Doctoral Work in Computing Education Research: Beyond Experimental Designs

Having recently participated in the Doctoral Consortium of the Australasian Computing Education Conference I was struck by the core methodological challenges faced by students. Initial proposals seemed to be along the lines of, "well I will need a control group and a comparison group to evaluate x or y condition or intervention." The corollary to this thinking then became "well how many students will be enough?" presumably so the findings could be generalisable in some manner. Yet do methods drawn from the natural sciences really furnish the most appropriate approaches for computing education research (CER), with its hybrid concerns of technology and people? If not, then which methods would be more suitable and how can students be guided towards their adoption?

I suppose the typical preparation of computing students is a fairly technical one and the more obvious scientific research methods to which they have been exposed will probably have been *experiments* or perhaps *simulations* of some form. Yet the literature on CS research methods suggests that even these classic scientific approaches are relatively uncommon (less than 6% for instance for both experiments and simulations in [1]), and the absence of experimentation was actively criticised in [2]. Given that the "dominant research approach in both the CS and SE fields is '*formulate*' – that is, the author is creating some product (algorithm, process, guideline, etc.) presumably of value to others in the field" [3], how that might translate to a research method in computing education is probably unclear to students. Then further as observed in [3], "The primary research method of CS was '*conceptual analysis*' based on a mathematical foundation. More interestingly, the primary research approach of SE was also *conceptual analysis*, but without a mathematical flavor", and as further noted "neither CS nor SE does much *evaluative* work", which tends to be more common in the IS discipline. So it could be said that the primary research methods in the more technical computing disciplines tend to be rather inexplicit and opaque, or either about formulating or building something or conceptually analyzing an artifact, process, model or phenomenon.

Therefore, it is little wonder that students undertaking doctoral projects in the sub discipline of computing education research (CER) tend to flounder. For how could a computing education research study that conducted no evaluation of its findings be termed rigorous? Merely building a new tool, or proposing a model would not suffice. Given the limited prior exposure to methods with a strong element of evaluation it is no wonder then that an experiment of some kind would be the method of refuge. This lack of explicit coverage of method has led the SIGCSE community to support an ITiCSE working group report on how to teach research methods in computing at postgraduate level [4]. For the sub discipline of computer science education research it spawned the excellent book by Sally Fincher and Marian Petre [5] as a primer to doing research in the field. An analysis of the types of research being conducted into computing education has been presented relatively recently at the ICER conference [6].

So there are some valuable resources which can give students and their supervisors insight into approaches they might adopt. Or for that matter, CS educators conducting research into their teaching or the learning of their

students may benefit from greater methodological rigor in evaluating their work. Thereby they may move beyond what Valentine [7] has termed "Marco Polo" presentations describing a personal journey such as; "how their institution has tried a new curriculum, adopted a new language or put up a new course". As Valentine critically elaborates "The reasoning is defined, the component parts are explained, and then (and this is the giveaway for this category) a conclusion is drawn like 'Overall, I believe the [topic] has been a big success.' or 'Students seemed to really enjoy the new [topic]."

As noted in [8] a number of elements and perspectives on reality and the construction of knowledge serve to frame a computing education research design. One useful way of framing their study, which I like to discuss with doctoral students, is the nature of theory that may be derived from their research. Shirley Gregor has proposed a thought provoking framework on the nature of theory in the IS discipline [9]. She classifies theories in a taxonomy incorporating five types: Type I a theory for analyzing; Type II a theory for explaining; Type III a theory for predicting; Type IV a theory for both predicting and explaining and Type V a theory for Design and Action. These are tabulated below with their distinguishing characteristics and mapped to illustrative examples of CER studies of each type.

Gregor's Taxonomy of Theory Types applied to CER Research		
Theory Type	Distinguishing Characteristics	CER Example
1. Analysis	***Says what is*** The theory does not extend beyond analysis and description No causal relationships among phenomena area specified and no predictions are made	Sheard et al.'s data derived framework for assessing examination complexity [10]
2. Explanation	***Says what is. how why, when and where*** The theory provides explanations but does not aim to predict with any precision. There are no testable propositions.	Eckerdal & Thuné's phenomenographic study of how students understand class and object [11]
3. Prediction	***Says what is and will be*** The theory provides predictions and has testable propositions but does not have well developed justificatory causal explanations.	Lopez et al.'s path analysis of the relationship between code reading and writing by novice programmers [12]
4. Explanation and prediction	***Says what is. how why, when, where and what will be*** Provides predictions and has both testable propositions and causal explanations	Mazur's work in Physics education on peer instruction [13] since adopted in CS contexts [14]
5. Design and action	***Says how to do something*** The theory gives explicit prescriptions (e.g. methods, techniques, principles of form and function), for	Denny & Luxton-Reilly's paper on the design of the Peerwise system [15]

	constructing an artifact	

Table 1: Gregor's Taxonomy of Theory Types applied to CER Research [Adapted from 9]

