Bringing CS Innovations to the Classroom: a Process Model of Educational Reconstruction

Andreas Grillenberger[1], Mareen Przybylla[2], Ralf Romeike[3]

[1, 3] Computing Education Research Group, University of Erlangen-Nürnberg, Germany
{andreas.grillenberger, ralf.romeike}@fau.de
[2] Didactics of Computer Science, University of Potsdam, Germany
przybyll@uni-potsdam.de

Abstract. Computer Science continuously brings forth innovations that also contain new methods, ideas and principles. When considering these innovations from a CS education point-of-view, not only the scientific content, but also students' and teachers' perspectives and social demands should be considered to prevent just chasing trends. A promising way to prepare them for teaching is the model of educational reconstruction. In this paper, we illustrate the application of an adaptation of this model in two projects on data management and physical computing. By examples, we show how it is used to develop general guidelines and learning environments as well as concrete lessons and courses.

Keywords: Educational Reconstruction • Data Management • Physical Computing • Research Framework • CS Education • Secondary Schools

1 Introduction

Computer Science (CS) is a highly innovation-driven field. There are many developments that have obvious implications on CS practice and that thus should also play a role in formal CS education. For example, the miniaturization, efficiency and inexpensiveness of modern microcontrollers allows new use cases that are present everywhere in forms of embedded and ubiquitous computing systems. Innovations in data management allow storing and analyzing immense amounts of data and therefore open completely new possibilities, not only for business. Also, such developments are often related to one another: for instance, current innovations in networked embedded systems that can be summarized under the term "Internet of Things" (IoT) are impossible without proper methods to manage all the data that are involved in IoT applications.

In order to provide up-to-date CS education in schools that integrates every-day experiences of students and thus also promotes their motivation, current developments and innovations in CS must not be neglected. At the same time general CS education needs to focus on central ideas and concepts of the science [2, 17]. Thus, CS education research is in the tension between these innovations on the one hand and requirements for general education on the other.

When preparing the contents of innovative topics for schools, merely reducing the complexity and perceived difficulty of the subject matter is not enough. Instead, the field needs to be thoroughly examined. The central question is: How can innovations in CS be didactically prepared for teaching? One way to face these challenges is provided with the model of educational reconstruction (cf. [9]). This model, however, was developed from a natural sciences perspective and according to Diethelm et al. lacks some important aspects that are relevant in particular for CS education, such as the selection of proper phenomena to teach relevant contents [3]. They extended the original ideas with missing aspects and also take into account the general educative nature of CS education in schools. In this paper we will shortly summarize the ideas behind the model and discuss a way how it can be applied as a research framework. Finally, we will illustrate the application of this framework with two example projects on data management and physical computing, both representing current innovations in CS.

2 Educational Reconstruction

The preparation of learning contents and competence goals requires a thorough examination of the content structure and the inclusion of different perspectives on the topic. Kattmann et al. argue that central aspects of lesson planning such as the perspectives of learners are often only considered after the clarification and analysis of the science subject matter, if considered at all [9]. They see a clear gap between science education research and science instruction practice, which they seek to close with the model of educational reconstruction (MER). Here, the clarification of the science subject matter and the investigation of student perspectives both influence the design and evaluation of learning environments. This way, students' conceptions are considered and contents are related to everyday ideas and experiences of the learners.

However, as Diethelm et al. [3] point out, CS differs from other subjects in goals, knowledge structure and teaching methods. They have therefore adapted and extended the MER for CS education (MER-CSE) and illustrated some of the components with examples. In addition to the aspects mentioned in the science models, they highlight the role of context and phenomena "to motivate the students, to open connections to prior knowledge or to show application situations of the intended knowledge." [3]. This approach also ties in with the ideas of Piaget's constructivism, i. e. that learning means to build knowledge structures from interpreting new information (e. g. acquired through playing with things or reading books) based on existing knowledge and experience. Further, in the MER-CSE, social demands are analyzed for verifying the educational significance of the intended learning content.

