Off to new shores: Preparing Primary School Teachers for Teaching Algorithmics and Programming

Katharina Geldreich
Technical University of Munich
TUM School of Education
katharina.geldreich@tum.de

Mike Talbot
Technical University of Munich
TUM School of Education
mike.talbot@tum.de

Peter Hubwieser
Technical University of Munich
TUM School of Education
peter.hubwieser@tum.de

ABSTRACT

In many countries, the demands are getting louder to bring computer science education into primary schools. Curricula and teaching approaches are evolving and educators have to work their way into new topics. Many primary school teachers feel overstrained by these developments and the need for appropriate teacher training is rising. In this paper, we describe the structure and contents of an in-service professional development workshop for primary school teachers without any previous knowledge in computer science (CS) as well as first results of the pilot run with 40 teachers. Throughout the three-day workshop, the teachers get the chance to follow the students' path of learning by taking a primary school programming course themselves, engage intensively with the underlying algorithmic concepts through in-depth exercises and work on their own ideas for implementations in the classroom.

CCS CONCEPTS

• **Social and professional topics** → *Computer Science Education*; *Adult Education*;

KEYWORDS

Computer Science Education; Primary School; Teacher Training; Algorithmics; Programming

ACM Reference Format:
Katharina Geldreich, Mike Talbot, and Peter Hubwieser. 2018. Off to new shores: Preparing Primary School Teachers for Teaching Algorithmics and Programming. In *Proceedings of the 13th Workshop in Primary and Secondary Computing Education (WiPSCE '18), October 4–6, 2018, Potsdam, Germany*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3265757.3265783

1 INTRODUCTION

The discussion about the necessity of Computer Science (CS) and especially programming in childhood education is growing steadily [27][35]. While several countries have already introduced CS in their primary school curricula (e.g. the UK [6] and Australia [14]), Germany has not yet developed mandatory guidelines for how to deal with the new topics. Anyhow, various efforts are being made to give all children the opportunity to develop fundamental skills in programming and computational thinking and to investigate which teaching methods and contents could be appropriate for students in German primary schools [4][12][15][16]. This also raises the question of how we could prepare and train primary school teachers to cope with these new challenges. For most primary school teachers, CS is a new subject area which has been a part of neither their own school education nor their formal teacher education [13]. Since both content and pedagogical knowledge are necessary to teach the subject effectively, primary school teachers are facing a double challenge [9][23].

In this paper, we first give a short overview of a recently launched project that tries to counteract these difficulties. 40 teachers of 20 different primary schools in Bavaria (Germany) get the chance to participate in an in-service professional development workshop on *algorithms* and *programming*. They will then be accompanied and supervised for one school year in the implementation of the new topics. Afterwards, we will describe the theoretical framework, structure and contents of the three-day teacher training workshop in more detail. We also give an insight into the evaluation of the workshop and the experiences we have gained during the first two implementations of the training. Finally, we give an outlook on the further steps that will follow in the project.

2 CONTEXT

The teacher training is part of a project that is carried out over a period of two years and funded by the Bavarian Ministry of Education. The project investigates how primary school teachers can be prepared and supported to deal with the topics *algorithms* and *programming* in Bavarian primary schools. In addition, both the teachers' lessons and the students' learning progress are scientifically analysed and evaluated. To this end, various methods such as questionnaires, interviews and teaching diaries are used.

Based on an already evaluated programming course for primary school students, a total of 40 teachers receive a three-day training in which they get the opportunity to expand their computing knowledge. After the training, they are provided with additional online material as well as the possibility to seek further support if required. In the following months, the teachers are to gain initial experiences with the new topics in their own lessons. In this phase they are in close contact with the project team and can request support if they feel insecure. After this period, further workshops will take place in which the participating teachers can exchange their experiences and identify possible further training needs.

3 THEORETICAL FRAMEWORK

Following the methodology of design-based research [34], we took into account theoretical principles and prior research from various

WiPSCE '18, October 4–6, 2018, Potsdam, Germany

ACM ISBN 978-1-4503-6588-8/18/10…$15.00
https://doi.org/10.1145/3265757.3265783

disciplines, including not only computer science education but also developmental psychology, cognitive science and teacher education. This resulted in the following basic principles to which we aligned the design of the teacher training workshop (cf. [18]).

