The Computer Journal Vol. 48 No. 1 © The British Computer Society 2005; all rights reserved
DOI:10.1093/comjnl/bxh064

Grand Challenges
Grand Challenges in Computing: Education—A Summary

ANDREW MCGETTRICK[1], ROGER BOYLE[2], ROLAND IBBETT[3], JOHN LLOYD[4], GILLIAN LOVEGROVE[5] AND KEITH MANDER[6]

[1]*Department of Computer and Information Sciences, University of Strathclyde*
[2]*School of Computing, University of Leeds*
[3]*School of Informatics, University of Edinburgh*
[4]*School of Computing Science, University of Newcastle*
[5]*Previous affiliation: University of Northumbria*
[6]*Department of Computer Science, University of Kent*
Email: andrew@cis.strath.ac.uk

The conference on grand challenges, held in Newcastle on 30 and 31 March 2004, occurred at a particularly opportune time. The strand on the educational aspects was particularly relevant and the idea innovative in the sense that this was the first occasion on which a grand challenge event with a focus on educational issues in computing had taken place. This paper provides some of the background and includes a distillation of the educational challenges that emerged from that event.

1. INTRODUCTORY COMMENTS

Until relatively recently, computing as a discipline and as an area of academic study has been a phenomenal success story. Since its humble beginnings almost 60 years ago, its importance and popularity have increased so much that it has become one of the largest areas of study in many higher education institutions. Many aspects of society, business, the economy of countries and the competitiveness of industry have come to rely on computers and computing.

However, over the last three or four years there has been a change in fortune. In some sense, the discipline has come of 'an awkward age'. For those who cherish the discipline this is a time to come to the rescue; it is time to question aspects of the existing situation, to ask what to do and to seek remedies. Almost inevitably, this involves focussing on existing problems and difficulties, but with the very positive intention of removing these and improving the situation. It is important to address these matters.

1.1. Grand challenges conference

There is very little doubt that this conference on grand challenges, held in Newcastle on 30 and 31 March 2004, occurred at a particularly opportune time. This was not a conventional conference. This conference was about identifying problems and not solutions. The strand on the educational aspects was particularly relevant and the idea innovative in the sense that this was the first time that such a grand challenge event with a focus on educational issues in computing had taken place. It is important to comment that, though very naturally UK academics dominated the attendees, there was an important, very welcome and very influential international presence.

A careful selection of challenges serves an important role for a community. The identification process can draw a community together; it can serve to heighten ambition and set the agenda; it can provide a focus for ongoing activity and for cooperation; and it can inform the community itself and others about developments in the area. An important consideration is choosing challenges that are appropriate; part of this is being able to communicate these easily to a sceptical but informed public and ensuring that it has an appreciation of the issues and the benefits that will stem from the resolution of the challenges. It should also be easy to determine when one accomplishes (in whole or in part) a challenge. From a different perspective, choosing challenges is easy; choosing the right challenges is difficult.

An important aspect of the grand challenges would be the existence of mechanisms for ensuring that the community would be able to take up the challenges and respond. Part of this would be the availability of suitable outlets for publications from those working on the challenges. Part of it also would be resources to support relevant activity and to monitor developments. Nevertheless, the major considerations would be the necessary coordination and commitment.

1.2. Nature of a grand challenge

The nature and characteristics of a 'grand challenge' require articulation. The following ideas derive from characteristics of purely research-based grand challenges. However, the nature of education-based challenges must also accommodate issues such as changes in perception, social change and changing attitudes. The effort involved in bringing about such changes on a grand scale must not be underestimated.

For the purposes of this document, an education-based grand challenge must be such that, when achieved, it will:

- lead to substantial improvement in some significant aspect of the educational processes in computing
- arouse curiosity and generate enthusiasm within the computing community
- be international in scope and so have wide and significant relevance
- be comprehensible and capture the imagination
- have the capacity to bring about changes in attitude, changes in expectation and even change at the social level.

