The Transition from School to University: Would Prior Study of Computing Help?

Martyn Clark and Roger Boyle

School of Computing, University of Leeds, UK
{martyn, roger}@comp.leeds.ac.uk

Abstract. We investigate issues in the preparation of students for undergraduate study. Specifically, we focus upon the question of whether computer science students would be better prepared if they were required to pass a school level qualification in the discipline. Thus we investigate the school level curriculum in detail and make a comparison with the demands of a typical UK university first year. We conclude that there is no reason necessarily to see a school level qualification as assisting the preparation of students for undergraduate study in computer science. Rather, we hypothesise that the value of the qualification will depend heavily on the nature of the teaching experienced.

1 Introduction

The preparation of students for university entry is known to be very influential on their success [1]. Further, the preparation of students for university study of computer science has recently been highlighted as a key issue [2, 3, 4]. Thus we address the issue of school level computer science as preparation for studying the discipline at university.

Specifically, we compare the English school level computer science curriculum with the requirements of a typical university first year and ask what grounds exist for believing that an applicant with a school level computing qualification will perform better in a university computing course than another applicant who has not studied the discipline before.

The paper is structured as follows. Section 2 provides background relating to university admissions and pre-university qualifications in England together with the methodology for the study. Section 3 details findings and these are discussed in section 4. The paper ends with conclusions and an outline of plans for further work.

2 Background

In England, compulsory education ends at age sixteen but most individuals do not move on to university until the age of eighteen. While a variety of alternative routes exist, most university candidates spend the intervening years at tertiary

R.T. Mittermeir (Ed.): ISSEP 2005, LNCS 3422, pp. 37–45, 2005.

institutions studying in preparation for A-level examinations. These examinations are offered in a wide range of subjects and most candidates will attempt three or, perhaps, four subjects. In the majority of cases results will play a major part in determining which universities and which degree programmes will accept the candidate.

Indeed, to be accepted at a particular English university candidates must satisfy up to three types of admission requirement.

- Matriculation requirements are designed to allow candidates to demonstrate that they have the potential to benefit from the university experience. For example, in addition to passing at least two A-levels at grade E or above, many universities require candidates to do well in at least five subjects in pre-A-level examinations taken at age 16.
- Popular and prestigious institutions and courses will normally set entrance requirements somewhat above this minimum level. For example, to study history at a member of the UK 'Russell Group' of leading research universities candidates must pass three A-levels at grades BBB [5].
- In addition, entry to many degree programmes is conditional upon passing an A-level in the student's discipline of choice. Thus applicants to physics should offer an A-level in the discipline at grade A or B.

We are aware of no degree course in computer science which requires students to have passed an A-level in the discipline. However, at our institution approximately half of each cohort present with this qualification [6, 7] and we are interested in whether these students have an advantage over their peers who are computing neophytes. An academic truism that entry qualifications are at best poor indicators of student achievement on graduation is supported by a growing body of research including our own study of computing [8, 9, 10, 11, 12]. However, these are complex issues and at least one large-scale quantitative study appears to contradict this finding [13].

Nevertheless, we are aware of no research which attempts to move back a step in the analysis and ask why A-levels, or any other tertiary exit qualification, *should* be a good predictor of university performance. Thus we ask: what grounds exist for believing that undergraduates with a school level computing qualification should get better marks than their peers?

A-level examinations are run by examination boards. *Inter alia*, examination boards publish specifications setting out the precise nature of the qualification, including the syllabus to be examined. They also set and organise examinations, including the marking and publishing results. A number of boards exist and although their respective A-level computing specifications are not identical, they are, of necessity, comparable. Indeed, the supervisory regime which assures the quality of A-level qualifications ensures that significant elements are common to each board's specification in a given subject. We examined the current A-level computing specifications offered by two of these examinations boards: the Assessment and Qualifications Alliance (AQA) and the Oxford, Cambridge and RSA Examinations Board (OCR). The selection of these boards largely reflects the accessibility of data but by studying two boards' specifications we were able to consider the relevance of their different emphases to our findings.

