Reflections on Theory

LAURI MALMI

Aalto University, Lauri.Malmi@aalto.fi

In this paper, I reflect on how my personal conceptions on "theory" have developed and become more diverse and elaborated during my career. I discuss early conceptions I learned in school and during my university studies, followed by the growing awareness of computing education research as a field that is distinct from many areas of computer science. Becoming aware of how research is carried out in social sciences and how theories are used in these contexts raised my interest in understanding what this implies for computing education research as a field, which has been building its own identity during the past 20+ years.

CCS CONCEPTS • Social and professional topics~Professional topics~Computing education

Additional Keywords and Phrases: theory, research quality, research community

Theory is a concept frequently associated with "doing scientific work". However, it is far from straightforward to fully understand and describe its various associations. In this editorial, I reflect on some of these associations that I have myself learned during my own studies and my career as a lecturer and professor, when carrying out computing education research (CER). While many of these aspects are probably familiar for most readers, I hope that this personal reflection or "my story" would spark readers to carry out their own reflections on what theory means to them.

1 THEORIES AS FORMAL ENTITIES

My first association with theory concerned natural sciences. For example, my high school physics courses discussed Newtonian mechanics in the context of motion and forces and Ohm's law in electricity. Some other briefly mentioned examples in school physics were Einstein's special and general relativity theories. In chemistry and biology, school courses discussed chemical bonds, behaviors of acids and alkalis, and Mendelian inheritance in genetics and evolution. In university courses, theories were naturally discussed in more depth, and there were laboratory exercises where empirical measurements were compared with theoretical calculations. I do not, however, recall from my studies whether there was discussion of theories as plain hypotheses of "how nature works" which could be falsified with new evidence. Theories were given "as is" and there were older and newer but more accurate theories. Some theories were called models but the conceptual difference between theories and models was not discussed.

In the university, courses in mathematics, logic, and computer science provided the second, quite a different meaning of theory. Theory is a logical claim, often called theorem, lemma or conjecture that can be formally proved based on a set of axioms. Some areas in mathematics and computer science, such as number theory, and theoretical computer science associated theory with a broader domain area of formal analysis.

My own doctoral research focused on main memory database algorithms. The algorithm design and analysis part relied on the mathematical association on theory. However, the empirical part where I carried out experiments to measure algorithm runtimes added another perspective. The formal algorithm analysis was based on using the classical random access memory model, where memory access time was

1946-6226/2022/1-ART1 $15.00
http://dx.doi.org/10.1145/3570728

considered constant. On the other hand, the empirical runtime results were obtained in an environment with multilevel caches, where memory access time was quite different from constant. There emerged major differences between "theoretical runtime behavior" and empirical observations. It became very obvious that the algorithm design and analysis should also have considered memory reference locality, and a better memory access model should have been used in the analysis. Algorithm analysis based on hierarchical memory models was, however, very new in the mid 1990s, and I did not proceed in that research.

2 THEORIES AS TOOLS TO DESCRIBE HUMAN BEHAVIOR

The third interpretation of theory, theories related to human behavior, I encountered on the pedagogical training courses I attended as a lecturer. I recall many discussions with my colleagues who had a similar background with me that textbooks dealing with theories of education were considered "weird" and hard to understand. Concepts were not formally defined, there were many different and alternative theories that could be used to guide pedagogical designs, and the same phenomenon could be explored from several different angles. Articles included a lot of discussion of different approaches to explain what happens in teaching and learning. In summary, the mindset of educational research appeared very different from what we had been trained as computer scientists or engineering scientists.