For many challenges, there may be additional expectations such as requiring collaboration involving a large number of individuals, requiring planning, requiring a significant amount of time. However, achieving them will not be easy; the hope and expectation is that individuals and individual groups will work at making inroads to address the challenges (or aspects of the challenges) with a view to reaching the intended goal.

This paper summarizes the results of this activity. For a complete report, see [1].

1.3. Contributions

The approach adopted by the organizing committee was as follows. A call for papers and ideas was issued. The submissions were peer reviewed and the committee selected about 25 submissions for presentation at the conference. Of the submissions, those from Peter Denning formed part of his invited address. The following submitted papers that were accepted for presentation at the conference: Peter Denning, Tom Boyle, Janet Carter, Lillian Cassel, Martyn Clark, Finton Culwin, Sharon Curtis, David Lightfoot, Maria Fasli, Neil Gordon, Gary Griffiths, Mike Lockyer, Richard Hebblewhite, Stuart Cunningham, Vic Grout, Mike Holcombe, M. Huggard, Adrian Jackson, Colin Johnson, Greg Michaelson, Margaret Ross, John Sargeant, Mary McGee Wood, Aaron Sloman, Mike Stannett and Ian Utting.

2. CHALLENGE AREAS

2.1. Perception of computing

Public image and public understanding of any discipline are important. These greatly influence parents, teachers, careers advisors and those who guide young people in their choice of more advanced study. Of course, they also influence young people themselves since they serve to make them curious about more advanced study and to consider a career in computing.

The achievements of computing over its short lifetime are under-reported. Mistakes and disasters rather than the achievements of the Internet or the promise of nano-computing appear in the press. There are many large and complex systems which function successfully and underpin many aspects of administration, the economy, societal matters and business; the Internet is used daily by many people and its availability is taken for granted. The infectious excitement of computing often manifests itself in applications and remarkable advances associated with other disciplines. For example, space flight as we know it could not occur without major support from computers and computing. Likewise, the advances associated with the genome project and with certain theorems in mathematics also fall into this category.

What kind of image of computing should one portray? It is certainly desirable to attract the best and responsible students into the discipline and to do so in significant numbers. The desirable image is one of an exciting, even vibrant, discipline where there are rapid and exciting developments that can help not only industry but also health care and the caring professions as well as education. Computing offers great opportunities for innovation, challenge and wealth creation.

2.1.1. Challenge

In the light of the discussion described for the situation above, the computing challenge for this area is as follows:

> Promote an improved and ultimately very positive public image of computing—ensuring that the public gains respect for the field and the professionals who practice within it.

2.1.2. Sub-challenges

Within the grand challenge for this section, it is possible to identify a number of interim challenges or sub-challenges that can inform and testify to progress towards meeting the main challenge.

(i) Promote a positive image of computing both by expounding the real achievements of computing and by articulating the aspirations and ambitions of research-based grand challenges that highlight a vision for development and innovation and create anticipation and excitement.

(ii) Illuminate the links with other disciplines, highlighting the advantages that such links provide and pointing to the future benefits of productive alliances with them; use these also, where appropriate, both to clarify the nature and the extent of computing as a discipline and to further the development of computing as that discipline.

(iii) Develop metrics that provide a barometer of the health of computing as a discipline, and then use these metrics to measure and monitor the progress towards improvement and to guide future developments.

(iv) Participate in research-based challenges whose purpose is to promote an improved image of computing.

2.2. Innovation

In his invited address to the Grand Challenges Conference, Peter Denning drew attention to a number of important matters. First, the size of the discipline of computing has grown and continues to grow at a dramatic rate. We can find the evidence for this by looking back at curricula development activities, e.g. those sponsored by the ACM and the IEEE-CS. In 1970 the number of core technologies was identified as being four; in 1989 that number had risen to nine; and by 2001 when Computing Curriculum 2001 was published, the number was in excess of thirty. Second, people viewed the curriculum as being too complex and too crowded. Evidence for this is the dropout rate in many institutions, which often is as high as 30–50%. Third, despite the many advances, we often see computing courses perceived by students to be dominated by programming.

