

DISCOURSES OF PROGRAMMING TEACHING WITHIN COMPULSORY EDUCATION – FIXED OR CHANGEABLE?

Eva Björkholm and Susanne Engström
KTH Royal Institute of Technology, Stockholm, Sweden

Knowledge linked to programming has recently been extensively strengthened in curricula and syllabi in the Swedish compulsory school. The introduction of this new content requires that teachers have to be trained in programming and programming teaching. The aim of this study is to investigate what content and values that emerge as important in a professional development course and in the participating teachers' teaching in their classrooms. Data was collected by observation of the teaching sessions within the course as well as in three of the teachers' teaching, where notes were made continuously. By using a discourse analytical perspective, content and values that emerged as important within the teaching were identified. The findings show that the content knowledge in the teacher training course is taken for granted within the school context, as well as in itself. In addition, normative values and steering strategies have been identified within the teacher training course. In the classroom studies, similar values were identified among the teachers. A potential problem within the teaching practice was found in terms of a lack of progression of the content related to programming, as well as difficulties in relating the use of programming to relevant contexts.

Keywords: Technology Education, Classroom Discourse

1. INTRODUCTION

Knowledge linked to programming has recently been extensively strengthened in curricula and syllabi in the Swedish compulsory school, mainly including the subjects of mathematics and technology (National Agency of Education, 2016). It is argued that this kind of knowledge is needed for students to manage their digital world and deal with their digital tools. In addition, increasing interest in technology in order to get more people into technology training is highlighted (Digitization Commission, 2014). Such a vigorous introduction of new content in school involves substantial efforts on implementation processes in terms of professional development activities for teachers, purchase of equipment, etc. In different areas of the implementation, several actors have interests striving for impact such as university departments of computer science and of teacher training, science centres, non-profit organizations, the business sector financing development project and companies selling digital equipment, and publishers. In this context of the forming of a new topic in school, the culture mediated by the actors becomes crucial for what specific content is emphasized and how it is taught. Previous studies suggest that the content in programming involves decomposition of the problem, abstraction, logical thinking, creating algorithms, and debugging (Yadav et al., 2014; Jaipai-Jamani & Angeli, 2016; Sullivan & Heffernan, 2016). Moreover, programming educators often teach with a narrow focus upon the technology tools, without explicitly taking students' and teachers' views into consideration (Berglund & Lister, 2010). A questionnaire focusing on teachers' views about student engagement about programming showed that students' in-the-moment enjoyment was valued highly (Black et al., 2013).

In this paper, we present a study of a professional development course for teachers focusing on programming. The course includes teaching sessions on programming that the participating teachers design and carry out in their classes. The aim of this study is to examine what content related to programming that emerges as important in the teaching throughout the course implementation, and what implicit values concerning this content that are imbedded in the teaching practice. Research question: What content and what values concerning programming emerge as important in the teaching practice in the professional development course and in the teachers' own teaching?

2. METHOD

Data were obtained through observation of teaching in all course sessions in the professional development course. The course was conducted by a science center, and the teaching was carried out by programming

skilled people who were not trained teachers. The course consisted of a total of six training sessions, each lasting a full day. In the course participated 16 practicing teachers (grades 1-9). The course would result in a small project on programming that the teachers implemented in their own classes, and that was presented and reported at the final session of the course. In addition, three of the participating teachers, teaching in Grade 3, 4 and 7 (pupils aged 9, 10 and 13 years) respectively, were observed when implementing the project in their classes. The three teachers themselves, at the request of the whole group, reported their interest in being observed while teaching in their classes. Two lessons with the duration of approximately one hour with a few weeks apart were observed for each teacher.

During all the observations, notes were made continuously by two researchers, independent of each other. Immediately after each teaching session, the researchers met and discussed their individual observations, and summarized the analysis. The analyzes of the lessons were based partly on identifying what knowledge content in relation to programming that was highlighted in the teaching, and a discourse analytical perspective (Börjesson & Palmblad, 2015; Gee, 2014) focusing on the values and steering strategies within the teaching practice (Öhman, 2015).

3. RESULTS

When analysing the observations of the teacher training course, the following results were found:

- *Normative values emerging within the discourse of the teacher training course*

(1) The specific knowledge content is seen as something obvious within the school context, and something implicitly obvious in itself:

- Formulation of well-defined thoughts, organisation of solutions and stepwise instructions, structures of problem solving, more explicit: coding. Understand the need of an exact language, not allowable to misunderstand. No interpretable knowledge, focus on the logic of programming and the essentiality of coding. The code has to be seen separated from the artefact. Important to learn the “handicraft”= coding, try and make mistakes, to troubleshoot. Necessary with massive practicing.

- Blue-Bots, Lego robots seem to be content in itself, a great focus on the robots and the controlling of the specific artefacts. A learning progression from using robots (*Blue-Bots*) to using just the computer, also when *Scratch* is used, the teachers still focus on the artefacts (the aquaria). The specific knowledge *coding* is not made visible clearly.