While practical application of the educational theories was in focus in the training, the given readings provided much food for thought and ideas for personal reflection. Moreover, they provided concepts, terminology, and explanations for many of the personal experiences and observations as a teacher. Later, when I got engaged with computing education research, this theoretical knowledge provided substantial guidance for the work. Some examples were, as follows. Together with two of my colleagues we developed a CS1 course based on problem-based learning, where students defined their learning goals in weekly sessions and convened to discuss what they had learned a few days later (see (Nuutila et al. 2005) for the course description and our observations on it). This supported their self-regulation skills. Programming exercises are difficult for many students because they cause high cognitive load for them. By careful exercise design, we may reduce extrinsic load while reducing intrinsic load is more difficult. Part of students' challenges is that they lack schemas in programming. Many students had a tendency for procrastination which was visible in that they started submitting their weekly exercises late (Auvinen et al. 2015). A substantial share of students aimed to get full points from exercises, even though it had no effect on their grade (90% was enough for the highest exercise grade) (Malmi et al. 2005). Thus, the system which allowed them to resubmit their solutions many times to improve their grade supported extrinsic motivation, which was likely associated with their performance orientation. On the other hand, a small group of students seemed to be tinkering, by using an extensive number of submissions while not getting top grades (Karavirta et al. 2006). Theories thus provided a common terminology to discuss pedagogical approaches and phenomena which we had observed as teachers and researchers. Moreover, they provided many plausible explanations for the observations and gave us more insight on how to improve teaching in practice.

I discovered that computing education research has clear similarities with some areas in computing research, such as HCI and empirical software engineering. The common aspect obviously is that the research concerns human behavior and data is collected from humans, though often supported with software or hardware. A common research process in CER, in addition to developing new educational innovations and/or software, includes literature review of related work, defining research questions, collecting data to address the questions, and analyzing collected data. Preferably, all these should be informed by some theoretical framework. In many areas of "hard" computing, the process includes different phases, such as setting problem statements and research goals, as well as designing and implementing a proof-of-concept solution for the addressed problem. No explicit theoretical framework is used. However, the analysis may well include formal analysis and theorem proving, in addition to empirical or simulation studies, as well as evaluation of the solution against a set of criteria based on argumentation. In summary, CER follows much more research practices in social science research even though a significant share of the work may involve software development, or tools research. Data can be collected directly from human participants or indirectly, for example, in the form of log data or submissions from the educational software tools and environments that have been used.

3 THEORY AS A FACTOR IN RESEARCH QUALITY

These reflections on the nature of research in CER emerged in parallel with some concerns of the quality of research that we had been carrying out in my team. For many years in the 2000s, I had followed and sometimes participated in discussions on the quality of research and the difference between research papers and experience reports (also called practice reports or Marco Polo papers: "I went there and saw this"). Moreover, I faced the impression from a few comments of some local colleagues that CER may not be considered "real research", and we (in my group) should be doing something more serious scientific work. It did not help that for some colleagues (as well as for myself) it was fairly easy (by that time, 15-20 years ago) to get a paper accepted into a computing education conference such that it simply presented a new software tool, learning resource or pedagogical innovation to support learning with very little, if any evaluation. Acceptance rates in most computing education publication venues were high, 30-50%, or even higher. The question was not just about my own work; I was supervising several doctoral students, and their future theses should be comparable in quality with theses from other areas of computing.

I was fortunate that professor Veijo Meisalo (now retired) from the Faculty of Education at University of Helsinki agreed to co-supervise two of my early PhD students, Sami Surakka and Päivi Kinnunen. I learned a lot of the research tradition and methods in educational sciences, including research design, process and use of theory, as well as developing theory. A further advantage was that Päivi had studied educational sciences, and not computer science, for her Master's degree, and thus had a different perspective on her PhD research concerning drop-out phenomena in CS1 (Kinnunen 2009). Her thesis included two important theoretical contributions. The feedback loop model (Kinnunen et al. 2015) presented a holistic model of how feedback can influence and guide the instructional process (setting goals – planning – teaching/studying – processing learning outcomes) not only for students and teachers but also on a teaching organization level and even on the societal level. The model made visible in a simple way the various phases in the process and interactions between the actors on different levels, thus greatly supporting the goal of understanding this complex process.

Another Päivi's theoretical contribution was the extended didactic triangle, which presents the manifold relations of central actors in the instructional process: students, teachers and learning goals/content. Using this theoretical framework, Päivi, prof. Meisalo and I analyzed research literature in CER to discover what aspects of the instructional process had been addressed widely and which had received little attention either at the classroom level, teaching organization level or societal level (Kinnunen et al. 2010). Compared with most literature reviews, where categories of research content are formed in a data driven way, our theoretical framework enabled us to better identify research gaps, i.e. relations of actors which the theory informs that there could or even should be and where little or no data (publications) were found. This theoretical tool has turned out to be a very useful analysis tool later when we have analyzed literature not only in CER but in engineering education research and science education research (Kinnunen et al. 2013, 2016; Lampiselkä et al. 2019).