The workshop should

(1) enable active participation right from the start.
(2) facilitate the development of content knowledge as well as pedagogical content knowledge.
(3) allow teachers to understand the students' gain of insights.
(4) set focus on the fundamental algorithmic structures and principles of programming.
(5) show the possibility of cross-curricular teaching.
(6) be the first step in a continuing relationship between the primary school teachers and the university.

The workshop enables active participation right from the start. According to the constructivist theory that builds upon the work of Dewey [11], Piaget [25] and Bruner [7], learning is seen as an active, subjective process in which knowledge and meaning are constructed in interaction with the new topics. This approach can also be pursued in the context of computer science education and emphasizes the active participation of learners in problem-solving and critical thinking [3]. We do not only follow this approach with regard to the students, we also wanted to give the teachers the opportunity to actively deal with the topics in a variety of tasks. The theoretical input required for a deeper understanding should always be given in connection with these concrete tasks and applications.

The workshop facilitates the development of content knowledge as well as pedagogical content knowledge. In order to deal with algorithms and programming in class, teachers need not only a substantial knowledge of the subject but also a solid background in how to teach these topics to students [19][32]. Although we know very little about what this pedagogical content knowledge looks like in computer science and how teachers develop these skills [37], we can provide teachers with knowledge about the needs of students [19] and involve them in a continuous dialogue about the implementation of the new topics in their teaching [26]. In addition, the teachers should get to know various didactical approaches, activities and types of tasks during the workshop sessions [18].

The workshop allows teachers to understand the students' gain of insights. In the natural sciences, it is common for teachers to be given the opportunity to experience the students' process of *conceptual change* in order to assess its significance and to evoke it in the classroom [36]. We applied this concept in our workshop by giving the teachers the opportunity to take on the role of the students and try out various tasks and methods for themselves. They should get the opportunity to follow the students' learning process, sudden insights and also get an impression of the effectiveness of the different methods.

The workshop sets focus on the fundamental algorithmic structures and principles of programming. Teaching algorithms and programming is much more than teaching a particular programming language. Knowledge of a programming language is a necessary but far from sufficient prerequisite for learning programming [8]. Especially in primary school, it is important to be aware of the big ideas that we want the students to take with them from the lessons [1]. According to the Böhm-Jacopini theorem [5], teaching programming should cover three basic constructs: sequence, selection and iteration. The workshop should emphasize these algorithmic structures as well as generally applicable computational thinking concepts, e.g. algorithmic thinking, abstraction, decomposition [2].

The workshop shows the possibility of cross-curricular teaching. Especially if CS is not yet part of the mandatory primary school curriculum, most teachers are constantly considering how the new topics can be included in their current curriculum [28]. Therefore, teachers should be given enough time to link the new content to the existing curriculum and to identify places where they can integrate algorithmic thinking and programming. One benefit of using cross-curricular teaching is that the teachers are working with partially familiar ideas, which could give them more security in the classroom. Besides, the new CS topics can have a positive effect on the existing curriculum instead of just reducing the time available [2].

The workshop is the first step in a continuing relationship between the primary school teachers and the university. Research shows that coaching and ongoing support can facilitate the effective implementation of new curricula, tools and approaches by educators [24][28][29][31]. Moreover, it is more likely that teachers who receive coaching will apply the desired teaching practices and apply them more appropriately than those who receive more traditional teacher trainings [10]. In addition, a sustainable continuing professional development that offers a variety of opportunities to address the new issues will change teachers' teaching practices rather than episodic and fragmented teacher trainings [20].

4 WORKSHOP STRUCTURE

The workshop takes place right at the beginning of the project and provides a three-day introduction to algorithmics and programming. Since it is held in a center for in-service professional development, the participants can stay overnight and are provided with meals.