We compared these specifications with the first year undergraduate curriculum in order to consider the extent to which experience of an A-level course might prepare students for the university experience. The approach is subjective and relies on our judgements as teachers of computing. However, we argue that this process mirrors that undertaken in tertiary establishments when teachers receive an A-level specification and plan activities to prepare students to sit examinations based upon it. Further, the analysis could be applied to other school level qualifications and to the entry level curriculum of other universities both in England and further afield.

3 Findings

Our findings, which necessarily rely on our interpretations of both university and A-level curriculum documents, are designed to identify those aspects of the level one curriculum in which students who have passed A-level Computing might be thought to have an advantage over their peers who have not.

The level one undergraduate curriculum under consideration is specified as twelve modules addressing ten sub-disciplinary areas; there are two modules each of programming and mathematics. For each module we report on the extent to which we perceive the A-level as preparation. Modules are grouped into: those we judge to be new even to students who have the A-level (*new challenges*); those we consider most likely to reveal an advantage for these students (*familiar territory*); and those where we feel the A-level may represent a good foundation, although this may not be revealed in results (*firm foundations*).

3.1 New Challenges

Two areas of the undergraduate level one curriculum emerge as being highly distinct from the A-level experience: mathematics and artificial intelligence. There is evidence that some mathematics did feature strongly in earlier versions of the A-level curriculum [14], while artificial intelligence is a particular specialism of the department under consideration. In our judgement A-level Computing should not be a discriminator in these modules.

– English students are not alone in finding that the importance of mathematics to the practise of computing often comes as both a surprise and a disappointment [15]. However, a variety of mathematical concepts and techniques are fundamental to computing and undergraduates are expected to become familiar with areas such as algebra, discrete mathematics, logic, geometry, probability and statistics.

  In contrast, the A-level syllabus contains almost no mathematics. Each Board's specification includes a requirement to study number systems but this is the only mathematical content we identify in the specification. It should be noted, however, that an A-level in mathematics exists and that a number of computing degree programmes make a pass in this subject a prerequisite.

– The artificial intelligence module is designed to help students to develop understanding of the fundamental ideas, issues and techniques of artificial intelligence. Its key syllabus content, knowledge representation and reasoning, leads to a survey of the main sub-areas including computer vision; computing using natural language; computer learning.

  While equivalents for the remainder of the level one modules are almost certain to be found in any university level one computing curriculum, artificial intelligence is a particular specialism of the department concerned. It is, therefore, not surprising that we found no evidence that these topics are included in the A-level curriculum.

3.2 Familiar Territory

In contrast, there may be grounds for believing that undergraduates who have taken the A-level Computing will get better marks than their peers who did not in the database, professional development, and architecture modules. The close relationship between the curriculum requirements of these modules and the A-level specifications suggests that while the translation from the school regime to that of the university may require some effort, students should be on familiar territory.

– Introductory databases focuses upon databases as programmable systems. The curriculum is based largely around data modelling techniques and the use of the Structured Query Language (SQL) to develop, maintain and manipulate relational databases.

  Databases are a major topic in the A-level curriculum which refers specifically to the SQL, relational models and a number of other concepts which feature in the level one curriculum. However, the extent to which this is made specific differs between the Boards considered: while the AQA specification clearly addresses many of the issues included in the level one curriculum, the OCR specification addresses them, but less obviously.

  Of course, expectations at undergraduate level will not be identical with those at A-level, but our judgement is that students who have considered the concept of the relational database and used the SQL should have skills and knowledge on arrival at university that other students must learn during their level one studies. Refining understanding as concepts and techniques are encountered in new contexts should be less challenging than starting from scratch.