3 Application of the MER for CS Education

The goal of research using the MER-CSE is to analyze the subject matter and develop lessons and courses or design principles for such. For this, students' and teachers' perspectives are investigated in addition to the science content structure. Diethelm et al.

highlight the role of context and phenomena for motivation and constructivist learning [3]. Further, social demands are analyzed to verify the educational significance of the intended content. We rearranged and slightly adapted the components in the graphical representation of the MER-CSE and this way transferred it to a process perspective, which is illustrated in Fig. 1. The four boxes on the left show the different perspectives (science, students, teachers and society) that are investigated in order to derive contents, contexts and phenomena suitable for CS teaching (box in the middle) as well as more general design principles (box on the right). It is noteworthy that there is no clear starting point, however all boxes should be processed at least once to design and arrange lessons and courses tailored to the needs of the particular learning groups. These learning units are evaluated together with teachers and students, so that through the reflection of their experience the single steps of the overall process can be repeated in order to adjust the resulting learning environments to the particular demands of a given setting. This is similar to design-based research in that it involves iterations with various projects in various contexts to create and constantly refine design principles and best practice examples for lessons and courses.

3.1 Underlying Perspectives

The aim of the analysis of the science content structure is to make clear which elementary ideas underlie the content in question, e. g. great principles [2] or fundamental ideas [17] and their relations; thus to provide a *science perspective* on the topic. The analysis includes the critical and methodologically sound investigation of the science content, theories, methods and technical terms. This builds the foundation to outline the field of research, to identify gaps to CS education and to illustrate the contribution to the aims of CS teaching.

Investigating social demands helps to identify contexts that are relevant for students to cope with requirements that society puts on them in their everyday lives. The significance of CS for general education is underlined from a *societal perspective*: How are jobs, everyday life and education affected? In addition to stakeholder interviews, documents that mirror society as a whole are regarded, e. g. newspapers, policy papers or project proposals. Given this general approach, social demands only need to be exemplified and reflected in larger time intervals or when outer circumstances change.

As suggested both by Kattman et al. [9] and Diethelm et al. [3], the *students' cognitive and affective perspectives* should be pervasive in all planning steps. According to

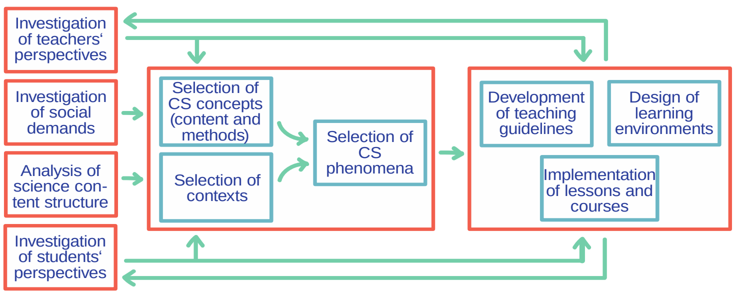

Fig. 1. Application of the MER for CS Education

Duit et al. [4], this includes pre-instructional conceptions, general cognitive abilities, interests, self-concepts and attitudes. The aim is to find out about more general perspectives of certain groups of learners and about different conceptualizations that students have when explaining CS phenomena, concepts or methods. This can be done in interviews, surveys, written tasks, etc. but also as meta-analyses of existing literature. Through reflecting learners' perspectives during several iterations of teaching, the resulting teaching guidelines, lessons and courses will be improved over time. This way, the investigation of students' perspectives helps to tailor lessons and courses to their needs in order to support learning.

Closely related to this are *teacher perspectives*, particularly focusing on their ideas about teaching, lesson planning, students' conceptions of certain CS phenomena and about their own conceptions of those phenomena. Introducing new topics to CS teaching affects three dimensions: content, tools and pedagogy. Teachers need support in acquiring new knowledge and skills in a so far unfamiliar content domain, they might have to familiarize themselves with new tools and possibly adapt to new teaching methods. This means that we have to investigate their personal attitudes towards those new elements. When it comes to implementing the ideas in lessons and courses, teachers will be able to provide answers to relevant research questions, e. g.: Which difficulties/problems can be expected and what are possible solutions? With the help of teachers, we are able to find out the most frequently used, reasonable and successful solutions.

3.2 Educational Content Preparation

The *selection of concepts* helps to focus on aspects that are interesting for students and at the same time fundamentals of the subject. Students' perceptions are important here because they tell us about their 'mental constructions' with regard to the content in question, which will affect the choice and preparation of concepts for contextualized learning. When social demands are analyzed, the outcome should be a catalogue of societal norms and requirements for CS education.