The topics of the three days are:

Session 1: Introduction to algorithmics and programming in the primary school context
Session 2: Engaging with basic algorithmic structures and programming in Scratch
Session 3: Own implementations at the teachers' schools

4.1 Session 1 *(4 hours)*

As an introduction to the workshop, the teachers get an insight into an already existing programming course for primary schools, which can be completed within three school days [16].

The course includes unplugged activities as well as working with the visual programming language Scratch [22]. At the end of the course, the students should understand that a device is following an

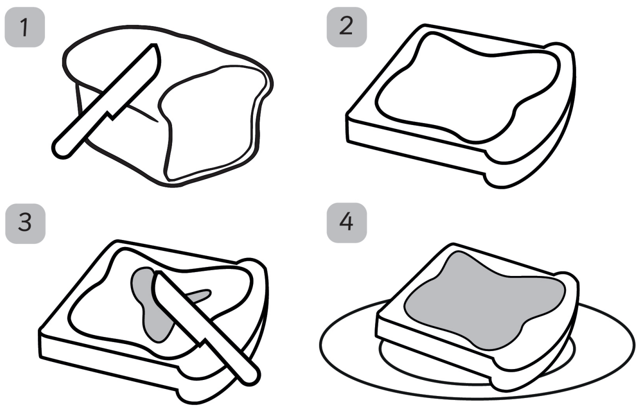


Figure 1: Pictorial representation of making a sandwich

algorithm that is implemented by programming the device. They should also get familiar with the process of testing and debugging a program and get to know the basic algorithmic structures *sequence*, *selection* and *iteration*. At the same time, the course promotes the computational thinking skills of *algorithmic thinking* (e.g. follow algorithms, create algorithms to solve problems), *decomposition* (breaking down problems into smaller steps), *logical reasoning* and *evaluation* (e.g. identifying possible solutions and choosing the best one).

To experience the students' process of *conceptual change*, the teachers themselves work on selected tasks from the primary school course. In addition, actual results of students, as well as video sequences from real teaching situations, are examined. All exercises are discussed and possible difficulties are pointed out.

As an introduction to giving precise and clear instructions, the teachers program a human robot. The robot, which has to walk a certain route in the classroom, only moves if given the right orders. To further practice giving commands, they have to convert a pictorial instruction of a process (see Fig.1: process of making a sandwich) into unambiguous speech-based commands. Afterwards, the teachers work in groups to solve more complex tasks in a grid that is built up from carpet tiles (see Fig. 2). They create programs using haptic Scratch blocks (see Fig. 3) and execute it in the grid. This way, they can physically experience what later happens in a programming environment. We designed the tasks that they can be solved by using *selections* and *iterations*, but also by sequences.

After these unplugged exercises, the teachers are introduced to the programming environment Scratch. They work on a learning circle in which the core operations of Scratch are gradually introduced and that everyone can handle at their own pace. Starting from questions regarding the software handling, the stations lead from simple *sequences* to the implementation of *selections* and *iterations*. After the learning circle, we presented and discussed a variety of more open tasks that can be worked within Scratch.

4.2 Session 2 *(7 hours)*

The second day starts with a short recapitulation of the previous session and a short introduction about the importance of a deep understanding of the underlying algorithmic structures. It is also stressed that the teachers should practice the correct terminology of the discipline right from the beginning.

Figure 2: Grid activity

Figure 3: Haptic representations of Scratch blocks

After discussing a definition of the term *algorithm* using multiples examples from the teachers' everyday life, the teachers are presented an overview of the algorithmic structures, that will be worked on during the session. It is highlighted that they have already encountered all structures in session 1.

During the session, each programming structure is briefly described in pseudocode and Scratch blocks and deepened with introductory and advanced tasks using Scratch or the unplugged approach (see Fig. 4). In order to enable active learning, the theoretical input is kept as short as possible and the focus is on the independent solution of tasks and problems. To keep everybody engaged, different exercises of increasing complexity are offered for each topic. It is also possible for the teachers to work in pairs. During the exercise phases, the course instructors are available to support the participants and answer possible questions.