In observing these matters, it is important to recognize the trends. If we allow these trends to continue unchecked, then the picture will worsen even further. The one certainty is that technological change will continue and the pace of it is unlikely to diminish. So issues of complexity will become even more acute and problematic. Yet it is vital, as part of education, that students learn how systems work and they must be able to identify ways of producing better and yet simpler systems.

An additional important observation is that, over the last 30 years or so, there has been a dramatic shift in emphasis towards interaction; the original emphasis on the concept and the properties of algorithms are now less pronounced, though of course still important. Recent research results have even made claims about the additional computational power of interactive computing. Nevertheless, algorithms and traditional programming often still dominate the courses.

The pervasive nature of computing and the broad range of uses and applications suggest that there must be opportunities for students from a wide variety of backgrounds and with vastly differing skills to find challenging and attractive opportunities to study computing. To meet this, a rich variety of opportunities should exist. There is scope for programmes of study in which different kinds of prerequisites and differing kinds of emphasis exist.

2.2.1. Challenge

In the light of the discussion described for the situation above, the computing challenge for this area is as follows:

> Provide simpler models of computing as a discipline, and have this simplicity reflected in a better mix of high quality computing courses that genuinely accommodate a broad spectrum of student ability and interest.

2.2.2. Sub-challenges

Within the grand challenge for this section, it is possible to identify a number of interim challenges or sub-challenges that can inform and testify to progress towards meeting the main challenge.

(i) Recognize the need for a rich variety of degree programmes (at all levels) accommodating diverse interests and diverse abilities of students; this needs to include programmes which are technical in nature and challenging but in which the main emphasis is not programming per se; these degree programmes (undergraduate as well as postgraduate) must meet the needs and the interests of students.

(ii) Capture the relevant and significant history of the discipline ensuring that a range of appropriate role models emerges for aspiring students of computing and that these reflect the range of computing.

(iii) Develop new models of the discipline of computing which present it as simple and appealing and yet ensure that it is easy to teach and capture its deep structure; these need to recognize that the discipline is not yet fully mature and they should offer encouragement to furthering its maturity, e.g. by being able to accommodate new advances.

(iv) Provide an ontology of computing that reflects a simple view of the discipline and yet can be used as a framework to maintain materials and resources of various kinds that will support rapid and effective course development; this can then be seen as a resource to inspire new approaches to learning and teaching.

(v) Develop new and more effective ways of delivering the curriculum—and so focus on ideas and basic approaches to computing that appeal to and excite students.

(vi) Identify ways of using technology to more effectively deliver the curriculum and to highlight the excitement and benefits of innovation.

(vii) Recognise the vital importance of links with other disciplines in these matters—the light that can be shed and the possibilities for innovative learning and teaching.

2.3. Competencies

Much discussion has occurred regarding the pace of change in computing and the need for computing people to keep up to date with that change. At a more fundamental and personal level, graduates need to recognize that the currency and the quality of their skills and competency levels will be significant determinants of success throughout their careers. Several implications of this exist. First, it is important that higher education instils, at least initially, high levels of appropriate and useful skills. It also needs to provide the wherewithal to learn and to know how to improve learning so that graduates are able to stay up to date ever more efficiently and effectively. Finally, there needs to be an infrastructure to support graduates as they advance through their careers.

Although licensing and certification in computing are not widely practised now in the UK, they do exist to varying levels elsewhere in the world. Issues associated with globalization and the global nature of the workforce become relevant. In the UK the situation could easily change. Graduates need to be aware of the possibilities and the implications of such changes and need to take cognisance of them.

Issues associated with skills and competence relate heavily to lifelong learning. It is easy to be glib about the learning process. Fundamentally, it is both important and difficult. It concerns students entering higher education; their experience should mean that upon graduation they are far more disciplined, imaginative and efficient in this regard. For those in employment, it becomes an ongoing concern and requirement. When there is an aging workforce, there are additional particular issues. However, there are different levels of learning as exhibited by the existence of Bloom's taxonomy of educational objectives [2]. These different levels, as well as the associated degrees of commitment required to achieve these levels, need recognition and their consequences understood.