4 WIDER PERSPECTIVES

These examples of the usefulness of theories and theoretical models in educational research have, on their part, inspired my interest in building a holistic picture of the international CER field. Together with several colleagues with a similar interest, we have sought answers to What do we research in the field and where are the gaps (Kinnunen et al. 2010), How the research is carried out (Malmi et al. 2010), and finally How theories and models are used and developed in the field (Malmi et al. 2014). My motivation for these analyses concerned that, perhaps, this collected knowledge could be used to support research training in the field, as well as raising a better awareness of quality aspects in research among the CER community.

My further work related to theories has led to several review papers where I and my colleagues, Judy Sheard, Simon, Päivi Kinnunen and Jane Sinclair have analyzed a substantial share of CER papers (Malmi et al. 2019, 2020, 2022). The last paper is included in this same special issue. During the same years, similar reviews have been carried out by several other authors (Lishinski et al. 2016, Szabo et al. 2019, Szabo & Sheard 2022); the last one is also included in this special issue. Additional reviews on more focused theoretical areas are included in the Cambridge Handbook of Computing Education Research (2019) focusing on cognitive sciences (Margulieux et al. 2019), learning sciences (Robins et al. 2019), as well as motivation, attitudes and dispositions (Lishinski & Yadav 2019). And naturally, several papers in the first special issue on theory in TOCE (Issue 4 / 2022) included focused reviews of specific theories. These reviews are not just a sign of a few individuals' interest in theory. Rather, the work is driven by the motivation to better understand our field, CER, as a whole and how research could be improved through richer and more appropriate use of theoretical frameworks.

Personally, my recent interest has focused especially on domain-specific theories in CER; I am convinced that such theories are needed to build a deep understanding on the complex phenomena related to teaching and learning computing concepts and processes. Theories from education, psychology and other social sciences are highly valuable tools for our research, but as they are domain general, it is hard to address in depth many challenges which we face in teaching and learning computing, especially programming, e.g., how students understand and conceptualize notional machines or program design.

In our reviews (Malmi et al. 2019, 2020, 2022), we have faced the challenge of defining what we mean with the concept "theory". We have not identified any domain-specific theories that can be used to address different phenomena broadly in the field, compared with domain-general theories, such as, cognitive load theory, self-regulation theory or self-efficacy theory which can be applied widely. Rather, our findings have revealed numerous small-scale theoretical developments in the form of, e.g., statistical models, grounded theories, phenomenographical outcome spaces or taxonomies. Most of these can be considered context-specific theories or models, as replication studies are rare in our field, and research which seeks to validate previous theoretical findings in other contexts is even more rare. What should we call these developments? Theory may often seem a too bold term; many of them could be called models, and in some cases theoretical framework seems to match. A further complication is that these terms do not have a commonly agreed definition; rather they are often used quite loosely in the literature. In our analyses, we therefore concluded to use our own new term (!) theoretical construct which covered all of them, and even theory-informed instruments. The terminology forms a challenge. When lacking a commonly agreed definition, researchers make personal associations with the terms they encounter, and this may lead to misinterpretations between them. I recommend reading the paper by Tedre et al. in this special issue to find some in-depth discussion of terms and the nature of theoretical development.

While the first special issue on theory focuses on single theories or families of theories, presenting their main content and application examples in the CER domain, this second special issue considers theories on a meta level, as objects of investigation, thus addressing questions, such as, what characteristics these objects have, how they were developed, how they could be categorized, as well as how they are used either alone or together with other theories. These papers thus extend the area I have been interested in by building a more comprehensive and holistic view of the use of theories in CER.

However, in this phase when the issue is ready for publishing, it is worth reflecting on the presented contributions, and ask: who needs this kind of perspective and what benefits it might provide? What I would answer (currently — my thinking evolves) is, as follows. Firstly, theories are tools or lenses which can support our investigation of specific questions; this is obvious in most cases. Secondly, theories in social sciences are many and we as researchers in our domain area, computing education, should take the perspective that we seek to find, apply or develop theories which match well our specific research domains. Obviously, this is not a trivial task, and few, if any, people in the CER community can claim that they master widely the available theories in social sciences. We hope that papers in this second issue will provide some new insights for researchers when considering which theories or models might be used in their current or future research contexts, and whether building new or extending current theories or models would be a feasible goal.

