TEACHER PERCEPTION OF OBSTACLES AND OPPORTUNITIES IN THE INTEGRATION OF PROGRAMMING IN K-12 SETTINGS

N. Humble, P. Mozelius

Mid Sweden University (SWEDEN)

Abstract

The act of programming can be traced back to the 19th century and Ada Lovelace. In the context of history, the idea to combine programming with mathematics and technology is not strange. Today programming is a much more widespread phenomenon. But the relevance of programming in mathematics and technology is still very obvious. However, with the mandatory addition of programming in K-12 setting a debate has been sparked concerning the impact on teaching activities. The aim of this study was to analyse and discuss possible obstacles and opportunities in integrating programming in K-12 settings from a teacher perspective. The two important main questions to answer were: 1) which are teachers’ perceived obstacles and opportunities in the integration of programming in K-12 setting? 2) which are the differences and similarities in expectation between teachers with and without earlier programming experience?

This study was carried out inspired of action research with the objective of improving an existing programming course for mathematic and technology teachers. Action research is a strategy that do not only strive to generate new knowledge but also has the aim to improve real world phenomena. Authors in this study had the double roles of being teachers and researchers, with the idea of establishing a cyclical process where course participants feedback should lead to extensions of the future course versions. Data were collected in the teacher training course and consist of 44 submitted essay answers on the question: In which aspects might programming be a positive and/or negative enhancement of you daily teaching? Collected data were grouped into categories with the use of content analysis.

Results indicate that there are both perceived opportunities and perceived obstacles among teachers concerning integrating programming in mathematics and technology. An obvious finding was the mix of positive and negative attitudes in the vast majority of essays. In the category of obstacles many teachers brought up the risk of time trouble in both their professional development and in their teaching activities. In the category of opportunities several teachers mentioned the potential of programming as a new and motivating learning tool in their subjects. Finally, it seems important for the participants to get concrete takeaways from the course that could be used in their daily teaching activities.

Keywords: programming, programming education, K-12, teacher perception, teacher professional training.

1 INTRODUCTION

The history of programming can be traced back to the 1900th century and the world's first programmer. In the 1840s Ada Lovelace, daughter of the famous poet Lord Byron, writes down some notes to what we today recognise as the world's first computer program [1]. Besides using technology, a machine, to do calculus, Lovelace was also the first who discussed to possibilities that computers could do other things, such as write music and texts with the help of programming, and not only calculus [1]. Looking back, the idea and practise of combining programming with other subjects, especially mathematics and technology, have been there for quite some time but have recently been actualized yet again with the mandatory addition of programming in K-12 educational setting in many countries [2, 3]. Concerns and opinions on how this integration will affect learning activities and other subjects have been debated by both media and researchers, but what are the teacher's perceptions?

In this study the aim was to analyse and discuss possible obstacles and opportunities in integrating programming in K-12 settings from a teacher perspective. In our work the two important main questions to answer were: 1) which are teachers’ perceived obstacles and opportunities in the integration of programming in K-12 setting? 2) which are the differences and similarities in expectation between teachers with and without earlier programming experience?
2 EXTENDED BACKGROUND

One of the reasons for introducing computer science and programming in many countries K-12 educational curriculum is a will to equip students with computational thinking skills which are seen as important for future generation to have for them to meet the needs of future society [2, 3]. However, what computational thinking skills are is not clear and the definition differs among many researchers [4, 5, 6]. Even if the concept of computational thinking is a bit vague there are some common components between different researchers: decomposition, abstraction, algorithms and debugging [4]. A more recent attempt to summarise and define computational thinking also added the components iteration and generalisation to the list above and defined computational thinking as “the conceptual foundation required to solve problems effectively and efficiently (i.e., algorithmically, with or without the assistance of computers) with solutions that are reusable in different contexts” [4].

Another computational thinking concept that has been analysed is automation [7], with the idea of automation as the means of evidencing computational thinking. A common perspective on automation is a digital computing device that runs programming code, but there are also studies defining automation as a process that must not necessarily include any digital device [8]. If digital automation is the execution of programming code on computers, analogue automation can consist of unplugged programming activities in unplugged mode only [9]. In digital automation programming can be divided into textual programming and block programming. Textual programming in K-12 settings are often carried out in high-level programming languages such as Java or Python. In textual programming concepts as statements, selection and iteration are all exclusively built up by combinations of textual codes that are checked and run by a compiler or an interpreter [10]. Block programming on the other hand consists of a programming environment where visual blocks can be dragged and dropped to lower the threshold to programming for novices and younger age groups [11].

