In the liminal space: software design as a threshold skill

Lynda Thomas, Aberystwyth University, UK
Jonas Boustedt, University of Gävle, Sweden
Anna Eckerdal, Uppsala University, Sweden
Robert McCartney, University of Connecticut, USA
Jan Erik Moström, Umeå University, Sweden
Kate Sanders, Rhode Island College, USA
Carol Zander, University of Washington Bothell, USA

"Threshold Skills" have been proposed as a complement to Threshold Concepts [Thomas et al. 2012, Sanders et al. 2012]. While students sometimes experience difficulties with Threshold Concepts, at other times they struggle due to lack of a skill or an inability to combine theoretical with practical knowledge. Improved skills can improve conceptual understanding and vice versa. See, for example, Eckerdal's discussion [2009] of the concepts and practice of programming.

Threshold Skills have many of the same characteristics as Threshold Concepts, with a slightly different focus. They are *troublesome* - they may seem strange, alien, or counterintuitive to the student. They are *transformative* – but they transform what students can do and their vision of what they can do, as opposed to how they see their discipline. They are *integrative* – but rather than unifying different concepts, they broaden the list of tasks students can perform. They are *semi-irreversible* – they do not completely go away but they do degrade over time with lack of use. Unlike Threshold Concepts they *must be practiced*.

Here, we apply this definition to software design: the phase of software development that takes a description of *what* is to be built and creates a detailed description of *how* it should be built. Designers must break a problem down into parts and describe solutions to those parts and how they will fit together, using diagrams and other computing language.

We asked a group of graduating computing students to 'design a super-alarm clock that University students could use to manage their sleep patterns'. This problem has been intensively studied [Eckerdal et al. 2006] [Loftus et al. 2011]. By examining our students' designs, we were able to see both *skill* – what they *do* when asked to design – and something of what they understand of the *concept* of design.

Although the student designs were better than those observed in previous studies (74% had at least a first step towards a design and 11% had designs judged 'complete', as opposed to 38% and 2% in the original study) most missed some critical part, failed to link parts, or failed to use correct professional language. The student designs displayed characteristics that could be expected of people in a liminal space [McCartney et al. 2009]: *partial understandings* of the nature of software design and *mimicking* the notation and language used by professional designers. The weakest students produced what we would not categorise as 'software' designs at all – some submitted attractive drawings of a clock or clock-face - these students are failing to observe the *boundaries* of the field.

Is software design a Threshold Skill? Overall, increased design skill is both transformative (it changes the students' understandings of what they can do) and integrative (the design for one program can be adapted and re-used for another). In terms of reversibility, there may be an issue with granularity: part of design – fluency in computing language and software design notation – is semi-reversible, but the ability to break down a problem into parts and see how a solution can be built from solutions to those parts may be irreversible. From the comparison of the results of this study

with other studies that used less prepared students, students improve with practice; yet software design continues to be troublesome and to serve as a boundary marker in the computing field. While learning software design, students exhibit some of the characteristics of being in liminal space.

References

[Eckerdal et al. 2006] Eckerdal, A., McCartney, R., Moström, J. E., Ratcliffe, M., and Zander, C. Can graduating students design software systems? In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education.* 403-407. (2006).

[Eckerdal 2009] Novice Programming Students' Learning of Concepts and Practice, Acta Universitatis. Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 600. 194pp. Uppsala. (2009).

[Loftus et al. 2011] Chris Loftus, Lynda Thomas, and Carol Zander. 2011. Can graduating students design: revisited. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education.* 105-110.(2011).

[McCartney et al. 2009] McCartney, R., Boustedt, J., Eckerdal, A., Moström, J.E., Sanders, K., Thomas, L. and Zander, C. Liminal spaces and learning computing, *European Journal of Engineering Education*, 34 (4), 383-391. (2009).

[Sanders et al. 2012] Sanders, K., Boustedt, J., Eckerdal, A., McCartney, R., Moström, J.E., Thomas, L. and Zander, C. Threshold Concepts and Threshold Skills in Computing. *Proceedings of the 9th Annual International Conference on International Computing Education Research.* 23-30. (2009).

[Thomas et al. 2012] Thomas, L., Boustedt, J., Eckerdal, A., McCartney, R., Moström, J.E., Sanders, K., and Zander, C. A Broader Threshold: including skills as well as concepts in computing education. In *Threshold Concepts: from personal practice to communities of practice. Proceedings of the National Academy's Sixth Annual Conference and the Fourth Biennial Threshold Concepts Conference*, Trinity College, Dublin, Ireland, 27-29 June 2012, ISBN: 978-1-906642-58-7