5 ARE WE MAKING PROGRESS AND WHAT IS PROGRESS?

One of the findings in our review paper on domain-specific theories in CER in this issue is that while many theoretical constructs have been developed, most of them are rarely used to inform further research. This seems concerning. Are we making progress or are we reinventing wheels or just building new types of wheels? More concretely, let us consider research in programming education. The field has carried out over 50 years of studies related to how people learn programming. Have we been able to build an advanced theory of learning programming, which steadily improves and provides more accurate explanations and predictions on students' progress? I fear the answer is negative. We certainly know much more about learning programming, but this knowledge is very much scattered in results from numerous different contexts and approaches.

The previous observations may seem discouraging. However, it is worthwhile to take a critical look at them. During these 50 years, programming technologies have continuously developed in different directions in terms of programming languages, their features, available libraries, and programming tools, and this development is not likely to cease. This implies that our research concerns a moving target. When CS1 was taught in Algol, Fortran or Pascal in the 1970s and 1980s with command line environments or using punched cards, it was a very different experience for students compared to courses today, which use IDEs and vast libraries in, e.g., Java, C#, Python or Scala or use some block-based language. While certainly many concepts, such as variables, conditionals or loops are conceptually the same or similar, it is not self-evident that we can build on all research which concerned learning introductory programming in the 1970s assuming that the same results and conclusions hold now. Replicating old works would be tempting, but there are many challenges, as the working environment is so different now. Moreover, the current student generation has different conceptions of computers and different IT skills as their background which further complicates interpretation and comparison of results. Therefore, the goal of building a "uniform theory of teaching and learning programming" seems unrealistic.

Another challenge is that how do we even define "progress" in this research context, and how could we measure it? In natural sciences, progress is often presented as new theories and models which allow us to make more accurate predictions of future events. In social sciences, this is not the only way of measuring progress. Rosenberg, in his book on philosophy of social sciences (2008), presents an alternative point of view. He notes that if we seek causes and consequences among human actions, we must acknowledge that our actions are determined by our "beliefs, desires, expectations, preferences, hopes, fears, wants that make actions meaningful or intelligible to ourselves and one another." (p. 17). Rosenberg discusses the concept "common sense" or "folk psychology" which we learn when we grow in a society. It informs us of many obvious things from ourselves and others that we regularly use to predict and explain our own and others' behaviors. But common sense is not a theory that might be abandoned. It is a must for us to be able to live in a society.

Rosenberg makes an important claim. *"But unlike natural science in relation to its theories, the main aim of social science is not to increase the predictive power of folk psychology. Rather, the aim is to extend folk psychology from the understanding of everyday interactions of individuals to the understanding of interactions among large numbers of individuals in social institutions and among individuals whose cultures and forms of life are very different from our own."* (p. 22).

I would interpret this view of progress, when applied to CER, that progress denotes our increasing understanding of how learning and teaching computing takes place in varying and diverse contexts. From this perspective, we have made much progress in programming education research. Moreover, much work in developing computing education seeks to make the learning content meaningful and intelligible for students.

6 WHERE I AM NOW?

When looking back at how my conceptions on theory have developed over time, one clear trend has been the growing understanding that theory has many and diverse interpretations in sciences. Moreover, this diversity is not limited only to theory; rather the research paradigms, how research is carried out and what is being valued in research are diverse. Interestingly, I do not recall that at any time during my formal studies this diversity was explicitly addressed, but I have discovered it only during my academic career. An interesting question emerges: Would it be actually feasible to build a research training course for doctoral students that would successfully discuss the scope of different conceptions on research? Or, can such an understanding be gained only in real academic work when carrying out research in different and multidisciplinary settings?

I finished my master's degree at a university of technology, where the engineering mindset dominated my degree program and my specialization in software technology. Thus, we learned to identify and solve problems seeking to design and implement feasible solutions. This mindset very much carried over to my teaching career: If I as a teacher or my students had some challenges or difficulties, my goal was to design and implement solutions to them, often by building software tools or innovating new pedagogical solutions and testing them in real courses. In this mindset, theories are considered as tools, which can support identifying the problems, guide designing solutions to them, and help in interpreting the evaluation findings. Moreover, as a teacher and researcher, I can select those theories which seem to match my educational context and goals.