(2) The attitude: much narrowed and with worship of the nature of programming and its possibilities:

- No unnecessarily talking, no discussions, short and quick instructions. The “right way” to do things, quick learning sequences, aiming to be inspiring, a great focus on creative skills, construct new things without reflections or problematizing. A playful attitude with uncomplicated views on related issues, and with emphasis on programming as easy and without limits, with an intention to be effective. Also an inviting attitude; important to share ideas and programming knowledge, “programming could lead us to something without boundaries”. Programming skills are highly valued.

- *Steering strategies used by the course leaders within the discourse of the teacher training course*

(1) The strategy of trivializing:

- Based on the course leaders statements about compulsory school teachers as programming incompetent. “How do you feel”? “Do you have fun”? “Do not panic”! “Have fun”! “This is easy, just as easy as writing a story... hello, we are going to play with robots”!
- The teachers accept the course leaders’ way of talking, and adapt to the expectations, they became “pupils”, applaud, laugh, trivialize themselves.

(2) The strategy of disjunction:

- The course leaders talk about “we and you”, “what do we mean by programming?” We, inside the programming context – you, excluded from that context. “You will get some knowledge”, “taking some knowledge to your classes”, “something that make your pupils positive, interested and engaged”, “the pupils have do believe that everything is possible”.

(3) The injunction of logic:

- The course leaders give orders about making everything right and keep to the logic: “Follow the instructions”! “Troubleshoot”! “Be careful and accurate”!

When analysing the classroom observations, the following results emerged:

Grade 3 classroom: the knowledge content (the concept of loop, steering Blue-Bots in relevant contexts, using a code language with consensus) is made clear, and the pupils have the possibility to learn. However, after a while, the pupils leave the coding, and focus more on the construction of the Blue-Bots-track, therefore the progression of learning becomes negative. Values expressed by the teacher: focus on teaching, aims for the teaching sequence, feedback to the pupils, a teaching environment based on discussions.

Grade 4 classroom: the knowledge content (decompose a problem, describe an event, divide the problem into parts, different parts will be structured in a specific order - “that is programming”!) is clarified. Good progression. However, as the context deals with science in a specific way, it turns out to be problematic and not logic. Values expressed by the teacher: expresses very clearly that a good structure is important, a serious approach (talking about useful knowledge – not only having fun), repeating the specific learning object, reminds the pupils about what to do, gives short and explicit guiding.

Grade 7 classroom: the knowledge content (focus on artefacts Lego robots, unclear about programming as content, give instructions to pupils about what to do in detail, no feedback or discussions). Values expressed by the teacher: short instructions, efficiency, no discussions, “Go on”! “Have fun!”

4. DISCUSSION

In this study we have identified content and values that emerge as important in the programming education context. The content itself was taken for granted, and was similar to content highlighted in previous studies (Yadav et al., 2014; Jaipai-Jamani & Angeli, 2016; Sullivan & Heffernan, 2016). In other words, the consensus of the content that seems to be established seems not to be questioned. A lack of clear progression in the teaching, and problems in relating the use of programming to different contexts were also identified (cf. Thuné & Eckerdahl, 2009).

5. REFERENCES

- Berglund, A., & Lister, R. (2010). Introductory programming and the didactic triangle. In *Proceedings of the Twelfth Australasian Conference on Computing Education-Volume 103* (pp. 35-44). Australian Computer Society, Inc.
- Black, J., Brodie, J., Curzon, P., Mykietiak, C., McOwan, P. W., & Meagher, L. R. (2013). Making computing interesting to school students: teachers' perspectives. In *Proceedings of the 18th ACM conference on Innovation and technology in computer science education* (pp. 255-260). ACM.
- Börjesson, M. & Palmblad, E. (2007). *Diskursanalys i praktiken*. Malmö: Liber.
- Digitization Commission (2014). Summary of report on ICT in schools – report to the Swedish Government published in March 2014 (SOU 2014:13)
- Gee, J. P. (2014). *An introduction to discourse analysis: Theory and method*. Routledge.
- Jaipal-Jamani, K., & Angeli, C. (2016). Effect of robotics on elementary preservice teachers' self-efficacy, science learning, and computational thinking. *Journal of Science Education and Technology*, 1-18.
- National Agency of Education (2016). Curriculum for the compulsory school, preschool class and the recreation centre.
- Sullivan, F. R., & Heffernan, J. (2016). Robotic construction kits as computational manipulatives for learning in the STEM disciplines. *Journal of Research on Technology in Education*, 48(2), 105-128.
- Thuné, M. & Eckerdahl, A. (2009). Variation theory applied to students' conceptions of computer programming. *European Journal of Engineering Education*, 34(4), 339-347.
- Yadav, A., Berges, M., Sands, P., & Good, J. (2016). Measuring computer science pedagogical content knowledge: An exploratory analysis of teaching vignettes to measure teacher knowledge. In *Proceedings of the 11th Workshop in Primary and Secondary Computing Education* (pp. 92-95).
- Öhman, M. (2007). Styrning i “jumpasalen”. In: M. Börjesson & E. Palmblad (eds.). *Diskursanalys i praktiken*, p. 128-148.