Computing at School in Sweden – Experiences from Introducing Computer Science within Existing Subjects

Fredrik Heintz, Linda Mannila, Karin Nygårds, Peter Parnes, and Björn Regnell

Linköping U., Linköping U., Sjöstadsskolan Stockholm, Luleå U. of Tech., Lund U., Sweden
fredrik.heintz@liu.se,linda.mannila@abo.fi,karinnygards@gmail.com,
peter.parnes@ltu.se,bjorn.regnell@cs.lth.se

Abstract. Computing is no longer considered a subject area only relevant for a narrow group of professionals, but rather as a vital part of general education that should be available to all children and youth. Since making changes to national curricula takes time, people are trying to find other ways of introducing children and youth to computing. In Sweden, several current initiatives by researchers and teachers aim at finding ways of working with computing within the current curriculum. In this paper we present case studies based on a selection of these initiatives from four major regions in Sweden and based on these case studies we present our ideas for how to move forward on introducing computational thinking on a larger scale in Swedish education.

1 Introduction

Computing is no longer considered a subject area only relevant for a narrow group of professionals, but rather as a vital part of general education that should be available to all children and youth. The December 2014 issue of ACM Inroads featured a special section on early computing education. The articles highlighted several questions related to the when, what and how of introducing computing prior to university level [1].

In October 2014, the European Schoolnet published a report on the current status of computing at schools in 20 European countries [14]. According to the report, a majority of the countries are introducing computing (or programming) at primary and secondary level, either as a subject on its own (e.g. England), in specific IT courses (e.g. Belgium and Estonia) or as an interdisciplinary strand in other subjects (e.g. Italy and Finland). In several countries, this work is, however, still in a very early phase, and generally there is no clear consensus on what computing education at primary and secondary level should entail or how it should be introduced into the education system.

Since making changes to national curricula takes time, people are trying to find other ways of introducing children and youth to computing. In Sweden, several current initiatives by researchers and teachers aim at finding ways of working with computing within the current curriculum. While focus in several countries is put on programming, we believe this to be too narrow focus. If you only focus on programming and code, you risk missing out on general and useful skills such as dividing problems in smaller parts, solving problems in creative ways, finding patterns, thinking logically, designing algorithms, working in a structured manner, making generalisations and finding models.

Similar to many others (e.g. [2, 16]) we use the term "computational thinking", rather than coding or programming, to refer to this set of skills and practices. Teachers already engage in several of these practices in their teaching, even without knowing it [7]. By making explicit what teachers already do, we can reduce the feelings of threat to teachers who may feel that they need to learn a lot of new things, many of which can be perceived as technically advanced.

The aim of this paper is to give an overview of some of these initiatives to give examples of how teachers can start working with computational thinking in their classroom, without any official policies requiring them to do so. We start by giving a brief overview of the Swedish school system and the current status of computing in K-12 education. Next we present a set of national and regional initiatives. The paper is concluded by some reflections on the current work as well as ideas for future directions.

2 Overview of Computing in Swedish Schools

The Swedish education system is divided into compulsory school (grades 1-9) followed by upper secondary school (grades 10-12), which offers 12 vocational programmes and 6 programmes preparing for further studies. The national curriculum for compulsory education does not include computing or programming. One of the programs of the general education strand is focused on Technology (*teknikprogrammet*), which also includes courses covering various aspects of computing. This means that only a limited number of students take such courses at upper secondary school. All students do, however, get to use technology, i.e. computers and applications, in their studies.

The situation in Sweden is hence similar to that of many other European countries, where digital competence has come to be seen as something of a synonym for basic digital literacy. Whereas an increasing number of countries are revising their curricula and policy documents to also include computing aspects, Sweden has a rather new curriculum, which most likely will not be updated within the near future.