– The inclusion of professional development as a topic in the undergraduate curriculum reflects the close relationship between the theory and practise of computing. Students are encouraged to become aware of managerial, social and legal issues arising from the practise of computing and are given opportunities to develop and refine various generic skills such as report writing, working in groups and communication skills. Professional organisations,

such as the British Computer Society [16], offer additional accreditation to degree programmes and in the UK this form of accreditation is widely sought by university departments as a form of endorsement of quality and comparability. These organisations see professional development as a particularly important aspect of the undergraduate curriculum.
These issues feature strongly in the A-level curriculum. Each of the specifications considered makes specific reference to legal, social, historical and ethical issues in computing. Therefore, we judge that students who have taken the A-level may be aware already of many of the issues of professionalism that this module seeks to highlight.
– The syllabus for the architecture module refers to computer internals, performance measures, arithmetic and logic operations and CPU internals. The A-level specification refers to many of these concepts specifically, including the expectation that students develop an historical perspective. This module, therefore, has the potential to highlight any advantage for students who have passed the A-level.

3.3 Firm Foundations?

The potential for undergraduates holding the A-level qualification to be advantaged in the remaining modules is less clear. Certain aspects of the undergraduate curriculum are specified for study at A-level, but there is clearly scope to address them in more or less detail depending on the Board chosen, the facilities available, and the interests of teachers and students.

In these modules the A-level may provide a useful foundation but we judge that, as activities at undergraduate level will tend to focus on the more advanced material, module results may not reveal much advantage for A-level holders.

– Introductory networking introduces ideas about operating systems and computer-to-computer communication. Key concepts from this area are required at A-level, for example, client-server models of communication, hyperlinking and the World Wide Web, and common network environments. Clearly, however, the undergraduate curriculum goes further, for example, it includes CGI scripts, data compression, and distributed applications.
– Similarly, the basics of human-computer interaction are specified as topics for A-level study including, for example, user models and interface design. The undergraduate curriculum builds upon the basics by discussing why interaction is as important as processing and analysing ‘human factors’, such as vision and memory, which are important in the design of interactions.
– Analysis of algorithms is a theoretical, rather than practical, module which, in addition to introducing data structures, invites students to consider particular problems and how they might be tackled using a computer. While the A-level specification does call for a familiarity with simple data structures and some description of algorithms we judge that the undergraduate curriculum requires a significantly more sophisticated understanding.
– The information systems module introduces a more ‘business-oriented’ or less technical (‘softer’) perspective on computing. It is concerned with concepts

relating to systems and information and requires students to consider different perspectives on the process of developing information systems. These concepts do not feature prominently in the A-level curriculum. However, courses based upon these specifications should consider the relationship between data and information which is an important foundation for undergraduate study. Similarly, the uses of computers in organisations, particularly management information systems, is part of the A-level specification.

3.4 The Special Case of Programming

Software engineering is a major component of any undergraduate programme in computing. The ability to program is one of the defining characteristics of members of the computing community [15]. Experience suggests that learning to program dominates their first year at university for a large number of computing students. Further, students who do not pass these modules are unlikely to enter level two.

The level one programming modules focus on the syntax of a programming language and techniques for ensuring that robust and reliable programs are developed which meet requirements: that is, techniques for engineering software. In contrast, the A-level specification does not require programming specifically. Rather, the emphasis is on systems development: an expression which could mean programming, but equally could refer to the development of systems using applications such as database management systems (e.g., Microsoft Access). Our judgement is that students arriving at university able to program would be at a substantial advantage but there is no guarantee that the A-level would deliver this.

Aspects of software engineering are required by the A-level specifications: particularly, aspects of design and testing. However, we consider these topics alone insufficient to confer a significant advantage because in comparison with programming they are not significant topics in the software engineering modules.