After each exercise phase, encountered problems, as well as commonly known problems, are discussed in detail in the plenum. Advantages and disadvantages of different solutions are discussed

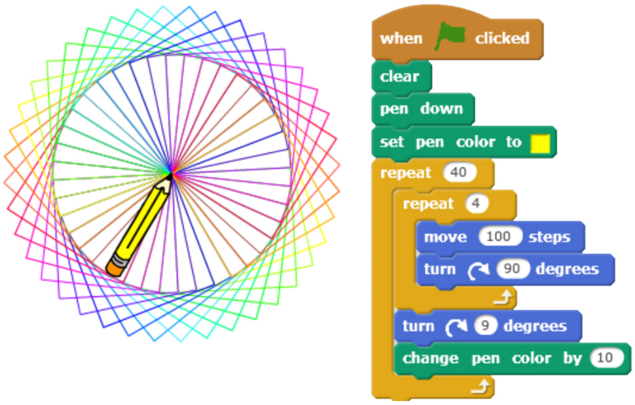


Figure 4: Task on nested repetitions

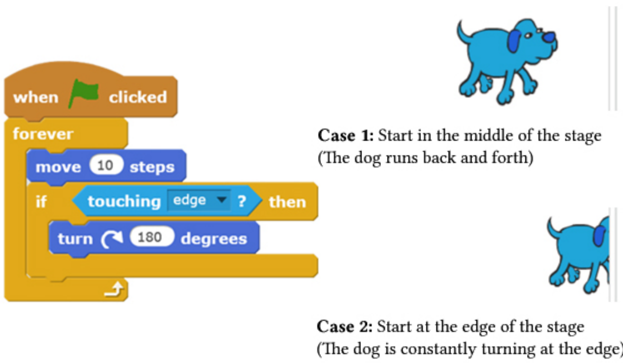


Figure 5: Example for a task in which code has to be interpreted *(Why is the dog constantly turning in case 2?)*

on the basis of the actual results of the teachers. In addition, they have to read and interpret code examples for each algorithmic structure (see Fig. 5). Furthermore, they get the opportunity to retrace different program flows instruction by instruction on presentation slides. This is an opportunity where common misconceptions about the algorithmic structures can be discussed.

This session is quite challenging and strenuous, as the work on the exercises requires a lot of concentration from the participants throughout the day. Therefore, sufficient breaks are scheduled.

4.3 Session 3 *(3 hours)*

In the previous sessions, algorithmics and programming have been introduced to the participants. Session 3 sets its focus on how the teachers could imagine implementing these concepts in their own teaching and the primary school curriculum.

In this context, it is pointed out that there are different possibilities to approach programming with the students and that they can set individual priorities (e.g. storytelling, focus on the planning process, animations).

In order to promote interaction and discussion, various poster walls are prepared on which ideas, inspirations and thoughts on different topics are captured by the participants (e.g. ideas for short-term implementations, ideas for medium to longer-term implementation, challenges). The collected results are then discussed in the plenum.

At the end of the session there is explicit time scheduled to discuss the further course of the project. It is also emphasized that the teachers can always contact us with any questions and possible problems. Support visits to the schools are also part of the project and can be requested by the teachers.

Table 1: Age Distribution of participating Teachers

participants	May workshop	June workshop	total
under 30 years	4	4	8
30-40 years	5	11	16
41-50 years	2	6	8
older than 50 years	5	2	7
	16	23	39

4.4 Materials

The following materials are available for the teachers:

Handouts: The participants receive a handout in which the basic algorithmic structures are described in detail. To further deepen the workshop contents, they receive the chapter of a student textbook on algorithms. In addition, a detailed concept of the primary school programming course, which they get to know in the first session, is provided (with didactic hints and a description of frequently made mistakes of the students).

Exercises: All tasks from the first and second session can be downloaded (both the worksheets and the Scratch files with and without solutions).

Images: The Teachers receive image files from various scratch blocks so that they can create their own teaching materials.