In 2012, the Swedish government established a committee with the task of giving recommendations and guidelines for how Sweden can – and should – benefit from the digitalisation. In a report published in March 2014 [3], the committee emphasises the need for an additional focus on digital competences in national curricula. One concrete recommendation is for programming to be introduced as a cross-curricular element in already existing subjects.

3 Bebras

Since computing has not traditionally been part of general education, other, more informal, approaches have been introduced in order to cover the same ideas. Contests is one example, where, for instance, programming competitions have been regularly arranged. These are, however, aimed at talented students and are as such not a suitable channel for introducing computing to students at a larger scale. To mend the situation, Bebras (beaver in Lithuanian) was initiated in Lithuania in 2004 as a contest suitable for all children and youth aged 8-19, inviting them to work with motivating and playful tasks related to computing and computational thinking. The contest is organised online

and participants are divided into five age groups (Mini, Benjamin, Cadet, Junior and Senior). Over the years Bebras has grown into a large annually arranged international contest, having about 900 000 participants from 35 countries in 2014.

Sweden has organised the contest twice in addition to a test round in 2012. In 2014 the contest had 18 problems in each category that should be solved in 40 minutes. In 2013 there were 15 problems in each category that should solved in 45 minutes. The reason for the change was to use both the same problems and the same rules as Germany, Lithuania and Finland in order to be able to compare the results between these countries. Unfortunately we had to replace one of the German tasks since it had already been used by Sweden and Finland earlier. Based on the results from the 2014 contest the best boys and girls in each category was invited to Linköping University to participate in an onsite final. The event was greatly appreciated by the participants even though they felt that it was not necessary to have another contest, especially not those that had won their categories since they could only do worse. In its short existence, the participation has grown dramatically from 1869 in 2013 to 7059 in 2014. To our delight, the number of girls participating has also increased from 37% in 2013 (695 out of 1869) to 44% in 2014 (3126 out of 7059). The detailed figures for 2014 can be found in Table 3.

Category	Participants	Teachers	Schools	Cities	Boys	Girls
Mini	1148	61	42	37	565	583
Benjamin	1499	54	51	41	767	732
Cadet	2045	62	60	43	1116	929
Junior	1701	39	37	31	924	777
Senior	666	22	22	20	561	105
TOTAL	7059	189	150	92	3933	3126

4 Activities in Linköping

Since 2013 the Department of Computer Science at Linköping University (IDA) has been actively involved in supporting computational thinking at all levels of the education system.

CoderDojo: To make programming available to as many people as possible, IDA has been active in starting a voluntary programming club based on the international CoderDojo concept (`coderdojo.org`). This is a fun and creative way of supporting kids between 7 and 17 in learning to program. The basic setup is to invite kids to learn to program supported by mentors. A mentor is someone with some experience of programming. The kids are given some hints on what they can do and then they use their creativity to find their own way to learning programming with some nudges and help from the mentors. It is not a class and we try to limit the similarities with school activities. We have encouraged everyone to start with the Hour of Code (`code.org`) and then explore programming through the use of Scratch [15], a visual programming language from MIT. More advanced kids have started with JavaScript using CrunchZilla. The first CoderDojo in Sweden was started in 2012 in Malmö. CoderDojo Linköping

has arranged CoderDojos every second week since January 2014, with about 20-30 kids each time. Most kids are around 7-10 years old. Among those about 60% are boys and 40% girls. Among the older kids almost all are boys. In the spring of 2015 a national organization, CoderDojo Sweden, was started which supports the growing number of CoderDojos around Sweden. There are currently about 15 active CoderDojos in Sweden.

Pupil and Teacher workshops: To encourage schools that arrange local programming profiles or have pupils that are interested in programmering IDA has arranged programmering a number of half-day workshops. These workshops usually start with an introductory lecture, followed by two lab exercises one involving programming the Nao humanoid robot using the graphical programming language Choregraph and one involving programming generative art in JavaScript, the workshop is concluded with a popular science presentation of research in AI and robotics. We have arranged roughly 15 workshops the last 18 months. We have also arranged a number of workshops to educate and inspire teachers to start using computational thinking in their classes. This also included two special events, one for politicians in Linköping and one for the general public at the big political event Almedalen.