4 Discussion

We have identified certain university modules as being highly convergent with the school level curriculum, but this is balanced by other modules where no relationship is apparent. This finding is consistent with the results when we compared the first year grades attained by students who were admitted with A-level Computing and those attained by students with no prior experience of studying the discipline; students with the A-level were shown to do better in databases, professional development and computer architecture but not in other modules [6]. Is this sufficient to make prior study of the discipline a requirement for university entrance?

The reasons for a university to view school level experience as a prerequisite for the study of a discipline are rarely made explicit. We suggest there are two main perspectives;

- Necessary preparation: the learning of material essential to even the most elementary study at university. An example might be an understanding of calculus to study mathematics – without it, the student would find it very difficult to participate in disciplinary conversations. Thus an accepted starting point for university study is defined.
- Gate-keeping: supplementing basic entry entry requirements in a largely arbitrary way for courses where applications exceed significantly the number of places available.

Whilst recognising that if the number of places is oversubscribed some means must be found for selecting candidates for admission, we see little to commend the latter approach. Similarly, we do not see in these findings a case for adopting A-level Computing as a prerequisite for university study on the basis of necessary preparation. Not only does the A-level appear to confer a significant advantage in only three modules but we suggest that crucial differences in the way computing is experienced at the school level may have significant implications for the relevance of this qualification as preparation for joining a university department.

In the UK the origins of a discipline of computing lie in the creation of computing facilities designed to service research in mathematics and the science and engineering disciplines [7]. Thus, ideas about a *higher* education in the discipline are founded not only upon relatively easy access to equipment of a certain standard but upon the methods and techniques devised in universities for the using computers in the solution of numerical problems. In contrast, while school level computing appeared in the late 1960s, the subject did not become widespread until the general adoption of the personal computer eliminated the issue of access to equipment [14].

Prior to the commercial development of microprocessors and home computers, school computing tended to rely on a postal service to a mainframe (commonly at a university or local government facility). While slow and cumbersome, such access at least guaranteed a quality of service; pupils were operating in the same computing environment as academics and professionals (albeit remotely). We hypothesise that the introduction to schools of personal computers changed this.

Of course, personal computers are used in undergraduate computing courses and many graduates of computing degree programmes go on to support their use in business and other environments. Significantly, however, much professional and academic work in such matters as programming, database management, the Internet, etc. is undertaken using equipment that would not normally be found in the average office or home. If this hypothesis is correct, one reason that A-level Computing is of limited relevance as preparation for university study is that pupils do not operate in computing environments typical of much professional and academic work.

A second hypothesis which may have relevance to the utility of A-level Computing as preparation for university study in the discipline relates to staffing. Universities began to offer post-graduate courses in the discipline during the late 1950s with undergraduate courses being introduced in the 1960s. The number of places available has grown dramatically, particularly during the 1990s, but we hypothesise that demand from industry for the graduates of these courses, and the consequent pay differential, has meant relatively small numbers of computing graduates have entered the teaching profession.

If this hypothesis were correct it might help to explain why the A-level Computing specification does not require that candidates learn computer programming. Access to equipment is unlikely to be an issue with respect to programming which can be learnt using personal computers and commonly available software. Rather, the absence of programming from many A-level courses may reflect the fact that many A-level teachers are not computing specialists. That is, although computing academics perceive significant differences between their discipline and skills associated with the *use* of computers, the implication of this hypothesis is that this distinction is less clear at the school level; computing is taught from the perspective of using rather than building computer systems.

This is not intended to be derogatory; the ability to use information technology efficiently and effectively has become an important life skill. However, we argue that the ability to manipulate applications such as Microsoft Access or Macromedia Dreamweaver via user-oriented graphical interfaces is a relatively poor preparation for studying a discipline concerned with what happens behind the interface. Certainly, the ability to program would be significantly more beneficial than the most highly developed IT skills in the study of computing.