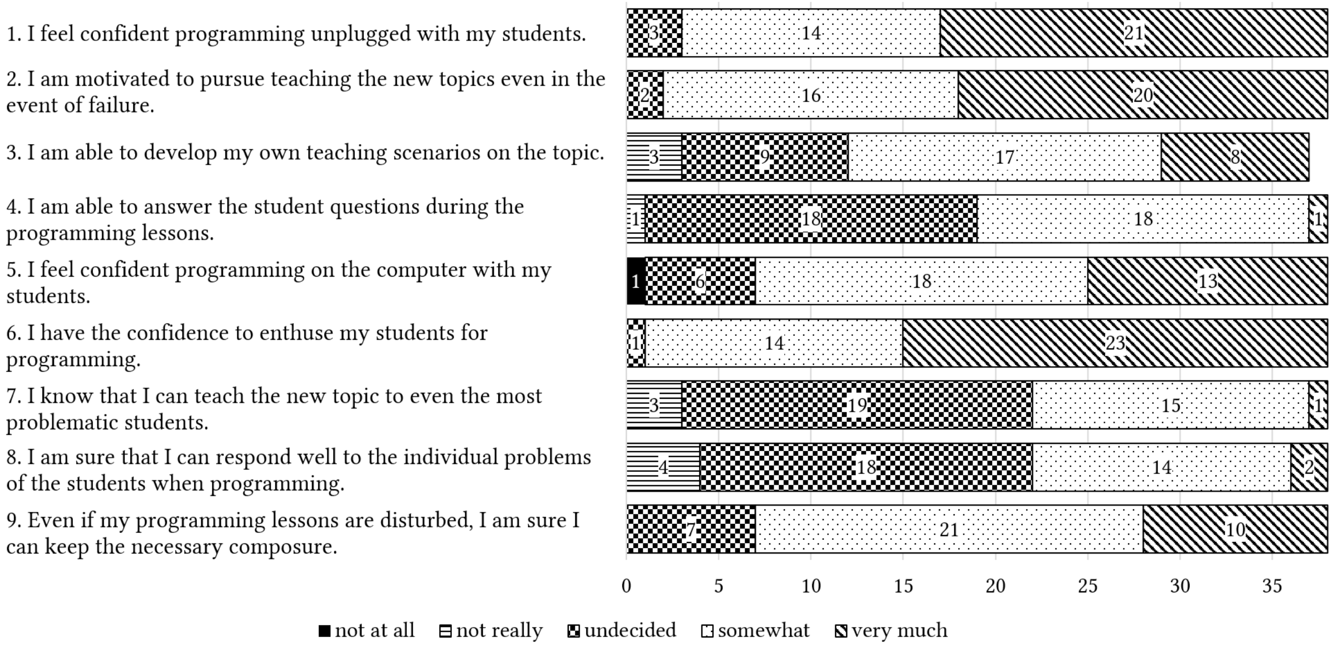
Teaching materials: All teachers receive a series of haptic Scratch blocks that they can use for unplugged exercises in their classroom.

5 PRELIMINARY RESULTS

We have held two teacher training workshops so far, one in May and one in June 2018. During both workshops, two course instructors who work in computer science education were present at all times. One of them has already gained a lot of experience in teaching algorithmics and programming in primary schools.

The May workshop was attended by 16 teachers, the June workshop by 23 teachers. In both courses one teacher was male, all others were female. The age of the participants ranged from under 30 years to over 50 years, while the group of 30-40 year-olds made up the largest part (see table 1). Across both groups, 24 teachers had no previous experience in Computer Science at all, 13 teachers had CS for 1-3 years as an elective or compulsory subject in school. 12 teachers stated, that they take part in the project on the initiative of their school principal and 15 stated they take part on their own initiative. Another 15 participants stated, that the initiative was

Figure 6: Please indicate to what extent the following statements apply.



evenly distributed between their own and the school principal's initiative.

The general response to the three-day teacher training was very positive both in the direct discussion with the teachers and the anonymous evaluation questionnaire from which we present an excerpt in the following. 33 teachers out of 38 stated, that they were *satisfied* with the workshop overall, 5 stated they were *rather satisfied*. This reflects the results of [17][21][33], who found out that that teacher trainings are met with satisfaction and acceptance when they are "close to the job" and relate to concrete teaching, when they offer opportunities for exchange with participating colleagues and when they take place in a pleasant atmosphere. All these characteristics were fulfilled by the structure of the workshops and the general conditions at the training center.