While this may seem straightforward, there are issues involved. When making decisions on how to address emerging problems in the current educational context, the decisions are often not explicitly informed by theory. Instead, they are often based on my own or my colleagues' experiences as a teacher, when recalling past observed problems and results of interventions, as well as my personal expectations of what students know and how they study. My thinking may be based on some implicit theoretical assumptions, but I often fail to make them explicit for myself. The case is different if I am planning a research intervention and making a study. However, selecting a theoretical framework is only possible among the theories that I am aware of, and thus I might miss approaches which would better match my context and goals. I was happy to read the papers in the first special issue of theories which thus provided several new opportunities for consideration. Finally, research is a long-term endeavor, and gaining an in-depth understanding of some problem area would require many years of studies using theories or their combinations to address the phenomena and guide the research. Too often, research is scattered, driven by the interests of current doctoral students, available funding opportunities and personal time. Moreover, while large courses provide good opportunities for making interventions and collecting large data sets, only very few interventions can be carried out in parallel, as they would interfere with each other and increase the complexity of the course for both the teacher and the students. Thus, we are not free to design research using the theoretical "tools" available for us.

A more subtle perspective is that theories have some ontological and epistemological assumptions, which may be relevant. Yet, I have rarely explicitly considered these aspects in my papers, but simply taken the pragmatist view. In qualitative analysis, I — often working together with colleagues – must accept that the results are not “objective truth” but include my or our interpretation of the data and findings. We negotiate with the theory if there is some theory involved in the analysis. In quantitative analysis, the process seems slightly different. The results may or may not seem to match with the theory (if theory is involved in the research design – much work has been exploratory without an explicit theory consideration in the beginning). The latter case, where the findings do not match with the theory, naturally raises the question of why this happened. Is it due to poor research design, too small or biased data or too many intervening variables? Often, we can only conclude that we do not know and “more research is needed”, but it remains somewhat unclear how we should proceed. Perhaps, the selected theory even does not apply in our specific context, as it was derived in another educational context which was too different from ours.

Finally, Rosenberg’s book was my first encounter with philosophy of social sciences, and it opened to me some totally new insights on research, which I only very briefly reflected above. Should I once more rethink my conception of theory and how research makes progress? I do not know yet. However, it seems there is much more that this branch of philosophy could provide for CER; but this is out of scope here, and there is much more for me to learn about it before I would dare to write more.

I hope you enjoy this second special issue.

REFERENCES

Auvinen, T., Hakulinen, L., & Malmi, L. (2015). Increasing students’ awareness of their behavior in online learning environments with visualizations and achievement badges. *IEEE Transactions on Learning Technologies*, 8(3), 261-273.

Fincher, S. A., & Robins, A. V. (Eds.). (2019). *The Cambridge handbook of computing education research*. Cambridge University Press.

Karavirta, V., Korhonen, A., & Malmi, L. (2006). On the use of resubmissions in automatic assessment systems. *Computer science education*, 16(3), 229-240.

Kinnunen, P. (2009). Challenges of teaching and studying programming at a university of technology-Viewpoints of students, teachers and the university. Doctoral thesis. Helsinki University of Technology.

Kinnunen, P., Lampiselkä, J., Meisalo, V., & Malmi, L. (2016). Research on teaching and learning in Physics and Chemistry in NorDiNa Papers. *NorDina: Nordisk tidsskrift i naturfagdidaktikk*.

Kinnunen, P., & Malmi, L. (2006). Why students drop out CS1 course? In *Proceedings of the second international workshop on Computing education research* (pp. 97-108).

Kinnunen, P., & Malmi, L. (2013). Pedagogical focus of recent engineering education research papers. In *Proceedings of the Annual SEFI conference 2013* (pp. 16-20).

Kinnunen, P., Meisalo, V., & Malmi, L. (2010). Have we missed something? Identifying missing types of research in computing education. In *Proceedings of the Sixth international workshop on Computing Education Research* (pp. 13-22).

Kinnunen, P., Meisalo, V., & Malmi, L. (2015). Feedback loop model-a tool for systematic analysis of challenges of instructional processes in science education. *Proceedings of ESERA 2013, European Science Education Research Association, Part 11 Strand 11 Evaluation and assessment of student learning and development,* pp. 1725-1732.