5 Conclusions

We have argued that there is some convergence between the school level curriculum in England and a typical first year university computer science curriculum. Students entering university with prior experience of the discipline are likely to have encountered already a number of ideas that computing neophytes will meet for the first time. However, unless students learnt to program as part of their A-level studies this advantage is unlikely to be significant. Thus we do not advocate making prior study of the discipline a condition of entry to university computer science degrees.

Our findings highlight the importance to university computer scientists of awareness of the treatment of the discipline in the school curriculum. In particular, support for the teaching of programming would appear a priority. Further our findings highlight the need to support students who have not studied the discipline before who may become demotivated if they struggle relative to students who have seen some of the subject matter before.

Finally, in this research we have relied upon our interpretations of A-level specification documents. Whilst we argue that our interpretations are sufficiently relevant for the findings presented here to be meaningful, it would be interesting to study students' experiences of the A-level courses based upon these documents. That is, further work will seek to compare our understanding of the requirements with the reality of studying computer science at school level.

Acknowledgements

Some of this work was conducted with grant assistance from the Learning and Teaching Support Network for Information and Computer Sciences, University of Ulster, which we gratefully acknowledge. We are also most grateful for the significant assistance provided by the archive sections of AQA (Hilary Nicholls), AEB (Jane Bradshaw) and UCLES (Gillian Cooke) and to Miriam Zukas for comments on an earlier draft of this paper.

References

1. Ozga, J., Sukhnandan, L.: Undergraduate non-completion: Developing an explanatory model. Higher Education Quarterly **52** (1998) 316–333
2. Clark, M.: Teaching computing in the liquid modern world. Presented at Grand Challenges in Computing, University of Newcastle, 29-31 March (2004)
3. Utting, I.: Mind the gap! Presented at Grand Challenges in Computing, University of Newcastle, 29-31 March (2004)
4. Johnson, C.: What do students want ... Presented at Grand Challenges in Computing, University of Newcastle, 29-31 March (2004)
5. University of Leeds: Coursefinder 2003 (2002) Available at: `http://tldynamic.leeds.ac.uk/ugcoursefinder/2003` [accessed 3.6.04].
6. Boyle, R., Clark, M.: A-level computing: its content and value. University of Leeds School of Computing Report No. 2002.15 (2002)
7. Clark, M.: Constructing the discipline of computing: implications for the curriculum. unpublished PhD thesis, University of Kent, UK (2004)
8. Lizzio, A., Wilson, K., Simons, R.: University students' perception of the learning environment and academic outcomes: implications for theory and practice. Studies in Higher Education **27** (2002) 27–52
9. Osborne, M., Leopold, J., Ferrie, A.: Does access work? The relative performance of access students at a Scottish university. Higher Education **33** (1997) 155–176
10. Brown, M., Macrae, S., Rodd, M.: Mathematics undergraduates' attitudes to their course (2004) In preparation.
11. Child, D.: A-levels as a predictor: the correlation between A-level grades and degree results for the 1981 entry to Leeds. The Reporter: The University of Leeds newsletter (1986) 4 Number 255.
12. Boyle, R.D., Carter, J.E., Clark, M.A.C.: What makes them succeed? Entry, progression and graduation in Computer Science. Journal of Further and Higher Education **26** (2002) 3–18
13. Bekhradnia, B.: Who does best at university? (2002) Available at: `http://www.hefce.ac.uk/learning/whodoes` [accessed 3.6.04].
14. Boyle, R.D., Clark, M.A.C.: A-level computing: its history and development. IEEE Annals of the History of Computing (2004) Forthcoming.
15. Alexander, S., Amillo, J., Boyle, R.D., Clark, M.A.C., Daniels, M., Laxer, C., Loose, K., Shinners-Kennedy, D.: Case studies in admissions to and early performance in computer science degrees. SIGCSE Bulletin **35** (2003) 137–147
16. British Computer Society: Exemption & accreditation for higher education institutions. Available at: `http://www.bcs.org/BCS/Products/HEAccreditation` [accessed 7.6.04] (2003)