# Computing Education in (the digital) Transformation

**Tilman Michaeli**
Computing Education Research Group
Technical University of Munich
TUM School of Social Sciences and Technology
tilman.michaeli@tum.de

## 1. Computer Science, Computing Education, and the Digital Transformation

Since its beginnings, computer science (CS) is characterized by continuous change, that throughout history is reflected in the corresponding K-12 subject. CS started out as a specialized subdomain within other disciplines such as mathematics and electrical engineering, focusing on hardware and computers. Correspondingly, one of the first approaches for an independent subject in K-12 education, developed about 50 years ago, was called "Rechnerkunde"[1]. It aimed for an understanding of the hardware of real data processing systems by focusing on mathematical-technical fundamentals such as circuits and architecture (Frank & Meyer, 1972). Over the course of the recognition of computer science as a new scientific discipline and its triad of theoretical, technical, and practical computer science, CS attained more and more influence on everyday life, in particular, due to the spread of personal computers. Once more, this change became apparent in the notion for a K-12 subject as well, as algorithms and applying programming to solve real-world problems gained importance. Going further, especially in the last decades we see a much broader perspective of what is considered CS. Fields that investigate the interactions between CS and society – thus going beyond the solely technical aspects of CS – are more prominent than ever. This was also reflected in computing education, as educational standards for CS such as the German "Bildungsstandards" (Brinda et al., 2008) or the American CSTA K-12 computer science standards (Tucker, 2003) included respective competencies.

Currently, another evolution for computer science and its significance in our daily lives is taking place: CS is the key driver of the so-called digital transformation that leads to fundamental changes in the way we communicate, use technology, work, or gather information. Furthermore, we see a growing importance of CS in science and research as a consequence of these developments. This becomes most apparent when looking at different application fields: disciplines such as bioinformatics, legal informatics, business informatics, geoinformatics, and the digital humanities investigate discipline-specific issues building upon CS approaches (Riedel, Streit, Wolf, Lippert, & Kranzlmüller, 2008). The discussion on Computational Thinking is also partly rooted in the increased importance of CS in other disciplines (Csizmadia et al., 2015). However, this is not limited to specific areas. All scientific disciplines are significantly affected by the digital transformation, e.g. by new methods with simulation or data analysis considered as the third and fourth pillars of science, new topics that emerge, or new tools that are used to solve problems (Tolle, Tansley, & Hey, 2011). This is becoming increasingly relevant for K-12 education as well, as this shift in the scientific disciplines is also reflected in the corresponding subjects. In consequence, CS skills are also needed to address corresponding subject-specific phenomena (such as digital business models or the ethical implications of artificial intelligence) appropriately (Seegerer, Michaeli, & Romeike, 2022). For computing education, this raises the question of what computer science competencies everyone needs to deal with these changes in a responsible manner.

Typically, the term "digital education" is used to describe competencies necessary for participating in a "digital world". However, the debate in politics, media, and society as well as parts of the scientific community is mainly limited to a reductionistic understanding in form of "digital" or "media literacy". This way, phenomena in students' daily lives as well as the aforementioned changes in all K-12 subjects – which go far beyond using media – are not even remotely taken into account. Furthermore, the digital

[1] roughly translates to "becoming knowledgeable in the subject of computers", see `digitalkunde.info`.

transformation is an ongoing process. Technology – and education on how to use it – that currently is relevant, might be outdated in the near future due to this rapid and dynamic transformation. Therefore, it is crucial to identify and convey the time-independent underlying ideas, concepts, and principles of the digital world, that are based on computer science (Denning, 2004).

In summary, these developments result in an updated "mission statement" for computing education in consequence of the digital transformation: Computing education can be considered as the foundation for digital education and *aims at empowering everyone to understand the digital world and its phenomena as well as getting involved in shaping it* (see for example (Brinda & Diethelm, 2017)).

## 2. Implications for Computing Education

Like CS itself, computing education has to transform to meet this purpose. While already aiming at everyone and general education ("Bildung") in K-12 – and not just preparing professionals –, the significance of CS concerning the digital world has to be emphasized. This results in several challenges, that will be discussed in the following. Not all of those are necessarily new (focusing on ideas, incorporating the impact of CS, or constructionist learning in meaningful contexts). However, they gain importance in light of the digital transformation and for computing education as the foundation for digital education.

