Foundations of Computer Science in General Teacher Education – Findings and Experiences from a Blended-Learning Course

Stefan Seegerer[1], Tilman Michaeli[2], and Ralf Romeike[1]

[1] Freie Universität Berlin, Computing Education Research Group,
[2] Technical University of Munich, TUM School of Social Sciences and Technology, Computing Education Research Group,
stefan.seegerer@fu-berlin.de

Abstract. With regards to the digital transformation, the consensus that computer science education plays a central role in shaping "digital education" is now emerging: Beyond the efficient and reflective use of information systems, new topics and methods arise for all school subjects that require computer science competencies and must be anchored in general teacher education. However, in light of students' heterogeneity, the question of how motivation, subject-specific demands, and applicability in subject teaching can be harmonized presents a particular challenge. This paper presents key findings and experiences from the research-led development and subsequent evaluation of a blended learning course offering. This course offering provides student teachers of all subjects and school types with basic computer science competencies for teaching in the digital world. On this foundation, success factors and good practices in the design of the course are identified. It is shown that the design of such courses can be successful if illustrative examples are used, communication and collaboration are promoted and, in particular, references and application perspectives for the respective subjects are taken into account.

Keywords: computing education, digital education, general teacher education, blended learning

1 Introduction

With the digital transformation, the way we communicate, use technologies, work, or gather information is changing in all areas of life. School subjects are also affected by this transformation process – and increasingly it is acknowledged that computer science competencies are not only relevant for the efficient and reflective use of digital media or information technology systems [1]. For example, in science classes, simulations and data analysis – also referred to as the third and fourth pillars of science – are used to gain insights. Additionally, sensors may be used to collect digital measurements. In economics classes, digital business models and their impact on the labor market are addressed; and in religion classes, ethics are discussed in the context of algorithms and artificial intelligence. These changes, which affect not only content such as artificial intelligence but also methods such as data analysis or simulations, represent an important aspect of “digital education”.

Teachers of all subjects and school types need the corresponding competencies to be able to address these changes appropriately in their teaching. Computing Education plays a central role in this context: Only with competencies in handling and evaluating data, or by having a basic understanding of algorithms, are teachers able to address corresponding phenomena in the classroom in a well-founded way. Accordingly, the training and continuing education of teachers is a central task in the context of digital education, in which computer science must assume a leading role. In light of the students’ heterogeneity, the question arises as to how motivation, content and applicability in the classroom can be harmonized. Therefore, this paper presents key findings and experiences from the research-led development and subsequent evaluation of a blended learning course. This course provides student teachers of all subjects and school types with basic computer science competencies for teaching in the digital world. The course has been running successfully now for 4 years with almost 1000 students participating at 3 universities.

2 Related Work

Digital education means new topics and methods for all school subjects [1], which go beyond the efficient and reflective use of computer science systems. This means they have to be anchored particularly in the education of teachers of all subjects and school types. There is now a consensus – not limited to the perspective of computing education – that computer science education plays a central role in the design of this kind of digital education and that teachers of all school subjects need corresponding computer science competencies. Various parties are calling for the necessary foundations of computer science to be anchored in teacher training. Following the strategy “Education in the Digital World” of the German Standing Conference of the Ministers of Education and Cultural Affairs, the *Research Group Digital Campus Bavaria* has formulated 19 competencies for teaching in the digital world [2]. There, so-called media-related computer science skills are explicitly required, which include “conceptual knowledge of databases and algorithms”. An expert commission convened by the German Ministry for Education and Research also calls for “[a]ll institutions of teacher education [...] to [promote] computer science competence (in the sense of algorithmic thinking, data literacy, computational thinking, and data security)” [3]. Also, on an international level, several initiatives emphasize the importance of computer science for digital literacy, such as the ”informatics for all” strategy [4].

