

2011 Sydney Region Computing Education Conventicle

Thursday 24 November, 10am-4pm

School of IT Building, University of Sydney
1 Cleveland St, Sydney
<http://www.cs.usyd.edu.au/about/directions.shtml>

Arrival and morning tea, 10am (all times are 3 hours earlier for WA attendees)

Tony Koppi, Australian Council of Deans of ICT (10.30-11)

What students are telling us about why they left their ICT course

Student attrition is an issue of particular concern in the field of ICT because the industry faces staffing shortfalls. The study described in this paper provides further understanding of the causes of attrition from ICT courses by exploring the reasons students give for leaving their ICT courses. An online survey of early leavers from four Australian universities was conducted. The results show that many factors can contribute to the attrition of ICT students, and that for many students it is a combination of issues that leads to their withdrawal. Only a relatively small number of ex-students had experienced serious life events that necessitated their withdrawal. It was much more common for the participants to cite reasons associated with the university environment, the teaching of their ICT course, and their inability to combine their studies with other commitments. Recommendations are made to address issues that could be mitigated by university action.

Paper to appear in ITALICS – Innovation in Teaching and Learning in Information and Computer Sciences

Richard Gluga, University of Sydney (11-11.30)

Coming to terms with Bloom: an online tutorial for teachers of programming fundamentals

This paper describes a web-based interactive tutorial that enables computer science tutors and lecturers to practice applying the Bloom Taxonomy in classifying programming exam questions. The structure, design and content of the tutorial are described in detail. The results of an evaluation with ten participants highlight important problem areas in the application of Bloom to programming assessments. The key contributions are the content and design of this tutorial and the insights derived from its evaluation. These are important results in continued work on methods of measuring learning progression in programming fundamentals.

Paper to be presented at ACE 2012

Rachel Cardell-Oliver, University of Western Australia (11.30-12)

Automatic Analysis of Elementary Java Programming Patterns

Elementary programming patterns are short, generic program plans that can be used for teaching students how to solve recurring problems. For example, a guarded statement block, or a loop to apply an operation to every element in an array. Previous studies have shown that explicit instruction in elementary programming patterns benefits novice programmers. We build on those studies by implementing and evaluating a framework for the specification and automatic evaluation of programming patterns in student Java code. We use the Checkstyle API to implement rules for selected programming patterns. The new rules are evaluated for robustness, correctness, usefulness and configurability by applying the rules on over 200 student programming assignments. Our analysis identified recurring problems in the way students apply expected patterns. Our tools and results can be used to improve teaching and learning by providing a new type of feedback to lecturers and to students and for further research into student programming practice.

Work in progress

Lunch, 12-1

Mark Guzdial & Barbara Ericson, Georgia Institute of Technology (1-2.30)

Why and how to teach computing to everybody

The people who invented the field of "Computer Science" explicitly envisioned everyone in academy learning to express themselves with computing -- explicitly, everyone learning programming. How would we get to that? In Georgia, we're working to build interest in computing in the early teens, improving CS education in secondary schools, and developing new kinds of University courses that meet the needs of non-CS majors, including liberal arts and architecture students.

Increasing enrolment and retention

Computing is a key to innovation in many disciplines, yet interest and enrolment in computing specializations has been declining -- it certainly isn't keeping pace with the demand for computing expertise in the economy. A particular problem is that the students we are drawing into computing tend not to be diverse. In Georgia in the United States, we have had success in driving enrolment up (with greater diversity), and in creating curricular changes that support greater retention. We describe our efforts and provide some research results showing the impact of the approaches.

Keynote

Afternoon tea, 2.30-3

Donna Teague, Queensland University of Technology (3-3.30)

Neo-Piagetian Reasoning and the Novice Programmer

Lister has suggested that novice programmers pass through neo-Piagetian stages: sensorimotor, preoperational, and concrete operational stages, before eventually reaching programming competence at the formal operational stage. My PhD project is to empirically test Lister's proposal. In this talk I will be presenting data from a 'think out loud' session with a student who appears to manifest sensorimotor or preoperational behaviours.

PhD work in progress

Simon, University of Newcastle (3.30-4)

Assignment and sequence: why some students can't recognise a simple swap

In the current cycle of the continuing drive to discover why some students have such difficulty learning to program, the goalposts have been moving ever closer to the very fundamentals. On the basis of some very simple multiple-choice questions asked in tests in introductory and subsequent programming courses, it is proposed that many students have not grasped the concept of sequence in programming, the concept that a group of statements in a procedural programming language will be executed in the order in which they appear. This problem was discussed in the programming education literature as long as 25 years ago, but it seems that little has changed in that time. The consequences of the preliminary finding are discussed, and further work is proposed to confirm it or otherwise.

Paper presented at Koli Calling 2011