Computational thinking in Swedish basic education: is a project whose purpose is to propose an approach to introducing computational thinking in Swedish basic education and to develop teaching materials for introducing principles and practices from computing as well as computational thinking in a variety of existing subjects (Swedish, Math, Sciences, and Technology). The project involves both researchers at IDA and teachers from Linköping schools and is funded by VINNOVA.

The aims of the project are to:

- write a scientific report on computational thinking and how it can be introduced in Swedish schools,
- develop lesson plans and activities together with teachers, which can be used to introduce computational thinking in different subjects,
- organise teacher training on computational thinking and related didactical aspects, and
- empirically evaluate the lesson plans and activities together with pilot teachers.

As we see it, introducing computational thinking at lower levels of education calls for a dialogue between teachers, teacher trainers and researchers in computer science education. As researchers, we do not have enough insight into the everyday practice in schools nor the long experience of teaching children and youth that teachers have. On the other hand, since computing is not part of the curriculum, most teachers do not have any background in computational thinking or programming, and hence need support in order to understand what computational thinking is, why it is important, and how it can be integrated in their teaching practice. Through the steps above, in particular the three latter ones, we aim at building a model where teachers bring their own expertise in terms of, for instance, subject knowledge and teaching experience, whereas researchers in computer science education can show how principles and ideas from computing can be used in a relevant way in various subjects. The goal is for the model to empower teachers through a continuous dialogue and concrete collaboration.

Discussion. The experience from the project has so far been positive. Teachers appreciate that we take their practice and their students' learning (what is in it for them?) as the starting point, not a given technology or tool (what can we do with x?), which is commonly the case when talking about IT at schools. Based on our discussions with teachers, computational thinking seems to be a more suitable concept to start with than programming or computer science, which are considered both too technical and narrow. In the pilot study 10 teachers were involved. They participated in three half-day workshops and did at least one activity related to computational thinking with computers and one without (unplugged) in their classes. In the first workshop the concept of computational thinking was introduced. In the second workshop concrete activities in the teacher's subjects were discussed and each teacher committed to performing one activity with computers and one without. In the third and final workshop the teachers reported their experience from running the activities. This showed that it was possible for teachers from grade 2 to grade 9 to perform activities related to computational thinking within their subjects with very limited training.

As a direct consequence of the project, IDA and the city of Linköping are discussing ways to extend the collaboration and introduce computational thinking to a much wider selection of schools.

5 Activities in Lund

Lund University is engaged in activities related to computing at school through its science center [6]. The general goal of the science center is to reach out to pre-university education and help to increase the interest in engineering education among the youth in southern Sweden. The science center opened in September 2009 and has since then had more than 166 000 visitors of all ages (as of January 2015). Visitors during weekdays range from school groups to company events and training. The center also takes bookings for a variety of different events and celebrations. It is open to the public on weekends and school holidays, when everyone is invited to try out interactive experiments, attend a show, or see an exhibition.

Before 2012, our science center experiments focused on areas such as physics, chemistry, and electronics, but the area of computer science was lacking, and to fill this gap, a project called "Programming for Everybody" (subsequently denoted PfE) started in 2012 funded by the Engineering faculty and hosted by the science center at Lund University with project members from the department of Computer Science. The main goals of the PfE project are to: (1) develop programming experiments for visitors with particular focus on groups visits from schools, and (2) to develop teacher training so that pre-university schools for all ages can help young learners to discover the excitement and importance of computer programming.

