.

It is easy to argue that variations are always used in teaching. The question is whether those variations are effective. From a variation theory perspective, the variations that are effective are those that help present the critical aspects of a phenomenon and not simply those that show how the phenomenon varies from some other phenomenon. If we revisit the 'What is a program(me)?' example, what worked were the variations that helped students to think about algorithm and data. They were presented with a number of familiar and maybe not so

[2] The use of the British spelling of programme in this context is deliberate and is intended to provide a link with the programme concept outside of the computing world. This is to enhance the linkage with the student's background knowledge.

[3] Note research is currently under way to gain a better understanding of the conceptions of an object-oriented program.

familiar program(me)s that focused on these two aspects. They were also given non-examples that focused either on the data or on the algorithm but did not include both. The examples also avoided including other complexities of programming such as syntax and semantics.

If variations are to be used in teaching then it isn't adequate simply to know the definitions and to be able to present these to the students. The teacher needs to understand the critical aspects that are encapsulated in those definitions and the variations of examples and questions that will help create an enacted object of learning that makes those critical aspects visible.

When a course is evaluated using variation theory, one looks for those variations that highlight the critical aspects. It is these that are going to define the enacted object of learning and create the space of learning. It therefore seems logical, if we are to use variation theory in planning teaching, that the course designer and teacher understand what these critical aspects are and the variations that will make them visible to the learner. Future research could be focused, for instance, on investigating what kinds of variation would be especially useful in computing subjects.

5 References

Bates, T. (1999) Cultural and ethical issues in international distance education. *Proceedings of the Engaging Partnerships Collaboration and Partnership in Distance Education UBC/CREAD conference.* Vancouver, Canada.

Berglund, A. (2002) On the understanding of computer network protocols. *Department of Information Technology.* Uppsala, Sweden, Uppsala University.

Bolhuis, S. (2003) Towards process-oriented teaching for self-directed lifelong learning: a multidimensional perspective. *Learning and Instruction,* 13**,** 327-347.

Booth, S. A. (1992) *Learning to program: A phenomenographic perspective,* Göteborg, Acta Universtatis Gothoburgensis.

Boyer, E. L. (1990) Scholarship reconsidered: priorities of the professoriate, San Francisco, Jossey-Bass Inc.

Cope, C. (2000) Educationally critical aspects of the experience of learning about the concept of an information system. *School of Arts and Education Faculty at Bendigo.* Bundoora, Victoria, La Trobe University. From Education/Phenomenography/Cope2000Thesis.pdf.

Cope, C. & Prosser, M. (2005) Identifying didactic knowledge: An emperical study of the educationally critical aspects of learning about information systems. *Higher Education,* 49**,** 345-372.

Glassick, C. E. (2000) Reconsidering scholarship. Journal of Public Health Management and Practice, 6, 4-9.

Eckerdal, A. & Thuné, M. (2005) Novice Java programmers' conceptions of "object" and "class", and variation theory. *ITiCSE'05.* Monte de Caparica, Portugal, ACM. From LearningToProgram/VariationTheory/EckerdalThune 2005ITiCSE.pdf.

Jenkins, T. (2001) The motivation of students of programming. *Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE'01).* Canterbury, UK, ACM Press.

Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., McCartney, R., Moström, J. E., Sanders, K., Seppälä, O., Simon, B. & Thomas, L. (2004a) A multi-national study of reading and tracing skills in novice programmers. *Inroads - The SIGCSE Bulletin,* 36**,** 119-150.

Lister, R., Box, I., Morrison, B., Teneberg, J. & Westbrook, S. (2004b) The dimensions of variation in the teaching of data structures. *Inroads - The SIGCSE Bulletin,* 36**,** 92-96.

Marton, F. (2000) The structure of awareness. In Bowden, J. A. & Walsh, E. (Eds.) *Phenomenography.* Melbourne, Australia, RMIT University Press.

Marton, F. & Booth, S. A. (1997) *Learning and awareness,* Mahwah, NJ, Lawrence Erlbaum Associates.

Marton, F., Runesson, U. & Tsui, A. B. M. (2003) The space of learning. In Marton, F. & Tsui, A. B. M. (Eds.) *Classroom discourse and the space of learning.* Mahwah, NJ; London, Lawrence Erlbaum Associates, Publishers.

Marton, F. & Tsui, A. B. M. (Eds.) (2003) *Classroom discourse and the space of learning,* Mahwah, NJ; London, Lawrence Erlbaum Associates, Publishers.

McCracken, M., Kolikant, Y. B.-D., Almstrum, V., Laxer, C., Diaz, D., Thomas, L., Guzdial, M., Utting, I., Hagan, D. & Wilusz, T. (2001) A multi-national, multi-institutional study of assessment of programming skills for first-year CS students. *Inroads - The SIGCSE Bulletin,* 33**,** 125-140.

Ramsden, P. (2003) *Learning to teach in higher education,* London, Routledge Falmer.

Robins, A., Rountree, J. & Rountree, N. (2003) Learning and teaching programming: A review and discussion. *Computer Science Education,* 13**,** 137-172.

Stein, L. A. (2003) Interactive programming in Java. From http://www.cs101.org/ipij/ObjectOriented/Interactive ProgramminginJava.

Thompson, E. (2006) What is a program(me)? From http://www.massey.ac.nz/~elthomps/What_is_a_program/programmes.html.

Wirth, N. (1976) *Algorithms + data structures = programs,* Englewood Cliffs NJ, Prentice Hall.