Together with the analysis of students' interests, perspectives and conceptions, an appropriate *selection of contexts* can be chosen. The aim of contextualizing learning is broadly accepted by the CS education community. In particular, with Informatics in Context (InIK), students learn in authentic contexts that help to motivate them, that show real-world applications and that offer anchor points to build on prior knowledge (cf. [3]).

A central question of the MER-CSE is, which CS phenomena can be explained with contents and methods from the subject. In the following, phenomena are understood as "occurrences of informatics in everyday life and society" [8]. They can be directly or indirectly linked to informatics systems or "have an inherent informatical structure or suggest informatical reasoning" [8]. Useful phenomena, according to Diethelm et al. [3], are all events or occurrences that are related to a specific topic and can be experienced by the learners. In our intuitive understanding, we would add that useful phenomena for CS teaching can be perceived with senses and ideally have something surprising or mysterious that is not immediately explicable by the learners and thus triggers their curiosity. We propose *deriving phenomena* in relation to concepts and contexts relevant for students.

3.3 Design and Implementation

The overall aim is to identify ideas and concepts relevant for teaching, to develop design principles and, using those, to put lessons and courses into practice. Through the reflections and feedback of all people involved, i. e. teachers, students and researchers, the implementations are revised and the guidelines and suggestions for learning environments improved over time. This way, practically usable examples, activities and materials that have been evaluated in real classroom situations are developed.

4 Research Examples

In the following, we will describe two different ways for applying this model. Both projects have in common that they introduce new, innovative topics to CS education at (secondary) school level, which entails that curricula and educational standards cannot be used for planning lessons and courses. Thus, the MER-CSE approach is very well suited in both cases, however, the projects set their focus on different aspects.

4.1 Physical Computing

This project deals with "Physical Computing" and the potentials it brings for the CS classroom. Physical computing covers the design and realization of interactive objects and installations and allows students to develop concrete, tangible products of the real world. In contrast to other hardware-centred approaches, e.g. robotics activities, it encourages learners to become creative inventors [14]. It shares concepts with embedded systems and similar technologies that are pervasive in students' everyday lives [10, 12].

Underlying Perspectives. When investigating the *science content structure* of physical computing, the perspectives of makers, interaction designers, embedded systems engineers and several more were taken into account. Physical computing integrates methods and ideas of embedded systems, cyber physical systems, interactive systems and smart objects and combines these topics with arts, crafting and engineering. In interviews with experts in the field we investigated their point of view on social impacts of embedded and ubiquitous computing systems and *social demands* that are connected to these. There was a clear tenor that in order to fully participate in our current and future society, at least a basic understanding of data collection and processing with hardware and how it influences the environment is required. This helps to understand privacy issues and to make informed decisions in many areas of our everyday lives.

To find out about *student perspectives*, we conducted a study among 115 students and found that embedded systems are not in their focus. None of them had encountered any physical computing activities in CS lessons and their interests in such activities vary a lot depending e. g. on gender [cf. 15]. We can clearly see a gap between what happens in the real world and what is taught in school. Based on these findings we developed learning environments and implemented project courses where students learn how devices interact and how to get, process and interpret real world data with sensors and actuators. In these projects we gathered additional data, such as concept maps, learner reports or video and voice recordings to extend our understanding about students' perspectives. They have only a vague understanding how hardware can be used in combination with software in order to perceive changes in the environment and interact with it (cf. [13]). To include *teachers' perspectives* regarding obstacles and concerns as well opportunities and motivational aspects, we integrated them into this research from the very beginning. We conduct professional development on physical computing, observe how they cope with the challenges, provide materials and support them in their pilot projects to analyze how their perspectives change when they gain practical experience [16].

Educational Content Preparation. Basic concepts in physical computing are sensing and reacting to events, usually changes in the environment. A context in which this is essential and that also affects students is assisted living. Although students may not directly be impacted, they can very well understand why older or impaired people need assistance. Interesting phenomena in this context are automated reactions to different events, e. g. calling keepers, when a dementia patient leaves his familiar environment or turning off household devices when no-one is at home.

Another example is wireless communication. A relevant and familiar context for all students alike, regardless of their age or gender, is traffic. In this combination of concept and context, the phenomenon of communication between cars is not explainable by using sensors alone. For instance, the questions might arise how a car can warn its driver of cross-traffic or emergency vehicles, that are not within sight.