When asked if their expectations regarding the benefits for the implementation at the school were fulfilled, 17 stated *entirely*, 18 stated *to a large extent* and 3 stated *in parts*. In the questionnaire, the teachers also answered various questions relating to the self-efficacy in relation to the implementation of the workshop topics (see Fig. 6) that were constructed according to Schwarzer and Schmitz' teacher self-efficacy scale [30]. Here one can see that teachers are more confident about unplugged tasks than about programming on the computer (statements 1 and 5). The results also show that the teachers don't feel totally comfortable answering the students' questions (statement 4), teaching the new topics to problematic students (statement 7) and helping the students on individual programming problems (statement 8).

In the conversations with the teachers, it became clear, that they feel confident enough to make first attempts at teaching algorithmics and programming. Most of the teachers stated, that they liked the idea of approaching the topics at first in unplugged tasks and afterwards in Scratch. Concerns were expressed regarding the technical equipment of the schools and possible computer problems. It was considered very positive that the teachers were able to deep-dive into the topics for three days and could leave professional and family duties aside. The intensive time together also enabled the participants to get to know each other and the course instructors quite well.

Many teachers gave us the feedback that in the beginning, they had no idea what programming is and how to implement algorithmic thinking and programming in the primary school context. We had the impression that it was very beneficial for the teachers to come into contact with student materials on the very first day. In this way, fears could be reduced and the teachers realized that some topics are already being addressed in primary school anyway (e.g. giving clear instructions). The teachers were also very positive about the second session. Even though it was quite strenuous at times, time flew by and it was considered absolutely necessary to engage intensively with Scratch and the underlying algorithmic structures. It was assessed as positive that the tasks went beyond the competence level of the students.

6 CONCLUSION AND FUTURE WORK

Based on our experiences during the workshops and the teachers' initial feedback, we see no reason to change the structure and content of the workshops. However, we will provide the teachers with further programming tasks so that they become more secure in programming and will feel more confident in dealing with students' problems.

After the teachers' first attempts at teaching algorithmics and programming, we can make further conclusions as to whether the three-day workshop was sufficient and whether the motivation to implement the contents can also be maintained in everyday school life. In the course of the project we will use further questionnaires on the teachers' perceived self-efficacy and we will have the opportunity to interview the teachers and discuss the experiences they made. We also accompany individual teachers during their computer science lessons. We intend to combine these results in order to make statements on the effectiveness of the professional development concept in the future.

ACKNOWLEDGMENTS

We would like to express our special thanks to all the teachers involved in the project for their openness, curiosity and commitment.

REFERENCES

[1] Tim Bell. 2016. Demystifying Coding for Schools: What are we actually trying to teach?. In *Proceedings of the 11th Workshop in Primary and Secondary Computing Education*, Jan Vahrenhold and Erik Barendsen (Eds.). ACM, New York, NY, USA.

[2] Tim Bell and Caitlin Duncan. 2018. Teaching Computing in Primary Schools. In *Computer science education*, Sue Sentance, Erik Barendsen, and Carsten Schulte (Eds.). Bloomsbury Academic, London and New York and Oxford and New Delhi and Sydney.

[3] Mordechai Ben-Ari. 2001. Constructivism in Computer Science Education1. *Journal of Computers in Mathematics and Science Teaching* 20, 1 (2001), 45–73.

[4] Nadine Bergner, Hilde Köster, Johannes Magenheim, Kathrin Müller, Ralf Romeike, Ulrik Schroeder, and Carsten Schulte. [n. d.]. Zieldimensionen für frühe informatische Bildung im Kindergarten und in der Grundschule. 15–24.

[5] Corrado Böhm and Giuseppe Jacopini. 1966. Flow diagrams, turing machines and languages with only two formation rules. *Commun. ACM* 9, 5 (1966), 366–371. https://doi.org/10.1145/355592.365646

[6] Neil C. C. Brown, Sue Sentance, Tom Crick, and Simon Humphreys. 2014. Restart: the resurgence of computer science in UK schools. *ACM Transactions on Computing Education* 14, 2 (2014), 1–22. https://doi.org/10.1145/2602484

[7] Jérome Bruner. 1996. *Towards a Theory of Instruction.* Harvard University Press, Cambridge.