There are several perceived opportunities in the introduction of programming in K-12 settings, for example in that it will help students develop skills that are important to other subjects (such as reasoning skills, problem solving and self-efficacy in mathematics), and that it will encourage students to be creators and not only consumers of technology [12, 13]. However, there are also several perceived challenges in introducing programming in K-12 settings. Some of these challenges are the question of handling the limitations in access to the internet, computers, other technical devices and lack of students’ technical skills and motivation [13]; and having enough time to both teach the students programming and using it as a tool for developing other skills [14]. Research have also indicated that the development of computational thinking and secondary skills in other subjects (such as problem solving in mathematics) may not necessarily follow by the introduction of programming in itself; rather it seems that it has to be more consciously and directly taught with programming as a tool for the development to occur [3, 12].

3 THE COURSE

The studied course of 7.5 ECTS was given on 25% study pace in twenty weeks during the 2018 autumn semester and the 2019 spring semester. The course was designed for blended learning with eight face-to-face meetings combined with the online learning periods organised in the Moodle virtual learning platform. Since all course participants work full-time as teachers in mathematics and technology the 25% study pace seemed like a realistic choice. At the initial face-to-face workshops course participants were encouraged to create local study groups to collaborate and share ideas. To provide content and activities to meet the learning objectives in the syllabus, the course is divided into five sections [15]. Each section combines lectures, code analysis, interactive workshops, programming examples, reflective assignment and programming assignments to meet the above-mentioned needs on the course.

3.1 Section 1: Programming in school, why, what and how?

Unlike traditional courses a first course section was outlined to presented and discuss the ‘why’ and ‘what’ of programming, before introducing the ‘how’ of programming. Lessons learnt from earlier similar courses is that only a minority of the participants has earlier programming experiences. This initial face-to-face session also discusses more general aspects of digitalisation of K-12 education. Finally, the how-part takes care of the installation of Python and Scratch programming environments.
3.2 Section 2: The fundamental building blocks of programming

In all courses on introductory programming, the fundamental building blocks such as variables, constants, selection and iteration must be presented and practised before course participants will be able to create their own programs. This section focuses on developing knowledge in the fundamentals of programming.

3.3 Section 3: Didactics for Technology and Mathematics

What distinguish this course from many other programming courses is that it also has an alignment towards technology education and mathematics education. This section aims to develop the participants skills and knowledge in how programming can be used as a tool in technology education and mathematics education.

3.4 Section 4: Didactics for programming education

An important part of this programming course is that all the participants themselves are teachers in either mathematics or technology in K-12 education and have the expectation on them to use programming as a tool in their subject. However, they also need some knowledge in how to teach basic programming to others, since many of their students will arrive with little or no prior experience in programming. This section aims to meet this need by developing the participants skills and knowledge in how to introduce others to programming within mathematics education and technology education.

3.5 Section 5: Project work

Important in these kinds of programming courses, where the participants are not only there to learn programming but also to do something explicit with this knowledge, is that they get something concrete with them from the course that they later can use in their daily work, beside the knowledge and skills that they have developed. This last section draws on the skills and knowledge developed in earlier sections and focuses on giving the participants time and tutoring in creating their own programming material to be used in their teaching activities in mathematics education or technology education.

4 METHODOLOGY

Action research is a methodology where the objective is to seek and bring about change and where the researchers often are as involved in the action as they are in the research [16]. The process of action research is often an interaction between action and research where the issue or problem is studied, findings presented and analysed, and suggestions are made for an iterative improvement of a social process [16]. This study was carried out with an approach inspired by action research and with the objective to study possible obstacles and opportunities in an existing programming course. The studied course is designed for mathematic and technology teachers, where authors have had the multiple roles of subject matter experts, course developers, teachers and researchers.