2.3.1. Challenge

In the light of the discussion described for the situation above, the computing challenge for this area is as follows:

> Ensure that the quality and currency of computing skills and competence are recognized as important by graduates throughout their career, and put in place an infrastructure to provide support and guidance on a career-long basis.

2.3.2. Sub-challenges

Within the grand challenge for this section, it is possible to identify a number of interim challenges or sub-challenges that can inform and testify to progress towards meeting the main challenge.

(i) Identify very clearly the personal and transferable as well as the technical skills (including the ability to learn effectively and efficiently and to innovate) that students should acquire throughout their programme of study in higher education.

(ii) Identify and then employ a phased development of all these skills, ensuring that the skill levels are such that graduates are internationally competitive in terms of their skills, have acquired a sound basis for lifelong learning, and so are equipped to maintain and improve their ability to learn and thereby the currency of their skills.

(iii) Ensure that graduates recognize that there are different skills levels and different competencies to suit different careers; the currency of their skill and the levels of competency (and the evidence for this) will be a significant determinant of success throughout their careers.

(iv) Ensure that innovation is recognized as a vital skill whose elements need to be taught, and the teaching of this should pervade and be properly integrated into all computing degrees (and all years of all degrees).

(v) Provide support for lifelong learning (especially for young computing professionals on whom much of the country's wealth depends) in the form of professional level qualifications and guidance and these should include incentives for excellence.

2.4. Programming issues

In considering programming (in the broadest sense), we address the whole process of software development that moves from identifying the main characteristics of a problem, through developing a systematic method for its solution capable of being undertaken automatically and ultimately presenting that systematic method as a precise expression in a specified programming language. This is a central element of the discipline of computing, an important practical skill for computing, and an essential component of the undergraduate curriculum. One can generally observe that a strong correlation exists between programming ability and other computing skills, reflecting, as it does, skills in abstraction, conceptualization, design and evaluation.

Yet, major concerns exist amongst the academic community internationally that when we set out to teach programming skills to students, we are less successful than we need to be and ought to be. As noted in other challenges in this series (what computing is; the pre-university terrain), it may be that, at an early stage, the view of programming as dry, uninspiring drudgery rather than an exciting creative pursuit discourages potential computing applicants. At the next stage, educators cite failure in introductory programming courses and/or disenchantment with programming as major factors underlying the poor student retention in computing degree programmes. Furthermore, it is particularly disappointing that even many of those who graduate from computing courses express a dislike of programming and a reluctance to undertake it.

From the point of view of degree programme designers, the objectives in this area are straightforward. Educators would agree that it is desirable to employ the following six 'right' attributes. Teach

- the **right things** by selecting the right paradigms and using the right language, developing effective semantic models, taking the right approach to the life cycle, identifying the necessary levels of competency required—from 'novice' to 'legend'
- to the **right students** by correlating the balance of the material presented to students with their individual potential across a range of skills, so that we can identify those students who have the ability to become professional programmers, so that others can develop the ability to contribute to the wider range of computing activities, and so that those who seem likely to struggle can be identified swiftly and be appropriately advised
- using the **right methods** by choosing between different approaches (e.g. those based on formal definitions of syntax and semantics and those relying on informal description and example) between conventional lectures and practical classes and e-learning, collaborative learning, peer tutoring and other approaches, and using the right assessment and evaluation strategies
- at the **right time** in relation to the development of other knowledge and skills required for computing
- in the **right environment** with the right support tools—design and development environments including those which adapt to the user, compilers, operating systems, documentation aids, etc.
- using the **right teachers** who can convey knowledge, skills and, above all, enthusiasm.

2.4.1. Challenge

In the light of the discussion described for the situation above, the computing challenge for this area is as follows:

> Understand the programming process and programmer practice to deliver effective educational transfer of knowledge and skills.

2.4.2. Sub-challenges

Within the grand challenge for this section, it is possible to identify a number of interim challenges or sub-challenges that can inform and testify to progress towards meeting the main challenge.