However, few approaches anchoring the foundations of computer science in general teacher education exist. One such approach is the lecture series “Computer Science in Everyday Life” at the University of Wuppertal [5]. The course focuses on everyday phenomena which are analyzed and evaluated from the perspective of computer science, in order to provide future teachers with "expert access to the science of computer science". Furthermore, Yadav et al. [6] integrated a one-week module (2x50 minutes lecture) on computational thinking (CT) into a psychology course for student teachers. This short module focuses specifically on the related concepts of abstraction, reasoning, algorithms, and debugging, and explaining their importance in the classroom in more detail. Other scholars developed courses specifically designed for prospective elementary school teachers [7, 8]. The focus of these courses was on teaching computer science content for teaching at a primary level. Therefore, in addition to subject-specific content, computer science education topics, e. g. the use of unplugged activities, were addressed as well.

To summarize, existing offers either choose a subject-oriented approach, are comparatively short, or primarily aim at enabling the teaching of computer science competencies. In contrast to this, the applicability in the respective subject, which is necessary due to the change of all subjects by the digital transformation, has not been the focus research so far. According to previous experience, with an approach focusing primarily on computer science content knowledge, this can be established "only to a limited extent" [5]. There is a lack of research on how the teaching of the foundations of computer science can be designed for student teachers of all subjects and school types in order to contribute to applicability in the subject. However, this would enable future teachers to address the new contents and methods, that are a result of the digital transformation, in their teaching.

3 Approach

In the following section, we present the central findings and experiences of the research-led development, testing, and evaluation of a study program, which provides student teachers of all subjects and types of schools with fundamental computer science competencies for teaching in the digital world. In the first step, conditions and challenges for the design of such a study program will be identified. Based on this step, the implementation is presented in the form of organizational decisions and theoretically-derived design principles. These have been refined in the course of the accompanying research in order to address the corresponding conditions and challenges. The evaluation accompanying the module is used to examine how certain design decisions were perceived. Finally, the entire study program is evaluated in a pre-post design. This allows for the drawing of conclusions regarding the suitability of the design principles and organizational decisions.

4 Design parameters, Implementation and Outcomes

4.1 Conditions and Challenges

Motivation. Computer science education is increasingly seen as an important part of general teacher education. However, it can be assumed that due to long-term established stereotypes, a lack of prior knowledge [8] and insufficient conceptions of computer science and its role in the digital transformation, only few students exhibit an intrinsic interest on topics of computer science. Furthermore, computer science is perceived as complex [5] and often reduced to working with computers [6]. Many students do not see the need to change their role from "outsider" to "insider" [9]. Also, the discourse on digital education is often dominated by the use of digital media in the classroom. In our view, the most important challenge thus arises from the motivation: *How can we attract students to want to deal with computer science topics and how can we continuously maintain motivation while contributing to a positively-shaped image of computer science?*

Organizational Conditions. The organizational conditions still pose a particular challenge for such a course, as at most institutions, digital education is not a (mandatory) part of teacher education curricula. Furthermore, there is a lack of corresponding qualified staff and resources: *How can the numerous student teachers per institution be provided with a well-founded computer science education offer in a timely, financially-feasible, and scalable manner that meets the demands of good education as elaborated in computing education over the last 30 years (for example, contextualized, modeling-based, idea-based)*?

Content-wise Challenges. Many years of experience in computer science education at all age levels show that the design of computing education programs poses special challenges. Computer science is often abstract and thus difficult to grasp, requires problem-solving skills, has a wide range of topics, and technical skills are needed to implement computer science models, which are regularly perceived as demotivating in particular in the context of learning to program [10]. Furthermore, a great heterogeneity of students is to be expected, both in terms of their prior computer science experience and in terms of the types of schools and subjects studied. *Given the students' expected heterogeneity, how can computer science competencies be developed in a way that offers concrete application possibilities for subject teaching in various subjects?*