### 2.1. Embracing the socio-cultural impact of computer science

In the German discourse, the so-called "Dagstuhl triangle" (Brinda & Diethelm, 2017) forms a widely accepted model for digital education. It proposes three perspectives on digital phenomena or systems: A "technological perspective", considering how they work, a "socio-cultural perspective" questioning the interaction with individuals and society as well as an "user-oriented perspective", focusing on their creative and effective usage. Various approaches in other (inter)national contexts essentially emphasize similar perspectives using other terms or structures (Brown, Sentance, Crick, & Humphreys, 2014). Using this categorization, CS curricula as well as the actual teaching in the classroom typically are focusing predominantly on the technological perspective – and thus "core CS" –, while the user-oriented perspective is mostly attributed to media education. In contrast, the socio-cultural perspective is widely overlooked. However, only a profound understanding of underlying CS principles of digital systems and phenomena enables a well-founded analysis and evaluation of possibilities, limits, and mutual interdependencies of technology, individuals, and society. Therefore, to fulfill its role as the foundation for digital education, computing education has to emphasize and explicitly include this perspective in a truly multi-perspective manner – despite being challenging and possibly out of teachers' comfort zone.

### 2.2. Re-evaluation of traditional content according to the requirements of the digital world

Furthermore, typical CS curricula and content areas have to be evaluated from the perspective of digital education. The topic of data management provides an impressive example of this: Traditionally, content such as database systems, the relational representation of data using tables, modeling entities and their relations, or database queries in SQL dominate respective CS curricula. However, from the perspective of digital education, a lot of (new) ideas gain importance, such as metadata, redundancy, consistency, synchronization, data analysis, big data, or data security (Grillenberger & Romeike, 2014). At the same time, this also reflects the development in CS, as the scientific field of data management evolved in the last 15-20 years from a focus on relational database systems towards areas such as NoSQL, data mining, data stream systems, Big Data, and so on. This calls for a re-evaluation for other "traditional" CS content in the classroom as well. The example of data management shows, that it's not solely about the underlying technology used to store or handle data, but the overall way we gather, process, and share data is influenced by the digital transformation. These skills are, what learners need in a digital world.

### 2.3. "New" content in consequence of the digital transformation

In recent years, more and more areas of our lives are heavily influenced and shaped by artificial intelligence (AI) and machine learning, in particular. As a result, an increasing number of CS curricula are being extended to include the topic of AI. This case is thus exemplary for "new" topics in the CS classroom to address the challenges of the digital world, as besides increasingly powerful computing systems only a steadily growing volume of data as a consequence of the digital transformation enabled those technological advancements. The same goes for other topics such as data science, cyber security, or embedded ubiquitous systems. For computing education research, this provides a huge challenge: To prepare students for their lives in the digital world in the long term, those competencies that are fundamental to those subject areas need to be identified, without falling for short-lived technological developments or “hypes”. For artificial intelligence, we propose a curriculum of learning objectives, based upon the Dagstuhl-triangle, taking into account the significant impact AI has on society and allowing for merging them into CS curricula (Michaeli, Romeike, & Seegerer, 2022). With quantum computing, another possibly disruptive technology that promises opportunities, but might also pose new challenges for our society is already on the horizon (Seegerer, Michaeli, & Romeike, 2021). To fulfill its role as the foundation for digital education, computing education has to make the corresponding fundamentals, applications, and implications of these technologies accessible. Once more, this also emphasizes the reflection of the development of CS within computing education.

### 2.4. Focus on abstract ideas, not technological details

In computer science, abstraction allows for making supposedly hard things easy – even trivial – and in consequence focus on things such as how CS impacts society instead of details of implementation. To achieve the goal of preparing students for their lives in the digital world, so that even in 20, 30, or 40 years they are able to have a certain level of understanding of situations and phenomena surrounding them, we have to provide them with very abstract ideas and principles of technology. Nevertheless, in computing education, there is still the tendency to emphasize the most elementary aspects. An infamous example for this is the binary system – in perfect tradition of the “Rechnerkunde” approach from 50 years ago. Obviously, teaching has to make things concrete and work on tangible examples. However, understanding binary won’t allow students to explain phenomena from the digital world – not to mention getting involved in shaping it. Therefore, the ultimate goal is the abstract idea, such as representing information digitally. However, the binary system or even the conversion in other systems is often a prominent content of CS curricula. This holds also true for currently discussed topics such as AI, where a lot of curricula and teaching concepts focus on a single perceptron or neural networks. In contrast, putting the spotlight on abstract ideas such as supervised learning and its applications for classification or regression problems (that can be implemented e. g. using neural networks) might actually help to understand phenomena in the students’ daily lives – independent of technological details.

### 2.5. Not just what but also how

While talking a lot about *what* to teach and incorporate in curricula, the perspective of *how* to do this is equally important. Again, AI will serve as an example, as numerous unplugged approaches are typically used for introducing this topic to students. Those approaches are a great teaching method and particularly suitable as they allow for focusing on the actual abstract idea and not technological details in a fun and engaging way. However, they do not allow for actively and creatively designing computational artifacts and experiencing their effects. Given the goal of empowering students to not just understand but also be involved in shaping the digital world, staying on an understanding level doesn’t suffice. Therefore, teaching must go further than unplugged and enable students to create and construct (see (Michaeli, Seegerer, Kerber, & Romeike, 2022) or (Jatzlau, Michaeli, Seegerer, & Romeike, 2019) for examples that show that this doesn’t require low-level technological details such as programming pitfalls). Furthermore, the digital transformation provides an abundance of interesting opportunities for contextualizing CS concepts in a personally meaningful manner which should be harnessed for education.