(i) There appears to be layers of programming attainment, from simple sequential WWW scripting, programmed spreadsheet manipulation and basic system configuration in shell languages, through classic imperative programming, via object orientation and other paradigms, to advanced multi-lingual programming skills. A significant milestone in meeting this overall grand challenge would be to clarify and characterize such layers of attainment, the relationships amongst them, and how best to teach them both individually and in relation to each other.

(ii) The concept of IQ,[1] though controversial, provides a measure, relative to the whole population average, of innate intellectual capability, independent of age. Similarly, we suggest a sub-challenge to develop a 'programmers quotient' to give a measure of programming ability, relative to the whole population, that would remain the same independent of programming experience.

(iii) A more specific, technically focused, sub-challenge is to build a 'one-to-one programmer's assistant'—a software aid that would assist the development of individual programming skills, adapting both to different individuals and to the evolving skills of a single individual.

2.5. Formalism

Some level of mathematical thinking is important for all those who are involved in computing. It encourages and facilitates clear and rigorous thinking as well as sound argumentation. We typically base this on the provision of sound evidence and the application of logical principles that result in sets of well-founded conclusions.

The nature and the level of mathematics needed by students of computing will vary depending on the particular choice of the programme of study. For many, mathematical ideas and concepts will be required. These tend to be grounded in mathematical logic and discrete mathematics. Such simple ideas as providing a mathematical definition carry over almost exactly to computing. Mathematical logic provides the formal basis for reasoning and deduction. Within a course on discrete mathematics, topics include the basic ideas associated with structuring and they highlight abstract concepts with their properties that commonly occur in a computing context.

These studies encourage and facilitate clear and rigorous thinking using logic with the provision of the vocabulary and the notations of mathematics. They foster the use of convincing and soundly based arguments founded upon sound evidence and expose false reasoning and counter-intuitive situations as being flawed. These cognitive skills appear not just in formal situations but also in reasoning about computing related issues.

[1] The intelligence level determined from the ratio of mental age to chronological age (the actual age of a student according to birthdate). Mental age is determined by the tasks a student can perform at a certain age and an age scale.

2.5.1. Challenge

In the light of the discussion described for the situation above, the computing challenge for this area is as follows:

> Ensure that students of computing see relevant mathematics and formalisms in a very positive light, as providing support, guidance and illumination.

2.5.2. Sub-challenges

Within the grand challenge for this section, it is possible to identify a number of interim challenges or sub-challenges that can inform and testify to progress towards meeting the main challenge.

(i) Recognize that different levels of mathematical competence in different areas of mathematics are required to underpin a deep understanding of the legitimate different flavours of degrees in computing.

(ii) Characterize these areas and levels of competency, and identify the crucial ideas and principles of mathematics that underpin them.

(iii) Having identified those mathematical ideas that are important, weave them into a coherent and attractive curriculum; find a variety of mechanisms to ensure that they are made readily accessible (through tools, visualization or whatever) to students who will not feel threatened but are excited and supported by them.

(iv) Ensure the routine use of tools to carry out mathematical and logical manipulation automatically (ideally as a background activity) so that developers are not distracted from the important practice of design.

(v) Ensure the firm establishment of suitable formalisms in computing curricula and ensure that they routinely bring benefit.

2.6. About e-learning

'The challenge of e-' refers to the challenge of making effective use of information and communications technology (ICT) in the 21st century rather than transferring ineffective techniques from the 20th century. It refers to the challenges posed by information born in digital form, by people born into a digital world, and by new business infrastructures that make more complex and more sophisticated the interchanges and transactions that take place over them. It refers to the challenges of linking people, their homes, their work, their study and their leisure activities to sources of information that are available anywhere at any time.

