COMPUTING EDUCATION RESEARCH

Lauri Malmi

Can We Show an Impact?

DOES COMPUTING EDUCATION RESEARCH (CER) have an impact on computing education practice? I have heard the question many times, and certainly it is a relevant question for the CER community. But how can we answer the question?

We can look at the question from different perspectives, from an individual teacher's point of view of adopting results from CER, and from the wider community's perspective of disseminating our work. Let me start with the first one.

For a computer science (CS) teacher, taking a research perspective on his/her own courses provides several benefits. First, following the literature—a central part of research—raises awareness of related work, and provides the possibility of adopting new methods, tools or results from other researchers' work, often supported by some evidence on their impact on students' learning. When we carry out any kind of research, building on others' previous work is natural. Interestingly, this is not a default attitude in teaching—too often we start developing our own pedagogical approaches from scratch (perhaps, supported by our predecessors' course material). We should take the research approach instead.

Second, theories and research from the disciplines of education and psychology can provide us valuable insights into the learning process and help us understand students' problems and then design better solutions to the problems. I have currently been supervising a rather exceptional master's thesis project where the student dug into cognitive learning theories, how they could be used to explain novice programming students' behavior and dropout phenomenon. Truly, I learned some new insights that I had not thought before. I have encouraged him to publish this work.

Third, designing and implementing an appropriate evaluation study to investigate the effect of a novel teaching method, learning resource, or tool can provide us with deeper information about the success, as well as limitations, of our novel approach. Of course, for a faculty member teaching a class of 20 students and following and getting to know each student individually, perhaps, over several courses, provides very rich information about students' problems and how they could be solved, without doing any research. However, not all CS teachers have such a luxurious teaching environment. Last spring I was teaching a data structures and algorithms courses with some 250 participants. From my perspective, almost all students were just numbers in the course result tables, and I had no personal contact with them.

> Does computing education research (CER) have an impact on computing education practice? … [And] how can we answer the question?

In summary, CER clearly has potential to affect an individual teacher's work. But this only happens for teachers' who take a research perspective on their teaching and very often these people already undertake CER. What about the impact on others? Are our teaching innovations, tools, and research results known and adopted among the larger computing education community? Although this may seem to be an obvious question, how we could seek an answer to it does not seem at all straightforward.

Our artifacts are publications, where we report on our innovations, concrete tools, i.e., educational software or hardware, and other learning resources that we have developed as a part of our research. In research, counting citations is a common method to measure impact. However, it is questionable whether citation counts of most CER papers measure anything more than the visibility of the work within the CER community itself. The number of downloads from ACM digital library may tell something more, but there is no information about who is downloading the papers. Despite this rather disappointing conclusion, I made a small survey among these numbers, hoping that the most commonly cited/downloaded papers might have influence on CS practitioners.

For simplicity, I used Google Scholar for the citations. Because programming education is the largest subarea in CER, I chose as the starting point the ITiCSE 2007 working group paper [2], where we had solicited suggestions from our CER colleagues to identify a set of papers as recommended readings for programming educators. Among the candidates, we chose 45 papers for the list. Of course, many important papers have been published after our survey, but it is reasonable to assume that this set would reveal something about citations counts for important papers.

The survey itself has only been cited 160 times (by November 15th, 2014). Two most cited papers in the list were Robins et al., "Learning and teaching programming: A review and discussion"

(2003) with 542 citations, and Naps et al., "Exploring the role of visualization and engagement in computer science education" (2002), with 471 citations.[i] Only a small minority reached over 100 citations; most had less than 50. A few other papers,

Some numbers are available but it seems unclear whether they are valid measures of impact in the sense of adopting and applying results from CER in practical teaching. What other measures could we use?

not on our list, like the famous working group reports by McCracken et al., "A multi-national, multi-institutional study of assessment of programming skills of first-year CS students" (2001), and Lister et al., "A multi-national study of reading and tracing skills in novice programmers" (2004), reached similar numbers (502 and 304 correspondingly).

As mentioned, downloads offer another and perhaps better measure. Such information is not, however, available for all papers on our list. From ACM Digital Library, we see that our survey has been cumulatively downloaded 4207 times, and two other most downloaded papers were Kelleher et al., "Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers" (2005), with 10724 downloads, and Ben-Ari's SIGCSE paper, "Constructivism in Computer Science Education" (1998), with 4,032 downloads. For most papers on our list the digital library download counts were a few hundred only. Though the numbers are bigger than citation counts, there is still a long leap to conclude that the papers have really initiated a change in the readers' teaching practices. We simply do not know.

In tools research, download counts could also be used as a dissemination indicator. However, they do not separate between trial users and production users. Tool developers can build better means of tracking the actual use of their software, but such information is seldom publicly visible. Some tool developers have formed user communities, such as Greenroom for Greenfoot [1] users or the Web-Cat community [3]. The former reports 3125 educators and the latter more than 50,000 members, but how many are active users (and how activity is defined) and who just visited the site once, is not visible.

To conclude, some numbers are available but it seems unclear whether they are valid measures of impact in the sense of adopting and applying results from CER in practical teaching. What other measures could we use? Number of computing education conference participants? Number and scale of projects and initiatives which are based on CER? Career tracks of CER graduates? Doing annual surveys among SIGCSE list? Counting references to CER papers and tools from university pedagogy courses?

Perhaps, we should start a research project to identify what kind of impact there could be and how to generate valid measures to evaluate it. **Ir**

References

[1] Greenfoot; www.greenfoot.org. Accessed 2014 December 20.

[2] Pears, A. et al. "A survey of literature on the teaching of introductory programming." *SIGCSE Bulletin* 39, 4 (2007): 204-223.

[3] Web-Cat; http://sourceforge.net/projects/web-cat/. Accessed 2014 December 21.

Lauri Malmi
Department of Computer Science and Engineering
Aalto University
P.O. Box 15400
00076 AALTO Finland
Lauri.Malmi@aalto.fi

DOI: 10.1145/2727129

[i] See [2] for the full citations of the most cited papers.