Development of an open pedagogical concept. Within the PfE project, a pedagogical concept for teaching programming is developed. The concept is targeting young learners of age 7 and upwards as well as their school teachers, with the only pre-requisite of being the ability to read from a computer screen and use a keyboard and mouse. When selecting a platform for the pedagogical concept, including language and integrated development environment, these criteria were stipulated:

1. The platform should be free to install, available as open source, and run on Linux, Windows and Mac.
2. The user interface should be available in Swedish, and the programing language should allow Swedish identifiers in the code.
3. The platform should offer 'real-world coding', i.e. it should be based on a modern and professionally used programming language that provides access to general-purpose code libraries.

Based on these criteria, the programming environment Kojo [4] and the programming language Scala [8] was chosen. Both Kojo and Scala are free and open source. The Kojo project is lead by Lalit Pant at Kogics in India [12] and the Scala project is lead by Prof. Martin Odersky at EPFL in Switzerland [13]. Prof. Björn Regnell at Lund University has contributed with translations of Kojo's turtle graphics API and graphical user interface to Swedish, as well as software development, testing, and project sponsoring.

The PfE pedagogical concept includes a series of *programming challenges* [11] that cover a progression of programming concepts including sequential execution, repetition, abstraction, parametrized abstraction, and nested abstractions. The PfE challenges have been iteratively developed based on feedback from kids and teachers.[1]

The PfE programming challenges are rooted in a contructionist approach [9] through a Swedish turtle graphics API in Scala. The initial challenges are based on reading and tweaking a given, worked example with a code snippet and hints, to enable learners to quickly grasp syntax as well as concepts, in line with suggestions that worked examples may (according to cognitive load theory) be effective in learning how to program [5].

Programming activities for science centre visitors. Since the PfE project started, we have had more than 10000 young learners have experienced our programming experiments, based on the PfE challenges with Kojo and Scala. The visits are mainly school classes that try out different experiments in the science centre, where programming is one station. The school classes are divided into smaller groups of around 10 students, often working in pairs to solve the PfE programming challenges [11], while one or two instructors are providing guidance. School class group session range from 20 to 45 minutes and focus on drawing pictures with turtle graphics, while learning about sequence, repetition and abstraction using procedures without and with parameters.

Programming education for teachers. More than 100 teachers have passed our programming courses comprising 2-3 half-days with assignments in between, to try out programming in class using Scala and Kojo. The teachers learn programming with the help of the PfE programming challenges, which they then try in their classes with young learners. Teachers then share their experiences with each other, including new challenges that they develop in relation to their subject curricula.

Discussion. Teachers that attend our courses are teaching different subjects at primary school, ranging from maths and science to language and sports, but also practical skills such as needle- and woodwork. Many teachers that attend our courses are at first uncertain if they will manage to run a programming class within their subject, but when they experience how their students engage in the challenges and with enthusiasm discover the playfulness and joy involved in creating software, many change their mind and

[1] The challenges are available with a Creative Commons Attribution-NonCommercial-ShareAlike licence [11].

see the opportunities. A major opportunity for the future is to create an active network of primary school teachers to enable sharing and inter-scholar learning of how to integrate programming in the existing curricula. The teachers that have taken our courses at the LTH Science Centre realize the potential, but they want more subject-specific study material and also assessment models so that the student learning outcome from programming projects can be assessed in relation to the learning goals of their subject.

6 Activities in Luleå

Starting in 2013 a collaboration between Luleå University of Technology, LTU and The Municipality of Luleå [2] started around how to increase awareness of computational thinking and digital literacy in Luleå's schools. The collaboration started informally and was made more formal when funding was secured during 2014 for several projects. It was early identified that no single solution would be enough to inspire and educate the more than around 1900 teachers and principals but rather that several different activities had to be organised and implemented. It was also identified that the best approach was to test different activities to see which were the most appreciated. The activities organised include *Pedagogical Pubs with TeachMeet*, *Open Educational Workshops*, *Programming inspiration* and *Making in Schools*.