A third example for very powerful concepts used in physical computing are timers and interrupts in the context of security in human-machine-interaction. Use cases and related phenomena are not only found in factories (e. g. assembly lines for cars), but for example also in automatically operated train systems or in arts projects like the *knife.hand.chop.bot*, a machine that plays the game of "five finger fillet" against the user. In all those cases it is crucial that the systems react immediately when body parts are in danger.

Design and Implementation. In this research, we started bottom-up with My Interactive Garden (cf. [14]), which encountered a lot of positive feedback from many experts. It has been implemented and adapted by many teachers, so that we were able to gather data in practical implementations running in parallel to the rest of the research process. This way we could investigate students' and teachers' perspectives intensively. At the same time the findings are used in addition to further systematic analyses on contents, contexts and phenomena, to improve existing lesson series, but also to develop new teaching units with learning material following typical methods of physical computing that are influenced by teachers' and students' needs.

4.2 Data Management

The topic of the second project is "Data Management" and especially the changes that come with many innovations in context of "big data". Data management evolved from the field databases and, in addition to traditional topics, incorporates recent changes and innovations. Many of these, even such that have been central to databases for years, were not examined from a didactical perspective yet. Thus, the focus of this project is on the content-oriented side of the model, i. e. to identify concepts and ideas of data management that are suitable for secondary CS education.

Underlying Perspectives. Many of the typical textbooks on databases are also fundamental to data management, which is classified in detail in the *Guide to the Data Management Body of Knowledge* [1]. Our analysis of the *science content structure* shows that new concepts have evolved (e. g. new NoSQL databases or data analysis methods), existing ideas are seen under a different light (e. g. redundancy, consistency or data quality) and the relevance of traditional topics like meta data, data privacy or data security is changing with the increasing potentials of data gathering, storage and processing. As teaching still focuses on traditional aspects like relational databases and SQL [5], changes in data management lead to a gap between what is taught in CS classes and the state-of-the-art in CS.

There is not only a gap between science and school, but also between our daily life and school: Nowadays, data analyses are often used in areas that affect us, e. g. analyses of credit card transactions to prevent fraud or user data in social media to suggest friends. We do not only come into contact with data regularly, but also produce large amounts of data. This builds a bridge to *social demands*: competencies for handling data are important, but people also need deeper knowledge about central concepts of data management to be able to manage and use data in a self-determined way (cf. [7]).

For analyzing *students' perspectives*, we used data from a study on teenagers' use of media and information [11]. It confirms the relevance of data-based technology in their life and shows that they typically use their own devices. They deal with various challenges and use a range of competencies, e. g. synchronize data between devices. Also, they are aware of risks involved when sharing information, but seem to underestimate the consequences, and they are undetermined how secure their data are in the services they use. This underlines the importance of relating central ideas of data management to the students' daily life.

Also for *teachers*, innovations and changes in data management are often hard to grasp because of their complexity and of the continuing developments in this field. We found in teacher-training workshops that depending on their experience, teachers need support in gaining expertise in this field or need materials and tools. Despite these differences, all participants confirmed the relevance of data management for CS education, even if they are complex. Also, they emphasized the need for teaching units they can flexibly adapt, as these topics are not yet part of curricula (cf. [5]).

Educational Content Preparation. Many of the concepts that are already part of database education continue to be relevant in data management teaching, but need to be considered from a wider perspective: redundancy, data schemes, relations between data, data analyses etc. But there are also concepts that are often addressed in other aspects of CS education, e. g. parallelization is typically considered in programming lessons, meta data are mainly discussed in relation to data privacy.

Various contexts can be used for linking these concepts with the students' experiences. Such contexts include, e.g. smartphones, smart homes, social networks or loyalty cards. Interesting questions in these contexts are: Why are we offered discounts when using a company's loyalty card? Why are data collections often focused on meta data instead of content? Also, there are various phenomena like potential data loss or duplicates when synchronizing data that can only be understood with knowledge about concepts like parallelization and redundancy.

Design and Implementation. This project has not been implemented as a course yet, however, a tool for analyzing a data stream using the block-based programming environment Snap! [6] was developed together with accompanying material. Using this tool, students analyze the Twitter stream and thus gain real insight into data stream analyses. Reactions have shown, that the approach to prepare smaller distinct parts of data management for teaching meets teachers desires, as they can use these modules flexibly. Also, they expected this tool to be suitable for raising students' awareness for the possibilities and threats of data analyses, e. g. in the context of social media.

