Effective Learning and Teaching in Computing

Alistair Irons & Sylvia Alexander (eds)
RoutledgeFalmer, 2004, 224pp. ISBN-13: 978-0415335010 (pbk)

This edited book is a collection of short chapters from authors that are, in the main, well known to those who are engaged in research in the subject area. One of a series of books entitled *Effective Learning and Teaching in Higher Education*, this title tries to add a computing flavour to the issues surrounding teaching and learning in higher education.

The result is a book that is both narrow and broad in its approach. Some of the chapters focus on a single aspect of the computing curriculum, notably programming, and discuss how specific pedagogical approaches can be used, whereas others paint a landscape of current practice with small attention to detail. To a newcomer to the field, some of these more general chapters may take some digesting, but they certainly provide a foothold that is often missing from books of this ilk.

The book begins with an editorial and introduction where the principal authors are introduced and the issues surrounding teaching and learning in computing are highlighted. These are flagged as i) the issues regarding entry requirements and the difficulties of teaching students with many different skill levels in computing, ii) student workload, especially as it relates to the acquisition of skills alongside the acquisition of academic knowledge, iii) the teaching of transferable skills, especially as constrained by modular degrees, iv) the challenges of motivation of staff and students in an environment of technological and pedagogical change, and v) issues relating to the trend towards widening participation.

The remainder of the book is presented in four parts. Chapters 1 to 6 make up Part 1, which is entitled *Teaching and the support of learning* and includes work on motivation, taxonomies and accessibility. In this part, the most useful chapter for me was the one on motivation. It was written in a very practical way and included some useful tips for keeping students motivated. The work on taxonomies was useful in that it provides a theoretical view of the field that would be useful for researchers and instructors who are new to the area. Other sections

in this part were concerned with accessibility and the design of learning environments and rather suffered from being too general (i.e. not computing specific) and from duplication, an inherent problem in edited books.

Part 2 had the curious title of *Learning activities for computing students* and without turning the page it was difficult to guess what might come next. What was presented was rather a let down as it was in fact a series of chapters on assessment methods, which are generally not seen as learning activities! Chapter 7 had a slight learning feel to it, presenting the case for group work and offering a range of good reasons for its inclusion. It was a disappointment that the chapter did not balance this with a discussion of the negative aspects of group work; the authors propose that group work should be 'fun and relevant to industry' whereas 'fair and manageable', which is the cry from students and lecturers, was not really explored. Chapters 8, 9 and 10 are all about assessment. There is a thorough and useful account of Computer Aided Assessment (CAA) that focuses on its use in programming courses. This chapter provides some sensible advice as well as providing a review of some of the commonly found CAA packages. The following chapter fits nicely with this by providing a reflective commentary on how electronic assessment impacts on student motivation. There is some duplication across Chapters 8 and 9, especially as both chapters consider programming. Chapter 9 concludes with some discussion of group assessment, peer assessment and self assessment, which at first glance appears to not fit in well with the rest of the chapter but is worth looking at, if only to motivate the reader to look at Chapter 10 which is an examination of plagiarism, both with code and with text. This chapter is well written (but arguably not really in an appropriate section as is it really about learning activities?) and it provides a helpful overview of the methods that can be used to prevent, reduce and detect plagiarism.

Part 3 of the book, *Developing effective learning environments*, describes a study that looks at the effectiveness of distance learning in computing, includes a discussion of ways that academia and industry might collaborate in computing, and considers entrepreneurship and employment in the sector. I found this part of the book especially confusing as the chapters all seemed very different, in layout, depth and structure.

The last section of the book focused on reflective practice and was, for that reason, possibly the most valuable part of the book, especially as computer science educators are not known for their reflective tendencies. Chapter 15 describes a range of methods that can be

used by academics to become more reflective, and Chapter 16 looks at the delivery and assessment of quality in the profession. The last two chapters set an agenda for action. These include the need to deal with a varied student base, the requirement for the profession to link with industry and provide students with skills that can be used in the marketplace, the impact of the Bologna agreement on the curriculum, the changes wrought on the curriculum by new technology and the impact of new technology on teaching and learning. Interestingly, most of these issues are not particularly well covered in the book.

To conclude, the book is well written and the authors write with authority and enthusiasm. The book does suffer from poor organisation and some rather misleading labelling, and the examination of learning is cursory. From the point of view of teaching and assessment, the book offers some useful information that is backed up with published experimental work and reflective contemplation. As a reference to the published work in computer science education, the book is very useful, providing a rich bibliography for future pedagogical research in this subject area. This is a book that it would be good to see in the bookcases of computing academics.

Janet C Read

University of Central Lancashire, UK.