Student teachers' preconceptions of programming as a content in the subject technology

Anna Perez, Linnaeus University
Correspondence: anna.perez@lnu.se
Maria Svensson, University of Gothenburg, Göteborg
Jonas Hallström, Linköping University, Norrköping,

ABSTRACT

In many countries, student teachers are not adequately prepared to teach programming in technology education once they have completed their training. There is a corresponding inadequacy of research regarding pre-service programming education in technology, although in recent years research in this area has increased. There is, however, a lack of research specifically regarding student teachers' preconceptions about programming, which would be important for developing competences needed for teaching in technology.

This paper presents a study with the aim of describing student teachers' preconceptions about teaching programming in technology. The study uses a phenomenographic approach investigating eight student teachers' preconceptions after a five-week technology course preparing for primary education, grades 4-6 (teaching pupils aged 10-12). Semi-structured interviews have been conducted with student teachers from two different higher education institutions in Sweden. From the first step of the analysis, three tentative categories have been obtained, describing student teachers' preconceptions as: 1) an understanding of a language and/or a tool, 2) an understanding and use of a language or a tool to solve technological problems, and as 3) a way of understanding and describing a technological environment.

The results of the study will contribute with research-based knowledge useful for developing new approaches on how to vary and design the teaching of programming in technology for student teachers to develop skills that are important for their future profession.

Key Words: student teachers, technology education, programming, phenomenography

1. INTRODUCTION

The technology we use today is more digitalised than it has ever been before. Over the last 20 years, our everyday lives have changed and become increasingly digitalised in the form of, for example, lawnmowers, vacuum cleaners, bank transactions, etc. The increased digitalisation of society has contributed to changes in school curriculum documents in Sweden (Skolverket, 2017) and in other countries. In Sweden, digital technology and programming have been included as educational content in, for example, the technology syllabus. Since 2018, teachers in the Swedish compulsory school (pupils aged 7-16) are therefore expected to have acquired the knowledge to teach programming as part of the technology subject. However, many teachers take on this new educational content with uncertainty (Sentance & Csizmadia, 2017; Webb et al., 2017) because the curriculum indicates what pupils should be taught, not how this should be taught or how to address learners' difficulties (Passey, 2017). There is therefore a need for more knowledge about what teaching programming as part of the technology subject entails.

In Sweden, programming has thus not been implemented as its own subject but as part of existing subjects with their established subject-specific content and learning goals. Consequently, teaching activities including programming have to be aligned with existing curriculum intentions (Vinnervik, 2022). The Swedish curriculum in technology involves developing pupils' capabilities to identify and analyse structure and function of existing programmed technology solutions, but also develop their abilities to design new ones and control them with programming (Skolverket, 2017). An important mission for teacher education should be to instil student teachers with the ability to plan, implement and evaluate teaching where such content is processed. Therefore, it is imperative to investigate how student teachers understand their upcoming teaching of programming in the technology subject, after completing their own technology teacher education course.

In many countries, student teachers are not adequately prepared to teach programming in technology education once they have completed their training. There is a corresponding inadequacy of research regarding pre-service programming education in technology, although in recent years research in this area has increased. Yet there is still a lack of research specifically regarding student teachers' knowledge about programming as a content in the subject technology.

1.1. Aim and research question

This paper presents a study with the aim of describing student teachers' preconceptions about teaching programming in technology after completing the technology course in teacher education. The student teachers in this study have not yet had the opportunity to concretize their understanding of teaching programming in classrooms. The following research question is posed:

What are student teachers' preconceptions about teaching programming in technology education?

2. BACKGROUND

Computers and computing were introduced in Swedish schools from the late 1960s, although it took until the 1980s before there was a more concerted effort when the Government pushed the interdisciplinary subject area "datalära" in compulsory school (Nissen & Stenliden, 2023). After several more or less successful attempts at introducing computing and ICT in schools more broadly, the Swedish National Agency of Education included programming in the subjects of technology and mathematics in compulsory school in 2018. As a consequence, a great need for both pre-service and in-service education in programming for mathematics and technology teachers arose, and the Agency subsequently arranged for such courses together with the universities.

There is generally a lack of research about technology teachers and how they conceive of and teach programming, but the little existing research shows that technology teachers feel uncertain about how to teach programming, probably because it has traditionally been a marginal component in technology education e.g., Sentance and Csizmadia (2017); Vinnervik (2022); Webb et al. (2017). Those who taught programming before it was formally included in technology education were mostly computer afficionados that had learnt to program on their own, cf., Nouri et al. (2020).