Pedagogical Pubs with TeachMeet: In an effort to inspire teachers and allow teachers to share their ideas, thoughts and progress several Pedagogical Pubs with TeachMeet were held during 2014 where invited speakers presented various topics related to ICT in Schools and followed by very short fire talks by teachers that wanted to share what they had done in their classrooms. Each of these gatherings had around 80-120 participants and they were open to the general public as well including teacher students from the university.

Open Educational Workshops – CS4HS Luleå 2014: The Open Educational Workshops were supported by Google via their CS4HS program during 2014. The goal of these workshops was to give teachers and principals a chance to meet and learn more about Computer Science and how it can be applied in grades 1-12. The workshops were focussed on giving concrete hands-on work with various tools including graphical programming like Scratch and Blockly via Hour of Code, testing electronics like Arduino and doing some more advanced Javascript programming. 5 workshops were held of which the first 4 were with computers while the final one was *unplugged* where the participants discussed how to apply their new knowledge in the classroom and also included a specific discussion on how to better include young females.

The workshops were held free of charge and over 80 persons applied of which 35 participants were selected based on geographical location as well as teaching level. The goal was to get a good mix between schools and teacher levels. The participants showed a real interest and really wanted to learn and also learn how to apply their skills in their classrooms and it was very rewarding to see how they immediately applied them the after the workshop was held.

[2] Many of the efforts described here were organised together with principal Agneta Hedenström from the Luleå Schools Antnäs and Måttsund.

Programming Inspiration – EU Code Week, Hour of Code and School Visits: During the autumn of 2014 two separate events around programming in schools were held. In October the EU Code Week was held and in Luleå 9 separate events were held (out of 90 in Sweden) where students got to try Scratch primarily at grades 2-6. The EU Code Week was held all over EU at the same time with more than 3000 events during one week. During December 2014 the global Hour of Code week was held and in Luleå the efforts were focussed to December 9 when with the help of volunteers we were able to reach more than 1000 students via more than 50 separate programming hours where students got to try the Hour of Code learning environment or Scratch. Besides these larger organised events a number of schools visits were held where programming and making was presented to students at various levels in the school system.

Making in Schools – Luleå Makerspace and the Skaepiedidh Project: Another approach has been through *Making* in schools where the students have gotten to test various technologies. This has mostly been done through collaboration with the Luleå Makerspace which is a non-profit organization with the goal to promote ICT and technology interest to persons of all ages. It was early identified that the teachers only needed a small amount of *pushing* in the right direction to get started with making and programming in school, but at the same it was hard to get a scalable solution as much relied on personal contact to inspire them. Thus, the Skaepiedidh project was created between LTU and the Luleå Municipality with national funding from VINNOVA with the goal to create an online system for doing the inspiration online where teachers could exchange recipes for making and programming in schools, comment on these recipes, remix them and spread their implementations further. More thoughts on Making in schools can be found in [10].

Discussion. The various efforts presented here have been very appreciated by those participating which is shown via that the teachers actually get started with computational thinking in schools as well as that they want more help and more inspiration. The Luleå Model is a combination of several different efforts as to reach as many as possible no single effort is enough. At the same time it has been identified that only a smaller portion of the 1900 teachers and principals are reached and some of the schools are not present at all and thus in turn a large portion of students are not reached unless a more formal mandate by the government and/or the schools leaders is made. During 2015, the efforts will continue but with more of a focus trying new things as well as influencing politicians and school leaders through local and national efforts. Several workshops and a larger educational conference is planned as well with the goal to spread good examples and get people to meet and talk and learn from each other as we foresee peer learning as one of the most important parts of this campaign.