In the classical action research that origins from Kurt Lewin [17], there should be a close collaboration between the researchers and the studied process. Another action research idea is that both parts in the collaboration should learn and benefit from the process. The idea of not separating research from the studied process can be seen as both the strength and the weakness of action research, with the risk of bias as the negative aspect. Finally, the aim of action research should be to contribute to solutions to practical problems in real world settings [18].

4.1 Data collection

Data were collected from an online forum where course participants posted essays and commented on other participants posted essays. 44 submitted essay were submitted as answers to the question: “In which aspects might programming be a positive and/or negative enhancement of your daily teaching?” The participants also answered a survey before the start of the course where they self-assessed their earlier experience in programming. 22 answered that they had some degree of earlier experience in programming, 19 answered that they had no or very low earlier experience in programming, and 3 did not answer the survey.
4.2 Data analysis

The typically use of content analysis is for research with an interest in communication content and where a systematic investigation of large amounts of data is needed for identification of topics proportion in the studied material [19, 20]. Inductive coding techniques can be used when there are limitations in previous research or theory for which codes to look for [20]. In this study content analysis was used to analyse and group the collected data into categories with the use of inductive coding techniques where the categories/codes were generated during the analysis.

5 FINDINGS AND DISCUSSIONS

An obvious finding was that the vast majority of the essays brought up a mixture of positive and negative attitudes towards the integration of programming through the perceived obstacles and opportunities mentioned. Maybe this is not that surprising since a big part of K-12 education is to teach students to be nuanced in their answers.

5.1 Perceived obstacles

As mentioned earlier, the question of having enough time to both teach students how to program and using it as a tool for developing other skills is of great importance [14] and was also mentioned as an obstacle in the essays. Nearly 9 out of 10 essays brought up time as an obstacle, for their own learning (many say that they get to little or no time from their employer to develop their own programming skills) and for their subject and students (especially in mathematics many teachers say that there is not enough time to teach anything other than mathematics).

Another obstacle mentioned in the essays, in about 3 out of 10, was the perceived lack of guidance, expectation and clarity of the integration from the Swedish National Agency for Education and schoolboards. In much the same way that some concepts in computer science, such as computational thinking, do not have a uniform definition [4, 5, 6] which makes it hard to investigate; it is perceived hard to teach something if you do not understand what you are expected to teach.

A challenge in integrating programming in K-12 education is the handling of limitations in access to the internet, computers and technical devises [13]. This was also brought up in about 2 out of 10 essays, where some also mentioned the risk of this integration leading to unequal development of student's technical skills due to economic and technical conditions in different schools and teachers' various interests and motivation for using programming in their teaching activities.

In nearly 2 out of 10 essays the obstacle of programming having negative consequences on students learning (for example on motivation and interests) were mentioned. Some of the reasons given were that programming could be too difficult for some students to learn and that the reason for using programming in the subject could be perceived as unclear for the students. This is also some of the challenges mentioned in previous research [13], but research also suggests that there might be some cause for worry. The secondary skills that many have hoped develops by learning programming (such as problem solving in mathematics) does not necessarily occur, but does rather seem to have to be taught directly and consciously by using programming as a tool for this development [3, 12]. If the teacher lacks a plan for what and how other skills (such as mathematical skills) will be developed by the use of programming and simple focusing on teaching programming and hopes that it will lead to the development of other skills, there could be cause for worry.

5.2 Perceived opportunities

As stated earlier one of the reasons in many countries for introducing programming in K-12 settings is a will to equip students with skills that are perceived as important or necessary for students to have in society today and in the future [2, 3]. About 6 out of 10 essays also mentioned the opportunity of implementing programming having these positive secondary effects. Essays mentioned that programming could have positive effects on students’ understandings, skills, creativity, curiosity, freedom, self-confidence and inspiration in other subjects and that it could help them in future society and education.

Further, previous research lifts the opportunity of students developing skills (such as reasoning skills, problem solving and self-efficacy in mathematics) by learning and using programming, and that it could encourage them to go from consumers of the technology and instead be creators [12, 13]. The opportunity of programming having these positive effects on the subject where it is integrated is also
mentioned in the essays. About 6 out of 10 essays mentions that programming could have a positive effect on students learning in mathematics or technology by giving a deeper, alternative or more fun learning experience in the subject, and that it could possibly make education more efficient.