When primary school curricula documents change, teacher education is also affected, as it has as its mission to educate teachers who can teach according to the current curricula. There is increasing interest in research on the implementation of programming in schools, but so far there has been a lack of didactic research in this area, and in particular research on teacher training. There is some didactic research on students' attitudes and understanding of technology, see for example Lee et al. (2020), but there is a lack of research on students' understanding of the role of programming in relation to the school subject of technology. Moreover, our knowledge of what prospective technology teachers learn about programming in the context of their teacher education programs is very limited. Student teachers' preconceptions about programming as well as what is the nature of programming in technology teacher education, is another under-developed area of research that could potentially shed light on what knowledge components need to be focused on to improve programming teaching in technology (teacher) education.

3. RESEARCH DESIGN

3.1. Phenomenography

The choice of phenomenography as the method for this study is relevant in several ways. Not only is it the world of student teachers and how they experience it that will be investigated (their preconceptions), but the phenomenographic approach is also appropriate because one purpose of phenomenographic studies is to examine the different ways in which teaching can be experienced (Wood, 2000). In phenomenographic studies, a second order perspective is adopted, i.e., descriptions are made of how other people experience different aspects of reality. The focus of phenomenography studies is on a comparison of people's preconceptions of a phenomenon (Marton, 1981) and the phenomena of interest in this study is *teaching programming in technology*. The phenomenographic approach will be used also because it is based on the basic assumption that we humans experience phenomena in partly different ways depending on our previous experiences (Marton & Booth, 1997). That different people may experience the same phenomena in different ways can be explained by the fact that there are differences in what a person's mind discerns. Based on our previous experiences of the phenomenon, what is foregrounded or backgrounded differs from individual to individual (Runesson, 2006). However, there are a limited number of ways to experience the same phenomenon, according to Marton (1981). This study will highlight qualitatively different ways in which student teachers experience or understand *teaching programming in technology*, that is, their preconceptions of this phenomenon. In phenomenographic studies, the area of interest is the collective preconceptions of a phenomenon rather than the individual ones (Booth & Ingerman, 2002; Marton, 1981; Marton & Booth, 1997; Trigwell, 2006). The researcher makes interpretations and describes the preconceptions of the interviewees but it is not of interest who or how many have which preconception but all the preconception are collected in a data set for categorization (Trigwell, 2006).

Phenomenography will be used both as a theoretical approach and as a tool in the analysis process to develop categories of descriptions. The descriptions will be based on the collected preconceptions of the students and will include the aspects that the student's preconception to be salient. The categorization will attempt to show a difference, a variation, that is brought out in the descriptions as the categories are divided by clearly drawing boundaries for when preconceptions are similar enough to be brought together into one category and different enough to be in separate categories. The categories are of interest because they contain the different aspects that students discern of the phenomenon. The categories are also arranged hierarchically so that the categories contain more and less complex preconceptions (Marton, 1981). At the time of data collection, the students have completed their education in technology and therefore it is of interest to see what the students do and do not discern about the phenomenon of teaching programming.

3.2. Data collection

This study builds on a previous study where extensive data was collected using semi-structured interviews and the interview guide was split into two parts. The interview guide included both questions about how the student teachers themselves experience programmed technological artifacts and how they see their future teaching of programming in technology. The latter will be focused for the data analysis in the present study. The student teachers interviewed were eight in number and volunteered to participate in the study in a sample of 30 students from two universities. In the data collection, pictures were used to start the conversation and the pictures depicted four everyday artefacts: dryer, traffic light, keyboard, and elevator. The artefacts have been chosen so that they are familiar to the students but also with the idea that they can be controlled by programming. In addition, these four artefacts can be linked to technical systems, which is an important part of the technology subject in primary schools.

The interview has followed a characteristic phenomenographic structure i.e., the interview guide has been used to support the conversation and the respondent's answers have guided the direction of the conversation. The interviewer's task has been to keep the focus on the phenomenon throughout the interview by repeating the respondent's answers and asking if they would like to elaborate on their answers or add further. Examples of questions that have been asked are "What do you think pupils need to know about programming in technology?", "What is important that we teach them?" and "What is important that you convey to pupils regarding programming in relation to technology?". The interviews were conducted via Zoom with an associated recording function. Each interview has had a time allocation of approximately 45-50 minutes. Each participating student was informed about the aim and design of the study and consented to participate. The responses were anonymised, and data is managed in accordance with the General Data Protection Regulation (GDPR).