7 Activities in Stockholm

The Stockholm region is a large area, with over two million inhabitants, which makes collaboration around topics, like computational thinking, challenging to coordinate. Within the region, some municipalities have been more progressive about getting their teachers to start using computational thinking in the classroom, even though still in small scale. During 2014 computing and programming became popular buzzwords. Consequently,

programming has been introduced at several schools in the region by passionate teachers. In February 2015, people representing the City of Stockholm, the Stockholm Chambers of Commerce and the company Spotify, wrote an article in Dagens Nyheter about a larger effort to set up pilot schools to teach programming in elementary schools in Stockholm. This is now under preparation. Teachers and principals from schools already working with computational thinking, are consulted to find best practice in the implementation to all 265 schools within the City of Stockholm. This is so far the largest intervention in Sweden and will be of big importance as an example for the rest of the country.

Programming Inspiration – EU Code Week, Hour of code: During the 2014 European code week, Academedia, the largest group of independent schools in Sweden, had students from all secondary schools teach 5th graders about programming. At least 20 schools within the Stockholm area participated. The documentation of participants haven't been that precisely. This is something to improve this coming year. As a total 10 schools within the area have reported participation during Hour of Code in December.

Organizations:

- Coder Dojo was the first code initiative in Stockholm in 2012. Two days a week, CoderDojo offers open programming sessions for kids, at a library in the city center, free of charge and there is no need to bring a computer.
- The National Computer Society, Dataföreningen, has organized several events for children and also teachers and other adults interested in coding with Scratch. The goal is to use the members of the society as mentors to the teachers who want to try coding in class. Dataföreningen also has RaspberryPis to bring to schools if they don't have enough computers. There are two networks within Dataföreningen, one for anyone interested in contributing to children learning more about programming, and one for teachers.
- Kosmosklubben, an NGO offering 10 different programming courses every week and 9 classes of robotics. There is also a special program for high school girls from suburbs, to get them more interested in code and IT.
- Kids Hack Day, is an after school program, where kids can pay to participate in coding and maker activities. They have had three groups during the semester. The cost is rather high but includes all kinds of maker supplies.
- Kodcentrum: Is a non-profit organization based in Stockholm offering children and youth hands-on experience in programming through an after school program. This is in contrast to CoderDojo more like a course were the children follow a prepared material during a whole semester. Kodcentrum has activities in Göteborg, Linköping, Stockholm, Sundbyberg, Umeå and Uppsala.

Tekniska Museet (Museum of Technology), organizes children's hack sever times during the year. They have both courses for beginners and more advanced programmers. Tha hack club uses Scratch, but the museum also offers robot programming with LEGO Mindstorms. In September a new part of the museum will open, called the Mega mind, with one maker space and one coding space.

Initiatives for girls: To try to get more girls interested in engineering and IT, separate groups are more and more popular. Here are some initiatives.

- Geek Girl Mini, an after school program for girls in 5th grade. Two schools in Stockholm has been involved. One in an upper middle class-area and one in an all immi-

grant area. The second group was the largest and most of the 21 girls showed up every time, even on holidays. It was a big success.
- Girls Code, after school program at Mälarhöjden for girls in grades 7-9.
- Tech Girl, an initiative from the company Valtech. Two groups of girls in 6th grade have been taught programming in scratch for one semester each. Volunteers from Valtech have been running the workshops.
- MakerTjej, an organization with a mission to get more girls into the maker movement.
- Tjejhack, an organization with a mission to get more girls into the gaming business and also to change the rather raw climate in the gaming community.
- Teklafestivalen, a festival about music and tech, organized by the Royal Institute of Technology and the artist Robyn. The evens was a huge success, with over 2000 applications for the 200 tickets. The participants were chosen to get a variation in age and geographic background. Companies as Valtech, Google and Spotify arranged workshops during the day. Hopefully the event will be annual.

Teacherhack: Since programming is not part of the national curriculum, a group of teachers started to search for ways to integrate different aspects of computational thinking and computing in every subject within the current curriculum. On the website teacherhack.com there are texts explaining all 20 subjects in the curriculum from a digital aspect, over 30 different lessons and comments from around 30 teachers representing different subjects. The website is not yet officially launched, but already it has some 200 visitors each day. The goal is for Teacherhack to become a valuable resource for teachers wanting to include computational thinking and programming in their classrooms.