5.3 Expectations

There are no significant difference in the essay's answers on perceived obstacles and opportunities in integrating programming between teacher with or without earlier experience in programming. However, there are some interesting answers on what expectation they have on a programming course for teachers, with some minor differences between those with and without earlier programming experience.

About 6 out of 10 of those with earlier experience in programming mentioned that they expect to learn how to program, get a foundation for future learning or get more confidence in programming from the course. The same was mentioned by nearly 9 out of 10 of those who had no earlier experience in programming.

About 5 out of 10 of those with earlier experience in programming mentioned in their essays that they expect the course to give them knowledge in how to use programming as a tool in mathematics or technology or to understand what expectations there are on them as teachers to use programming in their teaching activities. About 3 out of 10 of those without earlier experience in programming mentioned the same.

Nearly 3 out of 10 of those with earlier experience in programming mentioned that they expect the course to give them complete teaching materials for them to use in their teaching activities, or ideas or inspiration to create their own teaching material. About 5 out of 10 of those with no earlier experience in programming mentioned the same.

As mentioned, the difference between those with and without earlier programming experience is not very significant. What is interesting however, is that even though a majority of the course participant expect to learn how to program about half of them also expect to get something else out of the course, such as knowledge in how to use it as a tool or complete teaching material for them to use in their own teaching activities. These mixed expectations could possibly be a problem for the participants learning. Papert noted when he was having a summer workshop in programming for teachers that something was blocking their learning experience and concluded: “Their awareness of being teachers was preventing them from giving themselves over fully to experience what they were doing as intellectually exciting and joyful in its own right, for what it could bring them as private individuals. The major obstacle in the way of teachers becoming learners is inhibition about learning.” [21]. The double role of teacher and learner could be perceived as a strength that helps the participants to more actively look for concrete applications of their knowledge during the course, but it could also decrease their own learning and learning experience.

6 CONCLUSION

The aim of this study was to analyse and discuss possible obstacles and opportunities in integrating programming in K-12 settings from a teacher perspective. The study have been done with two main questions to answer.

Considering the first research question, on teachers’ perceived obstacles and opportunities in the integration of programming in K-12 setting, two main obstacles and two main opportunities have been identified. The most important obstacle identified is the perceived time-trouble of integrating programming. Another important obstacle identified is the perceived lack of directives on how to integrate programming in teaching and learning activities. The first important opportunity identified is the possible development of skills that the students will benefit from in other subjects and society. The second important opportunity is identified as the possibility for deeper or alternative learning experience in the subjects where programming is integrated.

Considering the second research question, on the differences and similarities in expectation between teachers with and without earlier programming experience, no significant differences were found. However, three interesting expectations on the programming course itself was identified. The expectations to learn programming, the expectation to understand how to use programming as a tool in their teaching activities, and the expectation to get teaching materials out of the course.
Some of the main difficulties in this integration has yet again actualising the words of Papert: “The central practical problem is to find ways in which teachers who are at different places in the willingness to work for change can do so. There cannot be a uniform change across the board – any attempt to do that will reduce the pace of change to that of the least common denominator. Society cannot afford to keep back its potentially best teachers simply because some, or even most, are unwilling.” [21]. The conclusion of this study is that a majority is willing and sees opportunities in the integration of programming in K-12 setting; but that there are obstacles for this to be successful due to a lack of time and clear directives for the integration.

The recommendation is that teachers, with and without earlier experience in programming, are given time, both for their own professional development and for planning, reflecting, discussing and upgrading their teaching materials to meet these demands on themselves and their subjects.

7 FUTURE WORK

The next natural step would be to address teachers’ perceived obstacles and perceived opportunities by revising the course and its content. In the spirit of classical action research this should be done iteratively in a series of steps where amendments and changes can be commented in the essays from the next course batch, to determine if changes are satisfactory.

REFERENCES


15. P. Mozelius, "Teaching The Teachers To Program: On Course Design and Didactic Concepts.", *11th annual International Conference of Education, Research and Innovation*, IATED, vol. 11, 2018

16. J. Harris & V. White, *A distortionary of social work and social care*. Oxford University Press, 2018


18. R. N. Rapoport, "Three dilemmas in action research: with special reference to the Tavistock experience.", *Human relations* 23.6, pp. 499-513, 1970