3.3. Method of analysis

The analysis will be phenomenographic, which means that the researcher seeks a deeper and more multifaceted understanding of student teachers' preconceptions of the phenomenon in focus, in this case teaching programming in the technology subject. The analysis of the transcripts from the individual interviews began through repeated read-throughs where the researcher, here the first author, delved into the material to find the different preconcepted ways that exist around the phenomenon. This material constitutes a "pool of meaning" which is formed from the researcher first adopting a more open attitude to the material to gradually become more focused (Wood, 2000). Within the "pool of meaning", similarities and differences in preconceptions are then sought to make an initial grouping where it is a matter of trying to discern a variation, or difference, between the respondents' preconceptions. While the first groupings are being made, all three researchers discuss and together question the groupings. Once similarities and differences have been grouped, a description of what constitutes the differences and similarities found between the groups is made to know whether the groups should be merged or new groups created. The goal of the analysis process is to consistently identify the qualitative differences in student teachers' preconceptions when they describe teaching programming in the technology subject.

3.4. Validity of the study

We ensure validity of the data analysis in a number of ways. First, we have chosen to include excerpts of the collected quotes which show answers received in the semi-structured interviews; we have specified the overall questions in the text above, but the follow-up questions have varied depending on the informant's response. In the analysis process, we have also continuously tested the categorisation on fellow researchers. As we are not yet sure that saturation has been reached in our data material, we will continue to collect data.

4. TENTATIVE RESULTS

The analysis yielded a result in the form of categories containing descriptions of students' already existing preconceptions of teaching about programming in the technology subject. In the initial analysis phase, there emerged three tentative categories that are qualitatively different from each other, describing student teachers' preconceptions of teaching programming as: 1) an understanding of a language and/or a tool, 2) an understanding and use of a language or a tool to solve technological problems and as 3) a way of understanding and describing a technological environment.

4.1. Category 1: An understanding of a language and/or a tool

In this category the student teachers describe the use of tools, codes, algorithms as the main purpose of teaching programming in the technology subject. It is the structure of building a code with instructions and consequences of these instructions that are focused. They mention coding activities and practical work with the aim of becoming aware of the instructions. The following quotes from Daniella and Frida illustrate their descriptions of a sense of closeness to the practical aspects:

> Daniella: [...] so how to start it on the computer, how to use these commands, how to twist and turn so you get comfortable using it.
>
> Frida: It's a lot of following instructions and doing it from the top down and this way, […] for example, you're going to guide your friend and give instructions, or you're going to write down an instruction and then the other person will try to follow it, for example, draw something after the instruction.

4.2. Category 2: An understanding and use of a language or a tool to solve technological problems

In this category student teachers describe the teaching as something that includes the use of programming when solving technological problems. Programming could make things happen and fulfill wishes. Unlike the previous category, programming does not just become a code or a language, but in this category, it is made clear what programming can be used for in relation to the knowledge area of technology. The following quotes from Clara, Daniella and Hanna show a continued interest in practical engagement, while also emphasizing the necessity of problem solving:

> Clara: […] and technology education is largely about, well, how should I put it, identifying needs, and perhaps finding solutions to those needs, and it's quite challenging nowadays to find solutions to needs if you don't have programming skills.
>
> Daniella: It's not just about coding on the computer, it's something that exists around us, and I can use programming to turn on a light or I necessarily have to use some form of programming if I want to control a lamp or a stove or a dryer. […] But I think that when you program something, it's because it's meant to be some kind of tool, like you want to see something, you want to cook something, you want to dry something. It has a purpose, and that purpose is what belongs to technology. It's not just the fact that it's programmed that makes it technology, but it's what comes after, in a sense.
>
> Hanna: […] programming in technology is more like we program, for example, [...] carousels, making carousels that make them spin and stuff. It's about making things work, you know. So, in technology, it's like, it's a bit more computer-oriented, in a way.