5 Conclusion

In this paper we described the adaptation and application of the MER-CSE as a research framework and illustrated two different approaches for preparing new and innovative contents of CS for teaching. With a top-down approach the design and implementation of lessons and courses is clearly seen as the result of the process. While this approach is very systematic, it brings the challenge to obtain students' and teachers' perspectives and to consider the feasibility of the scenarios that are derived from the research findings. The bottom-up approach on the other hand requires a very good and usable first implementation, which is the challenge in this case to avoid random experiments in class. In turn, students' and teachers' perspectives as well as practicability can be investigated very well. The research results from the different aspects investigated with this framework need to be carefully evaluated and discussed with experts both in theory and practice and have to be considered when designing guidelines, lessons and courses. We are aware that with the examples given we have not systematically validated the framework. Nevertheless, it was shown that it is suitable for developing general guidelines and learning environments as well as concrete lessons and courses.

References

1. DAMA International: Guide to the Data Management Body of Knowledge (DAMA-DMBOK). Technics Publications, Bradley Beach N.J. (2009).
2. Denning, P.J.: The great principles of computing. Am. Sci. 98, 5, 369–372 (2010).
3. Diethelm, I. et al.: Students, teachers and phenomena: educational reconstruction for computer science education. Proceedings of the Koli Calling '12. pp. 164–173 (2012).
4. Duit, R.: Science education research internationally: Conceptions, research methods, domains of research. Eurasia J. Math. Sci. Technol. Educ. 3, 1, 3–15 (2007).
5. Grillenberger, A., Romeike, R.: A Comparison of the Field Data Management and its Representation in Secondary CS Curricula. WiPSCE 2014. ACM, Berlin (2014).
6. Grillenberger, A., Romeike, R.: Analyzing the Twitter Data Stream Using the Snap! Learning Environment. Proceedings of ISSEP 2015. Springer International Publishing.
7. Grillenberger, A., Romeike, R.: Teaching Data Management: Key Competencies and Opportunities. In: Brinda, T. et al. (eds.) KEYCIT 2014. Universitätsverlag Potsdam (2014).
8. Humbert, L., Puhlmann, H.: Essential ingredients of literacy in informatics. Informatics and Student Assessment. In: Concepts of empirical research and standardisation of measurement in the area of didactics of informatics. pp. 65–76. Gesellschaft für Informatik, Bonn (2015).
9. Kattmann, U. et al.: Educational Reconstruction – Bringing together Issues of scientific clarification and students' conceptions. Annual Meeting of the National Association of Research in Science Teaching (NARST). , St. Louis (1996).
10. Lee, E.A., Seshia, S.A.: Introduction to Embedded Systems - A Cyber-Physical Systems Approach. (2011).
11. Medienpädagogischer Forschungsverband Südwest: JIM-Studie 2015 - Jugend, Information, (Multi-) Media. (2015).
12. O'Sullivan, D., Igoe, T.: Physical Computing: Sensing and Controlling the Physical World with Computers. Thomson Course Technology PTR, Boston, (2004).
13. Przybylla, M., Romeike, R.: Concept-Maps als Mittel zur Visualisierung des Lernzuwachses in einem Physical-Computing-Projekt. In: Gallenbacher, J. (ed.) Informatik allgemeinbildend begreifen. pp. 247–256 Gesellschaft für Informatik, Darmstadt (2015).
14. Przybylla, M., Romeike, R.: My Interactive Garden – A Constructionist Approach to Creative Learning with Interactive Installations in Computing Education. In: Kynigos, C. et al. (eds.) Constructionism: Theory, Practice and Impact. pp. 395–404 (2012).
15. Przybylla, M., Romeike, R.: Overcoming Issues with Students ' Perceptions of Informatics in Everyday Life and Education with Physical Computing - Suggestions for the Enrichment of Computer Science Classes. Proceedings of ISSEP 2014. pp. 9–20 (2014).
16. Przybylla, M., Romeike, R.: Teaching Computer Science Teachers – A Constructionist Approach to Professional Development on Physical Computing. Proceedings of Constructionism 2016. pp. 265–274 Suksapattana Foundation, Bangkok (2016).
17. Schwill, A.: Fundamental Ideas of Computer Science. Bull. Eur. Assoc. Theor. Comput. Sci. 53, (1994).