Sjöstadsskolan: At Sjöstadsskolan in Stockholm, programming has been taught since February 2013. During the first year, programming was introduced only to 50 students in 5th grade, but 2014-2015 all 800 students, including preschoolers (from 3 years old) have tried programming or computational thinking in some form.

- Preschool (3-6 years): Bee-Bots for creating simple programs after step-by-step instructions.
- 1st grade: Various games for understanding code ”unplugged”. iPad apps, such as BeeBot and Kodable. Code.org.
- 2nd grade: Games for understanding binary code and the Internet. Bee-Bot programming. Participated in Bebras.
- 3rd grade: Code.org and creating mazes for the BeeBot using paper, pen and ruler.
- 4th grade: Investigating geometry through the programming environment Kojo. Constructing real-life computer games in the school yard.
- 5th grade: Participating in a study on teaching Cartesian coordinates, comparing traditional instruction to introducing coordinates using Kojo. Participated in Bebras.
- 6th grade: Students already had 1.5 years of programming experience. Programming as an extracurricular activity.
- 7th grade: Code.org. Participated in Bebras.
- 8th grade: Code.org. Students learned how to use block based programming for creating visualizations in biology. LEGO Mindstorms were used in Technology class.
- 9th grade: Code.org.

– Special need school: Two students (aged 6 and 16) with severe autism have been practicing with BeeBot.

Research: Sjöstadsskolan has conducted three pilot studies about programming in school. The first pilot study was conducted on introducing computational thinking through programming in a Swedish course [8]. The initial motivation was to investigate the use of programming when teaching 4th graders (9-10 year olds) how to write instructions and reflections in an authentic way. The children were given assignments to be solved in pairs or small groups. A shared blog was used to distribute the results, which also made it possible to discuss and compare solutions. The same methodology was used in the 5th grade as well (10-11 year olds), then including more aspects of computational thinking. The second study was conducted with computational thinking as a separate subject. The conclusion was that the students found it fun and interesting, but not as part of the rest of the school day. The transfer from computing to the other lessons, just didn't happen. The third study, done in cooperation with Stockholm University and City of Stockholm department of education, was focused on using programming as a tool in teaching math. The study focused on Cartesian coordinates and the result will be presented at a conference for teacher's research in November 2015.

Discussion: Based on the initial pilot study and discussions afterwards we identified several ways in which computational thinking can be used in language courses. First, programming languages are also languages, sharing for instance the notion of grammar (syntax) and meaning (semantics); hence these can be discussed in traditional language courses. Moreover, computational thinking concepts can be introduced in a natural way when talking about languages. For instance, abstraction through nouns and verbs etc., problem decomposition through a grammar that breaks down the problem of describing classes of sentences into smaller problems; patterns through rules in a grammar; programming languages used for implementing rules in a grammar; introduction of algorithms and instructions through conditions, loops and iteration used in, for instance, storytelling. The second study implies that it's not for the best to teach computational thinking as a separate subject. To better use the transfer effect, we should integrate it in other subjects. The result from study number three, are yet not presented, but it seems to be a reasonable idea to use a program environment like Kojo, to teach Cartesian coordinates. As the capital and the largest city in Sweden, the path that the City of Stockholm will choose, will probably be of big importance for the implementation of computational thinking in schools.

8 Lessons Learned and Ways Forward

All the initiatives described above show that computational thinking and programming can be introduced and support learning at lower levels of education, although such elements are not part of the curriculum. There is nothing in the national curriculum that prevents schools and teachers from working with computational thinking, but also nothing that encourages them to do so. Since schools and teachers are already under a lot of pressure from many directions, few do anything. Those that do, however do very good things. In order for these initiatives to spread and become commonplace throughout the

nation, there is a need for a national initiative to introduce computational thinking in the school system. There are indications that such initiatives might be coming.