4.3. Category 3: A way of understanding and describing a technological environment

In this category, the teaching of programming is contextualised as a part of society and thus other aspects than in the previous categories where it was the tools, instructions and practicalities that were in the foreground. Now in this last category, tools and coding are clearly in the background and in the foreground, there is instead a system perspective that has not been visible in previous categories and where programming is instead seen as part of a larger whole, a human-built, technological environment. Björn and Daniella describe this by highlighting several components where they describe teaching that deals with consequences for decisions and actions:

> Björn: To understand that something is happening behind the scenes. There's a reason why the lights turn off in the school corridor when no one has been there. It happens automatically, and it's programmed to do so. [...] Many things can be done to maybe save electricity or save water, and it can also contribute to sustainability thinking. Because I think many students are very concerned about that nowadays. And through technology and programming, there are great opportunities to address those concerns.
>
> Daniella: No, but what I mean is that everything, you learn, it's something that gives you power over your life and how you relate to society. And given that we have a lot more technological gadgets, we also need to have more knowledge about them and how they work so that we can engage with society and its structures. […] So that they can see that programming exists all around, it's in the traffic light when I go to school, what would happen if something went wrong and what would be the effects in a larger context?

5. DISCUSSION

This paper presents a study with the aim of describing student teachers' preconceptions about teaching programming in technology once they have completed the technology course in teacher education but before they implement it in actual classroom settings. From the preliminary results presented above, it appears that there is a variation in how much of everyday life and reality – the human-built environment – the student teachers put into their descriptions. Some talk about teaching code, command, and function with only some everyday connection while others talk about how important it is to have a sense of reality when teaching programming in technology. To better clarify the variation in this study and to achieve saturation in the data material, we plan to conduct focus group interviews with student teachers. An in-depth continuation of this study gives us the opportunity to clarify what the critical aspects can be for teaching programming in technology.

Although the student teachers, in this study may have understood the purpose of the technology subject and how it can be linked to programming, we cannot see that they have sufficient knowledge to be able to teach programming in technology themselves as our results show that only in the last category the students mention conceptions of programming that can be linked to a systems perspective. Teaching with a systems perspective has been shown to be important to increase the understanding of programmed artefacts (Cederqvist, 2020; Perez & Svensson, 2023) and an understanding for the content to be taught is important for future teachers. Critical aspects specific to teaching programming in technology can form the basis for further studies more focused on lesson design.

In summary, the description of student teachers' understanding of programming in technology provides an overview of the knowledge that needs to be developed in student teachers to give them a good foundation for teaching programming in technology.

6. REFERENCES

Booth, S., & Ingerman, Å. (2002). Making sense of Physics in the first year of study. *Learning and Instruction*, *12*(5), 493-507.

Cederqvist, A.-M. (2020). Pupils' ways of understanding programmed technological solutions when analysing structure and function. *Education and Information Technologies, 25*(2), 1039-1065.

Lee, K., Courtney, M., McGlashan, A., Neveldsen, P., & Toso, M. (2020). Initial teacher education students' perceptions of technology and technology education in New Zealand. *International Journal of Technology and Design Education*, *30*(3), 437-458.

Marton, F. (1981). Phenomenography—describing conceptions of the world around us. *Instructional science*, *10*(2), 177-200.

Marton, F., & Booth, S. A. (1997). *Learning and awareness*. Routledge.

Nissen, J., & Stenliden, L. (2023). How computers entered Swedish classrooms: The importance of educating digital citizens. In J. Hallström & M. J. d. Vries (Eds.), *Programming and Computational Thinking in Technology Education: Swedish and International Perspectives*. Brill/Sense.

Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry, 11*(1), 1-17.

Passey, D. (2017). Computer science (CS) in the compulsory education curriculum: Implications for future research. *Education and Information Technologies, 22*, 421-443.

Perez, A., & Svensson, M. (2023). Student teachers' experiences of programmed technological artefacts: range of understanding and ideas for development. In J. Hallström & M. J. d. Vries (Eds.), *Programming and Computational Thinking in Technology Education: Swedish and International Perspectives*. Brill/Sense.

Runesson, U. (2006). What is it possible to learn? On variation as a necessary condition for learning. *Scandinavian journal of educational research*, *50*(4), 397-410.

Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies*, *22*(2), 469-495.

Skolverket. (2017). Läroplan för grundskolan, förskoleklassen och fritidshemmet 2011: reviderad 2017. In: Skolverket Stockholm.

Trigwell, K. (2006). Phenomenography: An approach to research into geography education. *Journal of geography in higher education*, *30*(2), 367-372.

Vinnervik, P. (2022). Implementing programming in school mathematics and technology: teachers' intrinsic and extrinsic challenges. *International Journal of Technology and Design Education*, *32*(1), 213-242.

Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2017). Computer science in K-12 school curricula of the 2lst century: Why, what and when? *Education and Information Technologies*, *22*(2), 445-468.

Wood, K. (2000). The experience of learning to teach: Changing student teachers' ways of understanding teaching. *Journal of curriculum studies*, *32*(1), 75-93.