A shared, applied epistemology of competency in computer programming

Raymond Lister

University of Technology Sydney

The project

- ALTC-funded project
- $155,000, 2 years, 2011-2013
- UTS, Newcastle, Monash, RMIT, QUT, Sydney
- Exploring how we assess competency in computer programming
  - what we assess
  - how we assess
  - how consistently we assess
  - how validly we assess

Sub-project: syllabus specification

- What's covered in a course/program; what can we expect students to know after completing it?
- Good students, but also bare-passing students
- Good if it can be expressed in terms of a specified curriculum
- Clearly depends on the assessment items
- Software product compares assessment items & student marks against a specified curriculum (which can be varied) and against various known measures such as Bloom's taxonomy

Sub-project: question benchmarking

- Many students struggle with programming
- Many academics fear that it’s just their students
- So let’s ask a common set of questions in exams at multiple institutions, and compare the results
- Also think-aloud studies, which are invaluable in seeing how students solve problems

Sub-project: exam classification

- Examining the exam questions from many institutions, at many levels of the degree
- What do they assess?
- How difficult are they?
- How complex are they?
  - linguistic complexity
  - conceptual complexity
  - external domain references
  - code length
  - explicitness
  - intellectual complexity