Another major challenge is teacher training. Teacher education programmes do not commonly include computing or computational thinking in their curriculum and consequently pre-service and in-service teachers lack insight and skills in these areas. We are therefore actively seeking a dialogue with the universities providing teacher education, but so far they have showed little interest. This calls for a dialogue between teachers, teacher trainers and researchers in computer science education.

Based on our current activities and lessons learned, we are currently working towards the following goals:

1. Establishing the term "Datalogiskt tänkande" as the Swedish term for computational thinking, implying the general skill set that can be trained through programming and that can be used in an interdisciplinary way throughout the curriculum.
2. Engaging as many schools as possible in Bebras, in order to stimulate the interest for computational thinking.
3. Supporting informal activities such as CoderDojos,and Maker Spaces which play an important role in giving students hands-on experience with, for instance, programming.
4. Collaborating with municipalities wanting to introdue or at least test computational thinking at a larger scale.
5. Supporting teachers in developing concrete example activities and lesson plans on introducing different aspects of computational thinking in a variety of subjects.
6. Designing concrete suggestions for professional development for teachers on computational thinking, for instance, in the form of a nation-wide MOOC supported by local study groups.
7. Engaging in continuous discussions with teacher education programmes in order to introduce at least one compulsory course on computational thinking for all pre-service teachers.
8. Developing suitable means for assessing computational thinking, for instance based on Bebras activities.

We have in this paper only reported activities that we are directly involved, but there are several other initiatives in Sweden that focus on the introduction of programming and computer science to young learners. Given the awakening interest from politicians and education developers, we believe that computational thinking soon will play a more prominent role in Swedish pre-university education.

References

1. ACM Inroads, December 2014.
2. Valerie Barr and Chris Stephenson. Bringing computational thinking to k–12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1):48–54, February 2011.
3. Digitaliseringskommissionen. En digital agenda i människans tjänst - en ljusnande framtid kan bli vår, 2014.
4. Kojo home page. `http://www.kogics.net/kojo`.

5. Raymond Lister. After the gold rush: Toward sustainable scholarship in computing. In *Proc. Conference on Australasian Computing Education (ACE)*.
6. Lund University Science Center home page. `http://http://www.vattenhallen.lth.se/`.
7. Linda Mannila, Valentina Dagiene, Barbara Demo, Natasa Grgurina, Claudio Mirolo, Lennart Rolandsson, and Amber Settle. Computational thinking in k-9 education. In *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference*, ITiCSE-WGR '14, pages 1–29, New York, NY, USA, 2014. ACM.
8. Martin Odersky, Philippe Altherr, Vincent Cremet, Burak Emir, Sebastian Maneth, Stéphane Micheloud, Nikolay Mihaylov, Michel Schinz, Erik Stenman, and Matthias Zenger. An overview of the Scala programming language. Technical Report, IC/2004/64, EPFL Lausanne, Switzerland, 2004.
9. Seymour Papert. *Mindstorms: Children, computers, and powerful ideas*. 1980.
10. Peter Parnes. Skapande och skaparkultur som drivkraft för kreativt lärande i skolan, Feb 2015.
11. Programming Challenges, Editor: Regnell, Björn. `http://fileadmin.cs.lth.se/cs/Personal/Bjorn_Regnell/uppdrag.pdf`.
12. Björn Regnell and Lalit Pant. Teaching programming to young learners using scala and kojo. In *LTHs Pedagogiska Inspirationskonferens*, volume 8, page 4. Lund University, 2014.
13. Scala home page. `http://scala-lang.org/`.
14. European Schoolnet. Computing our future. computer programming and coding - priorities, school curricula and initiatives across europe, October 2014.
15. Scratch home page. `http://scratch.mit.edu/`.
16. Jeannette M Wing. Computational thinking. *Communications of the ACM*, 49(3):33–35, 2006.