4.2 Design of the course

Organizationally, the course was designed and advertised as an optional blended learning course (5 ECTS) for "competencies for teaching in the digital world", addressing the aforementioned challenges as follows: By embedding it in the larger context of digital education, the prospect of being able to apply the acquired competencies to subject teaching, and the close cooperation and joint implementation with colleagues from media education, the computer science core was intended to remain in the background. Five of the twelve modules of the course deal with the acquisition of computer science competencies, whereby exemplary topics from the broad spectrum in computer science were interwoven with the requirements in the context of digital education (cf. tab 1)[3]. The comprehensive and appealing presentation of the modules as online learning units should, on the one hand, take into account the expected heterogeneity of the students, so that they can work as far as possible independently and at their own pace, as well as pursue personal interests. On the other hand, the few permanently-available staff and material resources should be used in such a way that as many students as possible can benefit from the offer. Within the ongoing scaling and the Covid19 pandemic, the synchronous part was reduced to the implementation and presentation of the final project phase. The loss of the opportunity to engage in collaborative learning experiences in face-to-face encounters was handled with appropriate alternatives considered in the design principles.

Table 1. Modules of the course, modules with focus on computer science in **bold**.

Modules 0 – 5	Modules 6 – 11
0: Digital literacy in the subject classroom	6: Research, store, and evaluate digitally
1: **Fundamentals of digitalization**	7: Communication, interaction, and collaboration
2: Media Culture History, Theory, and Ethics	8: **From data to professional knowledge**
3: **Computers and the Internet**	9: **Simulations in the professional context**
4: Creativity in digitalization	10: Social networks
5: **Solving subject-specific problems with algorithms**	11: Outlook: Digital opportunities and boundaries

The **content and methodological design** is guided by design principles initially grounded in theory and later sharpened during the accompanying research, which, in addition to meeting the challenges outlined, aim to illustrate the relevance, applicability, and creative possibilities of computer science:

Through *scaffolding*, the learning process is supported by guiding assistance and the degrees of freedom in performing a task are initially limited [11]. This (optional) temporary support is intended to help with the understanding of new concepts so that learners can later work on similar tasks independently. In the context of the course, the use-modify-create approach [12] was used for this purpose, explanatory videos were created, and practical tasks were initially guided in small steps, which was intended to take into account the students' limited prior experience. The free-text responses to the module-accompanying evaluations showed that the exercises on programming were challenging, but the

[3] A publicly accessible version of the modules can be found at blindedforreview.

explanatory videos were rated as particularly helpful, so that such assistance for programming tasks was consistently expanded.

To achieve *contextualization* in different subjects, examples of application or relationships to the subjects were shown and social implications taken into account [13]. The examples were transferred by the students to their own subjects in reflection tasks to ensure applicability in their teaching. The feedback from the students showed that the contextualization was important for recognizing the contents' significance for their own subject and teaching. Furthermore, the evaluations showed that the relevance of programming, in particular, must be made clear through appropriate contexts and illustrative examples.

The consistent *application of the "pedagogical double-decker"*, which refers to the idea of training teachers with methods and examples they themselves can use for their later teaching, is intended to underline the practical relevance of what has been learned and to make the learning processes more sustainable by making the contents also tangible at the action level [14], but the activities can also be applied directly or transferred into teaching. In the evaluations, students continuously emphasized this as particularly useful for their own teaching.

The continuous *promotion of communication and cooperation* turned out to be one of the most important design principles regarding the implementation as a blended learning course, both for motivational factors and the professional exchange. For this purpose, discussion forums, digital bulletin boards or tasks that required explicit collaboration were used. On the one hand, this stimulated exchange between the subjects, thus highlighting the interdisciplinary importance of the digital transformation with regards to changing subjects and schools, and on the other hand, it addressed the great amount of heterogeneity. The students found the tasks for the mutual exchange of ideas to be enriching, precisely because they contributed to the relevance of the content.