As can be seen from the table above, there are many ways of designing research studies in CER, which extend well beyond the experimental design. However, a key driver for the research design is the question relating to what sort of theory the study is aiming to derive. If the goal is a *theory for explanation and prediction*, then the painstaking, data driven and longitudinal work of scholars like Eric Mazur (who was the ICER2012 keynote speaker) provides one example of an approach. For most time constrained doctoral studies in CER (or for that matter many CER projects) a less ambitious design would be prudent. It would be wise to spend time on devising clear goals from the outset. Thereby the choice can be consciously made of which theory is motivating the study. Choice of a theory for *analysis* or *explanation*, or *design and action* with sound evaluation of impact [16] can also make a valid and useful contribution.

References:

1. Ramesh, V., Glass, R., & Vessey, I. "Research in computer science: an empirical study". *The Journal of Systems & Software, 70* (2004): 165-176.
2. Tichy, W. F. "Should computer scientists experiment more?" *Computer, 31*(5) (1998): 32-40. doi:10.1109/2.675631.
3. Glass, R., Vessey, I. and Ramesh, V. "RESRES: The story behind the paper *Research in software engineering: An analysis of the literature*." *Information and Software Technology*, 51 (2009): 68-70.
4. Holz, H.J., Applin, A., Haberman, B., Joyce, D., Purchase, H. and Reed, C. "Research methods in computing: what are they, and how should we teach them?" in *Working group reports on Innovation and technology in computer science education*. (Bologna, Italy: ACM, 2006): 96-114.
5. Fincher, S. and Petre, M. *Computer Science Education Research: The Field and The Endeavour*. (London: Routledge Falmer, Taylor & Francis Group, 2004).
6. Malmi, L., Sheard, J., Simon, Bednarik, R., Helminen, J., Korhonen, A., Myller, N., Sorva, J. and Taherkhani, A. "Characterizing research in computing education: a preliminary analysis of the literature." in *Proceedings of the Sixth international workshop on Computing education research*. (Aarhus, Denmark:ACM, 2010): 3-12.
7. Valentine, D.W. "CS educational research: a meta-analysis of SIGCSE technical symposium proceedings." in *Proceedings of the 35th SIGCSE technical symposium on Computer science education*. (Norfolk, Virginia, USA: ACM, 2004): 255-259.

8. Thota, N., Berglund, A. and Clear, T. Illustration of Paradigm Pluralism in Computing Education Research. in *Proceedings of the Fourteenth Australasian Computing Education Research Conference.* (Melbourne, Australia: ACS, 2012): 103-112.
9. Gregor, S. "The Nature of Theory in Information Systems." *MIS Quarterly*, *30*, 3 (2006): 611-642.
10. Sheard, J., Simon, Carbone, A., Chinn, D., Clear, T., Corney, M., D'Souza, D., Fenwick, J., Harland, J., Mikko-Jussi Laakso and Teague, D. "How difficult are exams? A framework for assessing the complexity of introductory programming exams." in *Proceedings of the Fifteenth Australasian Computing Education Research Conference*. (Adelaide, Australia: ACS, 2013): 145-154.
11. Eckerdal, A. and Thuné, M. "Novice Java programmers' conceptions of 'object' and 'class', and variation theory". *SIGCSE Bulletin*, 37, 3 (2005): 89-93.
12. Lopez, M., Whalley, J., Robbins, P. and Lister, R. "Relationships Between Reading, Tracing and Writing Skills in Introductory Programming." in *the Fourth International Computing Education Research Workshop.* (Sydney Australia, ACM, 2008): 101-111.
13. E. Mazur. *Peer Instruction: A User's Manual*. (Upper Saddle River, NJ: Prentice Hall, 1997).
14. Simon, B., Kohanfars, M., Lee, J., Tamayo, K. and Cutts, Q., "Experience report: peer instruction in introductory computing." in *Proceedings of the 41st ACM technical symposium on Computer science education.* (Milwaukee, Wisconsin: ACM, 2010): 341-345.
15. Denny, P., Luxton-Reilly, A. and Hamer, J., "The PeerWise system of student contributed assessment questions." in *Tenth Australasian Computing Education Conference.* (Wollongong, NSW, Australia: ACS, 2008): 69-74.
16. Bain, J. "Introduction (to the special Issue on Evaluation)." *Higher Education Research & Development*, *18*, 2 (1999): 165-172.

Categories and Subject Descriptors: K.3.2 [Computers and Education]: Computer and Information Science Education – Computer science education, information systems education.
General Terms: Experimentation, Theory
Keywords: Computing Education Research, Research Approaches, Research Methods, Experimental Designs, Theory Types