[8] Michael E. Caspersen. 2018. Teaching Programming. In *Computer science education*, Sue Sentance, Erik Barendsen, and Carsten Schulte (Eds.). Bloomsbury Academic, London and New York and Oxford and New Delhi and Sydney.

[9] Elizabeth A. Davis and Joseph S. Krajcik. 2016. Designing Educative Curriculum Materials to Promote Teacher Learning. *Educational Researcher* 34, 3 (2016), 3–14. https://doi.org/10.3102/0013189X034003003

[10] Mary Devine, Claude Houssemand, and Raymond Meyers. 2013. Instructional Coaching for Teachers: A Strategy to Implement New Practices in the Classrooms. *Procedia - Social and Behavioral Sciences* 93 (2013), 1126–1130. https://doi.org/10.1016/j.sbspro.2013.10.001

[11] John Dewey. 1938. *Experience and Education.* Collier Books, New York.

[12] Ira Diethelm and Melanie Schaumburg. 2016. IT2School – Development of Teaching Materials for CS Through Design Thinking. In *Informatics in Schools: Improvement of Informatics Knowledge and Perception*, Andrej Brodnik and Françoise Tort (Eds.). Vol. 9973. Springer, Cham, 193–198. https://doi.org/10.1007/978-3-319-46747-4{_}16

[13] Katrina Falkner and Rebecca Vivian. 2016. A review of Computer Science resources for learning and teaching with K-12 computing curricula: An Australian case study. *Computer Science Education* 25, 4 (2016), 390–429. https://doi.org/10.1080/08993408.2016.1140410

[14] Katrina Falkner, Rebecca Vivian, and Nickolas Falkner. 2014. The Australian Digital Technologies Curriculum: Challenge and Opportunity. In *Proceedings of the Sixteenth Australasian Computing Education Conference - Volume 148 (ACE '14)*. Australian Computer Society, Inc, Darlinghurst, Australia, Australia, 3–12.

[15] Anja Gärtig-Daugs, Katharina Weitz, Maike Wolking, and Ute Schmid. 2016. Computer science experimenter's kit for use in preschool and primary school. In *Proceedings of the 11th Workshop in Primary and Secondary Computing Education*, Jan Vahrenhold and Erik Barendsen (Eds.). ACM, New York, NY, USA, 66–71. https://doi.org/10.1145/2978249.2978258

[16] Katharina Geldreich, Alexandra Funke, and Peter Hubwieser. 2016. A Programming Circus for Primary Schools. In *Proceedings of the 9th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*. 46–47.

[17] Thomas R. Guskey. 2002. Professional Development and Teacher Change. *Teachers and Teaching* 8, 3 (2002), 381–391. https://doi.org/10.1080/135406002100000512

[18] Patricia Haden, Joy Gasson, Krissi Wood, and Dale Parsons. 2016. Can you learn to teach programming in two days?. In *Proceedings of the Australasian Computer Science Week Multiconference*. ACM, New York, NY, USA, 1–7. https://doi.org/10.1145/2843043.2843063

[19] Heather Hill, Deborah Ball, and Stephen Schilling. 2008. Unpacking pedagogical content knowledge: Conceptualizing and measuring teachers' topic-specific knowledge of students. *Journal for Research in Mathematics Education* 39, 4 (2008), 372–400.

[20] Linda Darling-Hammond, Maria E. Hyler, and Madelyn Gardner, and with assistance from Danny Espinoza. 2017. *Effective Teacher Professional Development.* Learning Policy Institute, Palo Alto, CA.

[21] Frank Lipowsky. 2010. Lernen im Beruf. Empirische Befunde zur Wirksamkeit von Lehrerfortbildung. In *Lehrerinnen und Lehrer lernen. Konzepte und Befunde der Lehrerfortbildung*, F. H. Müller, A. Eichenberger, M. Lüders, and J. Mayr (Eds.). Waxmann, Münster, 51–70.

