Innovative Teaching Practices in Computing Education: The TLA Project

Jan Holden and Alison Young, UNITEC, New Zealand {jholden,ayoung}@unitec.ac.nz

In 1998 UNITEC Institute of Technology in Auckland, New Zealand, carried out a scheduled Quality Management System (QMS) monitoring process, which was a self-evaluation of course delivery and assessment [1] This internal audit was validated in 1999 by an external audit by two independent people (Grace Sylvester, MIT, Auckland and Andrew Gonczi, UTS, Sydney) [2]

A key recommendation arising out of these reports was that there should be institution-wide academic staff development initiatives that give priority to improving:

- the quality of learning outcomes
- teaching, so that the focus shifts from content to learner
- assessment practice

As a result the School of Information Systems and Computing appointed a faculty member as the Teaching, Learning and Assessment (TLA) coordinator to address this recommendation in relation to the undergraduate computing courses.

Strategies being developed within the teaching area include a model of reflective practice for faculty, identification of competencies of a "good teacher", and the sharing of innovative methods and best practice. An emphasis on targeting appropriate professional development, linked with performance appraisal is also included

Students will play a part in the development of the learning strategies and their input into what helps them learn will be incorporated into the implementation of the strategies.

One of the main areas of innovation will be the development of new assessment practices. In our evolving learning environment that is categorized by such innovations as the trend towards electronic delivery, methods of assessment that have been used for many years may no longer be appropriate. Innovative new methods will be sought and implemented to not only ensure rigorous scrutiny but also encourage student learning and development.

This poster presentation is a "work in progress" which will outline the goals for 2001, the strategies that have been developed. and the progress to date in implementing those strategies.

[1] UNITEC Institute of Technology, *Audit of QMS Element 9: Programme Delivery and Assessment. Self Evaluation Report*, UNITEC: Auckland, 1998

[2] Gonczi, A and Sylvester, G. *Audit of Programme Delivery and Assessment at UNITEC.* UNITEC: Auckland, 1999

Motivation = Value x Expectancy

Tony Jenkins, University of Leeds, UK, tony@comp.leeds.ac.uk

Instructors are often tempted to blame a lack of activity (or learning) on the part of their students on a lack of motivation. Students are no longer motivated to learn and, if they are motivated at all, their only aim is an eventual highly paid career. They are prepared to expend only the minimum effort necessary to achieve this.

This is not so. Motivation is a complicated concept - it is hard to quantify in any meaningful sense, and what motivates one individual may demotivate another. It is possible to observe a student, or a class, and to infer their likely type and level of motivation, but it is not possible to be certain.

One view of motivation sees it as a function of two factors, value and expectancy [3]:

motivation = value x expectancy

These two factors are said to multiply, rather than add, since there will be no motivation if either factor falls to zero. A student must value the learning to be gained, and must expect success in assessment.

Much existing work on presenting basic computing concepts to "unmotivated" classes focuses on motivating students through arousing their interest (for example, [1], [2], [4], [5]) thus addressing the "value" part of the formula. Little has been done to consider how instructors should ensure that their students will expect success.

This poster will present some of the influences at work on a student's motivation, and will consider which an instructor can reasonably expect to influence.

[1] Owen Astrachan, *Hooks and Props in Teaching Programming*, Proceedings of ITiCSE '98, PP 21-24.

[2] Harriet J. Fell and Viera K. Proulx, *Exploring Martian Planetary Images: C++ Exercises for CS1*, Proceedings of SIGCSE '97, PP 30-34.

[3] John M. Keller, *Motivational Design of Instruction*, In Charles M. Reigeluth (ed.), Instructional-Design Theories and Models: An Overview of their Current Status, PP 383-464, Lawrence Erlbaum Associates, 1983.

[4] Robert Moser, *A fantasy adventure game as a learning environment: Why learning to program is so difficult and what can be done about it* Proceedings of ITiCSE '97, PP 114-116.

[5] Eric V. Siegel, *"Why Do Fools Fall Into Infinite Loops: Singing To Your Computer Science Class"*, Proceedings of ITiCSE '99, PP 167-170.