Kinnunen, P., & Simon, B. (2010, August). Experiencing programming assignments in CS1: the emotional toll. In *Proceedings of the Sixth international workshop on Computing education research* (pp. 77-86).

Lampiselkä, J., Kaasinen, A., Kinnunen, P., & Malmi, L. (2019). Didactic focus areas in science education research. *Education Sciences*, 9(4), 294.

Lishinski, A., Good, J., Sands, P., & Yadav, A. (2016, August). Methodological rigor and theoretical foundations of CS education research. In *Proceedings of the 2016 ACM conference on international computing education research* (pp. 161-169).

Lishinski, A., & Yadav, A. (2019). Motivation, Attitudes, and Dispositions. (pp. 801-826). In: Fincher, S. A., & Robins, A. V. (Eds.). *The Cambridge handbook of computing education research*. Cambridge University Press.

Luxton-Reilly, Andrew, Ibrahim Albluwi, Brett A. Becker, Michail Giannakos, Amruth N. Kumar, Linda Ott, James Paterson, Michael James Scott, Judy Sheard, and Claudia Szabo. "Introductory programming: a systematic literature review." In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, pp. 55-106. 2018.

Malmi, L., Karavirta, V., Korhonen, A., & Nikander, J. (2005). Experiences on automatically assessed algorithm simulation exercises with different resubmission policies. *Journal on Educational Resources in Computing (JERIC),* 5(3), 7-es.

Malmi, L., Sheard, J., Bednarik, R., Helminen, J., Korhonen, A., Myller, N., Sorva, J. & Taherkhani, A. (2010, August). Characterizing research in computing education: a preliminary analysis of the literature. In *Proceedings of the Sixth international workshop on Computing education research* (pp. 3-12).

Malmi, L., Sheard, J., Bednarik, R., Helminen, J., Kinnunen, P., Korhonen, A., Myller, N., Sorva, J. & Taherkhani, A., 2014, July. Theoretical underpinnings of computing education research: What is the evidence? In *Proceedings of the tenth annual conference on international computing education research* (pp. 27-34).

Malmi, L., Sheard, J., Kinnunen, P., & Sinclair, J. (2019, July). Computing education theories: What are they and how are they used? In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (pp. 187-197).

Malmi, L., Sheard, J., Kinnunen, P., & Sinclair, J. (2020, August). Theories and models of emotions, attitudes, and self-efficacy in the context of programming education. In *Proceedings of the 2020 ACM conference on international computing education research* (pp. 36-47).

Malmi, L., Sheard, J., Kinnunen, P., & Sinclair, J. (2022). Development and Use of Domain-Specific Learning Theories, Models and Instruments in Computing Education. *ACM Transactions on Computing Education (TOCE).*

Margulieux, L. E., Dorn, B., & Searle, K. A. (2019b). Learning sciences for computing education (pp. 208-230). In: Fincher, S. A., & Robins, A. V. (Eds.). *The Cambridge handbook of computing education research*. Cambridge University Press.

Nelson, G. L., & Ko, A. J. (2018, August). On use of theory in computing education research. In *Proceedings of the 2018 ACM Conference on International Computing Education Research* (pp. 31-39).

Nuutila, E., Törmä, S., & Malmi, L. (2005). PBL and computer programming—the seven steps method with adaptations. *Computer Science Education*, 15(2), 123-142.

Robins, A. V., Margulieux, L. E., & Morrison, B. B. (2019). Cognitive sciences for computing education. In: Fincher, S. A., & Robins, A. V. (Eds.). *The Cambridge handbook of computing education research*. Cambridge University Press.

Rosenberg, A., *Philosophy of Social Science*, 3rd edition, (2008). Westview.

Szabo, C., Falkner, N., Petersen, A., Bort, H., Cunningham, K., Donaldson, P., Hellas, A., Robinson, J. & Sheard, J., (2019). Review and use of learning theories within computer science education research: primer for researchers and practitioners. *In Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education* (pp. 89-109).

Szabo, C., & Sheard, J. (2022). Learning Theories Use and Relationships in Computing Education Research. *ACM Transactions on Computing Education (TOCE*).