### 2.6. Provide the basis for digital education in other subjects

As pointed out, as a consequence of the digital transformation CS is influencing other K-12 subjects as well. Therefore, as long as there is no mandatory coverage of CS skills in K-12, computing education has to provide teachers of all subjects with necessary competencies that go beyond “technological knowledge” for using media in their classroom (Koehler & Mishra, 2009), but allow them to assess and address the CS-related changes to their subject (Döbeli Honegger, 2021) and respective new topics, methods, and tools. To this end, the developments in other subjects with regards to necessary CS competencies have to be investigated together with domain experts, for example by examining changes in the corresponding scientific disciplines (Seegerer & Romeike, 2018). Such an approach will not result in a “CS light” course, but a specific selection of necessary competencies in a heavily contextualized manner (Seegerer et al., 2022) that emphasizes the specific requirements for digital education.

## 3. References


Brinda, T., & Diethelm, I. (2017). Education in the digital networked world. In *Ifip world conference on computers in education* (pp. 653–657).

Brinda, T., Fothe, M., Friedrich, S., Koerber, B., Puhlmann, H., Röhner, G., & Schulte, C. (2008). Grundsätze und standards für die informatik in der schule-bildungsstandards informatik für die sekundarstufe i.

Brown, N. C., Sentance, S., Crick, T., & Humphreys, S. (2014). Restart: The resurgence of computer science in uk schools. *ACM Transactions on Computing Education (TOCE)*, *14*(2), 1–22.

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). Computational thinking-a guide for teachers.

Denning, P. J. (2004). Great principles in computing curricula. In *Proceedings of the 35th sigcse technical symposium on computer science education* (pp. 336–341).

Döbeli Honegger, B. (2021). Covid-19 und die digitale transformation in der schweizer lehrerinnen-und lehrerbildung. *Beiträge zur Lehrerinnen-und Lehrerbildung*, *39*(3), 412–422.

Frank, H., & Meyer, I. (1972). *Rechnerkunde: Elemente der digitalen nachrichtenverarbeitung und ihre fachdidaktik*. Kohlhammer.

Grillenberger, A., & Romeike, R. (2014). Big data–challenges for computer science education. In Y. G"ulbahar & E. Karatas (Eds.), *Informatics in schools. teaching and learning perspectives - 7th international conference on informatics in schools: Situation, evolution, and perspectives* (Vol. 8730, pp. 29–40). Springer, Cham.

Jatzlau, S., Michaeli, T., Seegerer, S., & Romeike, R. (2019). It’s not magic after all – machine learning in snap! using reinforcement learning. In *2019 ieee blocks and beyond workshop* (p. 37-41).

Koehler, M., & Mishra, P. (2009). What is technological pedagogical content knowledge (tpack)? *Contemporary issues in technology and teacher education*, *9*(1), 60–70.

Michaeli, T., Romeike, R., & Seegerer, S. (2022). What students can learn about artificial intelligence - recommendations for k12 computing education. In *Proceedings of world conference on computers in education, wcce 2022.*

Michaeli, T., Seegerer, S., Kerber, L., & Romeike, R. (2022). Data, trees, and forests–decision tree learning in k-12 education. In *Proceedings of the 3rd teaching machine learning and artificial intelligence workshop, pmlr.*

Riedel, M., Streit, A., Wolf, F., Lippert, T., & Kranzlmüller, D. (2008). Classification of different approaches for e-science applications in next generation computing infrastructures. In *2008 ieee fourth international conference on escience* (pp. 198–205).

Seegerer, S., Michaeli, T., & Romeike, R. (2021). Quantum computing as a topic in computer science education. In *The 16th workshop in primary and secondary computing education* (pp. 1–6).

Seegerer, S., Michaeli, T., & Romeike, R. (2022). Foundations of computer science in general teacher education – findings and experiences from a blended-learning course. In *Proceedings of world conference on computers in education, wcce 2022.*

Seegerer, S., & Romeike, R. (2018). Computer science as a fundamental competence for teachers in other disciplines. In *Proceedings of the 13th workshop in primary and secondary computing education.* New York, NY, USA: Association for Computing Machinery.

Tolle, K. M., Tansley, D. S. W., & Hey, A. J. (2011). The fourth paradigm: data-intensive scientific discovery. *Proceedings of the IEEE*, *99*(8), 1334–1337.

Tucker, A. (2003). A model curriculum for k–12 computer science: Final report of the acm k–12 task force curriculum committee.