[22] John Maloney, Mitchel Resnick, Natalie Rusk, Brian Silverman, and Evelyn Eastmond. 2010. The Scratch Programming Language and Environment. *ACM Transactions on Computing Education* 10, 4 (2010), 1–15. https://doi.org/10.1145/1868358.1868363

[23] Minjeong Park and Youl-Kwan Sung. 2013. Teachers' perceptions of the recent curriculum reforms and their implementation: What can we learn from the case of Korean elementary teachers? *Asia Pacific Journal of Education* 33, 1 (2013), 15–33. https://doi.org/10.1080/02188791.2012.756391

[24] William R. Penuel, Lawrence P. Gallagher, and Savitha Moorthy. 2011. Preparing Teachers to Design Sequences of Instruction in Earth Systems Science. *American Educational Research Journal* 48, 4 (2011), 996–1025. https://doi.org/10.3102/0002831211410864

[25] Jean Piaget. 1950. *The Psychology of Intelligence.* Harvard University Press, Cambridge.

[26] Douglas R. Powell, Karen E. Diamond, Margaret R. Burchinal, and Matthew J. Koehler. 2010. Effects of an early literacy professional development intervention on head start teachers and children. *Journal of Educational Psychology* 102, 2 (2010), 299–312.

[27] Kiki Prottsman. 2014. Computer science for the elementary classroom. *ACM Inroads* 5, 4 (2014), 60–63. https://doi.org/10.1145/2684721.2684735

[28] Tracie Evans Reding and Brian Dorn. 2017. Understanding the Teacher Experience in Primary and Secondary CS Professional Development. In *Proceedings of the 2017 ACM Conference on International Computing Education Research*, Josh Tenenberg and A. Special Interest Group on Computer ScienceC.M. Education (Eds.). ACM, [S.l.], 155–163. https://doi.org/10.1145/3105726.3106185

[29] Kathleen J. Roth, Helen E. Garnier, Catherine Chen, Meike Lemmens, Kathleen Schwille, and Nicole I.Z. Wickler. 2011. Videobased lesson analysis: Effective science PD for teacher and student learning. *Journal of Research in Science Teaching* 48, 2 (2011), 117–148. https://doi.org/10.1002/tea.20408

[30] Ralf Schwarzer and Matthias Jerusalem (Eds.). 1999. *Skalen zur Erfassung von Lehrer- und Schülermerkmalen: Dokumentation der psychometrischen Verfahren im Rahmen der Wissenschaftlichen Begleitung des Modellversuchs Selbstwirksame Schulen.* Berlin.

[31] Sue Sentance, Mark Dorling, and Adam McNirol. 2013. Computer Science in Secondary Schools in the UK: Ways to Empower Teachers. In *Informatics in Schools*, Ira Diethelm and Roland T. Mittermeir (Eds.). Springer.

[32] Lee S. Shulman. 1986. Those Who Understand: Knowledge Growth in Teaching. *Educational Researcher* 15, 2 (1986), 4–14.

[33] Christine Smith and Marilyn Gillespie. 2007. Research on Professional Development and Teacher Change: Implication for Adult Basic Education. *Review of Adult Learning and Literacy* 7 (2007), 205–244.

[34] The Design-Based Research Collective. 2003. Design-Based Research: An Emerging Paradigm for Educational Inquiry. *Educational Researcher* 32, 1 (2003), 5–8.

[35] Heikki Topi. 2015. Gender imbalance in computing. *ACM Inroads* 6, 4 (2015), 22–23. https://doi.org/10.1145/2822904

[36] Paul Webb. 1992. Primary science teachers' understandings of electric current. *International Journal of Science Education* 14, 4 (1992), 423–429.

[37] Aman Yadav, Sarah Gretter, Susanne Hambrusch, and Phil Sands. 2016. Expanding computer science education in schools: Understanding teacher experiences and challenges. *Computer Science Education* 26, 4 (2016), 235–254. https://doi.org/10.1080/08993408.2016.1257418