In order to lower motivational and technical hurdles for the students, *low-threshold access* through active learning and playful tinkering was applied as a consistent design principle [15]. Animations, applets, games, and the programming environment Snap*!* were used for this purpose, among others. The latter is not only very accessible but also allows for easy creation, investigation, and further development of simulations or data analysis, which additionally directly provides applicability in subject teaching. These elements were often mentioned positively by the students in the free-text comments and described as motivating.

4.3 Overall Evaluation

Since the piloting of the first modules in winter term 2018/19, the course has been offered every semester at three universities due to increasing interest. Overall, out of the 709 students thus far, an exceptionally high proportion of female students (75%) for computer science courses stands out. This is notably due to the fact that a large proportion of the participants could be recruited from the elementary school teacher training program. Given the growing importance of computer science education in elementary school [16], this is particularly gratifying.

Looking at interest in computer science, before the course began, 62% of the 424 participants in the evaluation already agreed with the statement "computer science is interesting"; 17% of the respondents initially saw computer science as rather boring. In the pre-post comparison of interest, the Wilcoxon signed-rank test[4] yields a significant increase with medium effect size according to [17] at a significance level of $\alpha = 0.05$ with a median of 5 in the pretest and a median of 6 in the posttest ($n = 231$, $p < 0.001$, $r = 0.36$) – 79% of the respondents now see computer science as at least rather interesting, only 8% as rather boring. It is likely that demonstrating the width of computer science and the corresponding creative and collaborative opportunities have contributed to the increase in interest.

A central goal of the course is to promote computer science competencies, particularly ensuring their applicability in the subject classroom. The students' self-assessment of computer science competencies (for example, on the components of a computer, the coding of data, or the subject-specific effects of algorithms), we found only little prior knowledge – in line with our expectations (cf. fig. 1). Comparing the results of those who attended computer science within their K-12 education (46%), the only significant difference was for the question "I can explain how computers store data in 0 and 1" (Mann-Whitney U tests at a significance level of $\alpha = 0.05$).

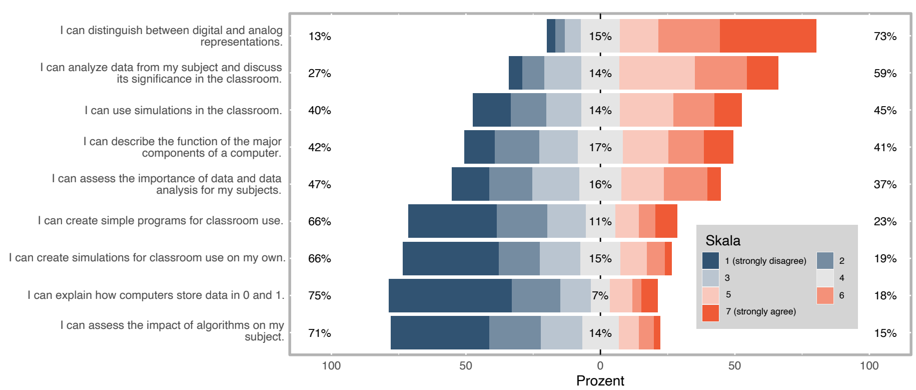

Fig. 1. self-assessment of students in pretest (n=424)

In the posttest, matching was established for 231 participants based on the individual participant code. According to the Mann-Whitney U test, the answers of the participants for whom this matching was possible did not differ significantly at a significance level of $\alpha = 0.05$ in both the pretest and the posttest (in all questions except the one about the importance of data and data analyses

[4] Due to the absence of a normal distribution, non-parametric test procedures were used consistently.

for own subjects) from those for whom no matching was available. Therefore, it can be assumed that the results are sufficiently representative. In Tabular 2, the respective medians, the p-value of the Wilcoxon signed-rank test (H_0: No or negative trial effect)[5] and the correlation coefficient r[6] are used as a measure of effect size.

Table 2. self-assessment of competencies (n=231, Likert scale from 1 (does not apply) to 7 (applies completely)).

