Programming 2 - Exploring “student thinking”

Christo Potgieter
WINTEC
Tristram Street
Hamilton, New Zealand
christo.potgieter@wintec.ac.nz

Lesley Gousmett
WINTEC
Tristram Street
Hamilton, New Zealand
lesley.gousmett@wintec.ac.nz

This poster paper appeared at the 2nd annual conference of Computing and Information Technology Research and Education New Zealand (CITRENZ2011) incorporating the *24th Annual Conference of the National Advisory Committee on Computing Qualifications*, Rotorua, New Zealand, July 6-8. Samuel Mann and Michael Verhaart (Eds).

Abstract

Poster “Programming 1: Local pass rates improvement” shows improvements and possible reasons. This poster briefly “problem-solving thinking” considering well-known project BRACElet, to identify key principles for use in poster “Programming 3...” when practices are reviewed.

Keywords

Novice programmers, programming curricula, schema, patterns, comprehension, problem solving

“Thinking”, “problem solving” and “logic”

Winslow (1996) connected programming with Physiology while Weinberg (1998) then defined “the psychology of computer programming” with cognitive psychology. Also relevant might therefore be aspects from the fields of Problem-Solving Heuristics, Critical Reasoning, Philosophy Logic, Cognition & Learning, Educational Psychology and Patterns. This connection seems familiar in BRACElet project.

BRACElet project 2007-2010

Problem-solving skills for success in programming courses was studied in BRACElet project (Whalley, 2006). During the project, De Raadt (2007) summarized difficulties experienced by novices while learning programming and explored the explicit incorporation of problem-solving strategy instruction into teaching.

It’s a project of 12 tertiary institutions (Australasia and USA) investigating reading and comprehension skills of novice programmers, covering SOLO taxonomy (Biggs & Collis, 1982) using Bloom’s taxonomy:

Extended abstract	Articulation beyond coherent whole, placing topic area into broader context
Relational	Articulates relationships between facts, articulating a coherent whole
Multi-structural	Many connections among the facts without articulation of major

	connections that render a coherent 'big picture'
Uni-structural	Simple, obvious connections between facts' no articulation of significance of connections
Pre-structural	Isolated facts

Lister & Edwards (2010) shows how explanation of code by students are mapped onto SOLO when assessing their knowledge integration. It was noticed that many students can say what each line of coding does (Multi-structural) but not the program as whole (Relational) - students then also struggle to write coherent programs of their own.

De Raadt (2007) proposed computing education research should explore the relationship between comprehension-generation and knowledge-strategy dimensions". Actual work done includes empirical studies on a hierarchy of programming task "complexity" (Lister & Edwards (2010, pp10)) - independent and consecutive studies with different statistical analysis (both collected data on programming task "complexity" from student articulation and their actual code written). A hierarchy of dimensions was statistically confirmed. How to plan it in teaching?

De Raadt (2007) summarizes several initiatives to do more explicit building of *intellectual frameworks* for aspects of programming. This includes work by de Raadt, Toleman & Watson (2006-2008) who developed an approach to teach problem-solving strategies explicitly, specifically an experiment involving the solving of problems typical of those for a novice programmer at the end of a first-year introductory course in programming.

BRACElet Fellowship at the University of Technology, Sydney, ended in 2010, but the final report by Lister & Edwards (2010) states that connecting BRACElet to a wider set of literature was only begun and is likely to be ongoing, mentioning problems experienced for future consideration, such as Ethics (using student data).

Proposed principles/fundamanetals/essence

- Understanding SOLO hierarchy/ies in program code
- Assessing student "thinking" from their articulation
- Doing exercises, developing skills re problem solving
- Keeping it simple rather than comprehensive-complex
- Checks the complications re ethics for publication

Conclusion

Principles of SOLO, general problem solving and broader psychology could be used as reference, context or filter to review the practices of the tutor. However, to minimize preconception, this knowledge is not shared with the tutor who will simply be asked to describe her approach and practices. Her write-ups are then reviewed for links to aspects in above literature, which she will assess for truth. This is covered in the third poster of the series.

Acknowledgement:

Fernanda Potgieter shared theories and ideas regarding "Thinking, Problem Solving, Logic" and other aspects from cognitive psychology.

References

Biggs, J.B. & Collis, K.F. (1982). *Evaluating the quality of learning: The SOLO taxonomy* (Structure of the Observed Learning Outcome). New York: Academic Press.

de Raadt, M. (2007). A Review of Australasian Investigations into Problem-Solving and the Novice Programmer, *Computer Science Education*, 17(3), pp. 201-213.

Lister, R. & Edwards, J. (2010). Teaching Novice Computer Programmers: bringing the scholarly approach to Australia. A report on the BRACElet project, *Australian Learning & Teaching Council*.

Weinberg, Gerald M. (1998). *The Psychology of Computer Programming* (Silver ed.), New York: Dorset House

Whalley, J.L., Lister, R., Thompson, E., Clear, T., Robins, P., Kumar, P.K.A. & Prasad, C. (2006). An Australasian Study of Reading and Comprehension Skills in Novice Programmers, using the Bloom and SOLO Taxonomies. *Proceedings of the Eighth Australasian Computing Education Conference* (ACE2006), 52, 243-252.

Winslow, L.E. (1996). Programming Pedagogy – A Physiological Overview. *SIGCSE Bulletin*, 28 (3), 17 – 22.