Statement	med pre	med post	Wilcoxon-Test	r
I can explain how computers store data in 0 and 1.	2	6	$p < 0,001^*$	$0,67$
I can assess the impact of algorithms on my subject.	2	5	$p < 0,001^*$	$0,66$
I can assess the importance of data and data analysis for my subjects.	4	6	$p < 0,001^*$	$0,62$
I can use simulations in the classroom.	4	6	$p < 0,001^*$	$0,59$
I can create simulations for classroom use on my own.	3	5	$p < 0,001^*$	$0,61$
I can analyze data from my subject and discuss its significance in the classroom.	5	6	$p < 0,001^*$	$0,52$
I can describe the function of the major components of a computer.	4	6	$p < 0,001^*$	$0,54$
I can create simple programs for classroom use.	3	6	$p < 0,001^*$	$0,63$
I can distinguish between digital and analog representations.	6	7	$p < 0,001^*$	$0,38$
Total	3.66	5.78	$p < 0,001^*$	$0,69$

The results in the pre-post comparison show a significant increase for the self-assessment of competencies – in all sub-questions as well as overall. The effect sizes are strong in almost all cases.

5 Discussion and Conclusion

Anchoring computer science foundations in general teacher education is a central task in the context of digital education. Our results confirm the assumption that high heterogeneity and low prior experience of students in particular have to be

[5] Significant test results to a significance level of $\alpha = 0.05$ are indicated by a $*$.

[6] The correlation coefficient r is defined as $r = \frac{z}{\sqrt{n}}$, where z indicates the standardized test statistic of the Wilcoxon signed-rank test and n indicates the sample size. According to [17], $r = 0.10$ and above is considered a weak effect, $r = 0.30$ and above is considered a medium effect, and $r = 0.50$ and above is considered a strong effect

taken into account for the design of appropriate course offerings. Furthermore, we found no significant influence of prior K-12 computer science education.

A notable observation of the implementation was that the *digital transformation* in the students' view was initially almost exclusively related to media *use*. Within the course, the consequences of the digital transformation for everyday life and the subjects and scientific disciplines were emphasized. As can be seen from the feedback, the students were thus able to better understand phenomena or topics that are gaining relevance in their subjects in the course of the digital transformation after the course. Therefore, the modules create the necessary basis for discussing the effects of digital change in their subjects. We see the consistent transfer of interdisciplinary computer science education to a contextualized view in relation to the subject studied as a central criterion for success. However, programming remains a particular but rewarding challenge for students. Here, we have learned that – especially in online environments – intense scaffolding is necessary. Furthermore, appropriate contextualization is central to clarify relevance and thus contribute to motivation, so that most students could say of themselves, not without pride: "I have programmed for the first time".

The students' interest in computer science also increased slightly during the course of the study program. Such an increase cannot be taken for granted. For example, [6] finds that prospective teachers' interest in computer science did not change as a result of their corresponding course offerings, and [18] concludes that interest in computer science actually declined slightly in an after-school learning lab regardless of the module, with older visitors' interest actually declining slightly more in comparison.

In summary, it can be seen that not only can foundations of computer science be prepared in an appropriately accessible way for students of all subjects and school types, but that students were also able to recognize the meaningfulness or necessity of these computer science concepts and that their confidence in their competencies for teaching in the digital world could be strengthened. A strong contextualization in the respective subjects, intensive scaffolding, the promotion of communication and collaboration, playful approaches, and the use of the "pedagogical double-decker" proved to be particularly successful in the design. The results thus provide promising guidelines for the development of computer science courses for general teacher education that prepare future teachers for teaching in the digital world.

References

1. GFD: Fachliche Bildung in der digitalen Welt – Positionspapier der Gesellschaft für Fachdidaktik [professional education in the digital world – position paper of the society for subject matter teaching and learning] (2018), http://www.fachdidaktik.org/wp-content/uploads/2018/07/GFD-Positionspapier-Fachliche-Bildung-in-der-digitalen-Welt-2018-FINAL-HP-Version.pdf, last accessed 2022/08/01

2. Forschungsgruppe Lehrerbildung Digitaler Campus Bayern: Kernkompetenzen von Lehrkräften für das Unterrichten in einer digitalisierten Welt [core competences of teachers for teaching in a digital world]. merz 4, 65–74 (2017)
3. van Ackeren, I.e.a.: Digitalisierung in der Lehrerbildung: Herausforderungen, Entwicklungsfelder und Förderung von Gesamtkonzepten [digital transformation in teacher education: challenges, areas of development and advancement of concepts]. Die Deutsche Schule 111(1), 103–119 (2019)
4. Caspersen, M.E., Gal-Ezer, J., McGettrick, A., Nardelli, E.: Informatics for all the strategy. ACM (2018)
5. Losch, D., Humbert, L.: Informatische Bildung für alle Lehramtsstudierenden [computer science education for all pre-service teachers]. In: Pasternak, A. (ed.) Informatik für alle. pp. 119–128. GI, Bonn (2019)
6. Yadav, A.e.a.: Computational thinking in elementary and secondary teacher education. ACM TOCE 14(1) (Mar 2014)
7. Casali, A., Monjelat, N., San Martín, P., Zanarini, D.: Primary level teachers training in computer science: Experience in the argentine context. In: Pesado, P., Arroyo, M. (eds.) Computer Science – CACIC 2019. pp. 389–404. Springer International Publishing, Cham (2020)
8. Döbeli Honegger, B., Hielscher, M.: Vom Lehrplan zur LehrerInnenbildung - Erste Erfahrungen mit obligatorischer Informatikdidaktik für angehende Schweizer PrimarlehrerInnen [from curriculum to teacher training - first experiences with compulsory computer science education for pre-service swiss primary teachers]. In: INFOS 2017. pp. 97–107. GI (2017)
9. Schulte, C., Knobelsdorf, M.: Attitudes towards computer science-computing experiences as a starting point and barrier to computer science. In: Proceedings of the Third International Workshop on Computing Education Research. p. 27–38. ICER '07, Association for Computing Machinery, New York, NY, USA (2007), https://doi.org/10.1145/1288580.1288585
10. Kinnunen, P., Simon, B.: Experiencing programming assignments in cs1: The emotional toll. In: Proceedings of ICER'10. p. 77–86. ACM, New York, NY, USA (2010)
11. Lin, T.C.e.a.: A review of empirical evidence on scaffolding for science education. International Journal of Science and Mathematics Education 10(2), 437–455 (2012)
12. Lee, I.e.a.: Computational thinking for youth in practice. ACM Inroads 2(1), 32–37 (Feb 2011)
13. Guzdial, M.: Does contextualized computing education help? ACM Inroads 1(4), 4–6 (2010)
14. Arnet-Clark, I., Smeets-Cowan, R., Kühnis, J.: Competences in teacher education at schwyz university of teacher education (phsz), and the swiss education policy. e-Pedagogium (2) (2015)
15. Petre, M., Richards, M.: Playful pedagogy: empowering students to do, design, and build. In: Informatikkultur neu denken - Konzepte für Studium und Lehre, pp. 41–54. Springer Vieweg, Wiesbaden (2014)
16. European Union and Audiovisual and Culture Executive Agency Education: Digital education at school in europe. Publications Office of the European Union, Brussels (2019)
17. Cohen, J.: A power primer. Psychological bulletin 112(1), 155–159 (1992)
18. Bergner, N.: Konzeption eines Informatik-Schülerlabors und Erforschung dessen Effekte auf das Bild der Informatik bei Kindern und Jugendlichen [Conceptualisation of a computer science student laboratory and research of its effects on the view of children and adolescents on computer science.]. Phd thesis, RWTH, Aachen (2016)