Why is 'e-' important? First, it gives the illusion of knowing everything—to be more precise, given a modest personal computer and an Internet connection, ICT has the potential to deliver facts about almost any subject at the touch of a few keys. As a mark of the impact that this phenomenon has had on lives, the verb 'to google' is quickly passing into the language. Second, where facts alone are sufficient, 'e-' removes the need to know anything. There is no longer a need to remember the times of trains, or the date of the Great Fire of London, for instance. The 'e-' replaces the skill of memory by the skill of being able to find out—'just in time' information, rather than 'just in case'. Third, the 'e-' displaces space and time. We are as likely to receive electronic mail from someone in the next office, the next house or the next city as we would the next country. We receive it when it becomes convenient for us; the senders send it when it is convenient for them. In many cases, we perform shopping more easily from home than from the high street. Hence, 'e-' becomes the natural medium for collaboration.

For those in education, the challenge of 'e-' is that these characteristics—the illusion of knowing everything, 'just in time' information, and the displacement of space and time—are being applied to education by our students. Some of them are millennial students, natural collaborators born into a digital world, who are as likely to work between 5 pm and 9 am as their teachers are to work between 9 am and 5 pm; they work together, take the World Wide Web as common knowledge, and are surprised by regulations on plagiarism. Others balance study with paid employment or family responsibilities. Some just want to learn for learning's sake. How do we develop and support such online communities?

2.6.1. Challenge

In the light of the discussion described for the situation above, the computing challenge for this area is as follows:

> Establish e-learning as a credible, viable complement to face-to-face education.

2.6.2. Sub-challenges

Within the grand challenge for this section, it is possible to identify a number of interim challenges or sub-challenges that can inform and testify to progress towards meeting the main challenge.

(i) Devise systematic and disciplined approaches and principles that one can reliably employ to guide the development (by informed teachers) of high quality e-learning materials. These approaches can mirror corresponding concepts from engineering, which encourages reuse and interoperability, replicable development of high quality materials as well as effective process models seen as indicators of well-controlled and mature processes.

(ii) Develop appropriate development languages, tools and systems that support and encourage the ready creation of materials which address issues of customization (including personalization and other aspects of context) and exhibit intelligent and adaptable behaviour as well as take advantage of the benefits of digital libraries which are themselves evolving and changing dynamically.

(iii) Carefully address usability requirements ensuring that such systems are attractive, appropriately challenging and interesting to use for a wide variety of prospective users, including those who benefit from multimedia and other forms of interface. Such experiences should ensure that e-learning is, from a learning perspective, a sustainable engaging experience; related to this is re-interpreting the notion of quality in the context of e-learning systems.

(iv) Provide reliable and effective approaches, within some carefully defined and controlled context, to online assessment (which take account of issues such as plagiarism) and provide useful feedback to students.

2.7. Pre-university issues

Universities recruit students from high schools and colleges and then transform them into graduates. Students come from disparate backgrounds and offer a wide range of entry qualifications. As official governmental policies tend to widen participation, this trend will continue and the accepted benchmarks regarding entry may seem less applicable.

Computing is in an odd position. Currently, no universally accepted pre-university computing qualification for entry exists; many programmes will require some level of mathematics, for instance, but few if any require a prior qualification in computing. A steady but clear growth in demand for university computing places has existed for many years; this growth peaked during the 'dot-com' bubble and has now fallen back to a point nearly in line with the long-term trend. In the short to medium term, the trend should continue.

The vocational nature of the subject makes it a popular choice for many students, especially as universities demand explicit sums of money for their wares. The motivation for this choice is not always primarily academic. There is often a very poor understanding by new students of what the subject is and what they will study (often related to the preceding point). In computing, the issue is exacerbated by the wide exposure of the pre-university community to computers in many routine ways, often misleading students about the nature of the subject.

The subject appears in some guise in most schools. Professionally, however, computing qualifications are widespread, and information technology (or ICT as it is sometimes called) appears as a qualification and manifests as a 'key skill'. Abundant anecdotal evidence and statistics reveal that the number of qualified computing graduates in school staff is very low. Teachers with a first speciality in another subject usually purvey computing. There is no evidence that computing is inadequate or poorly taught (indeed, evidence to the contrary is available), but there is typically an absence of the enthusiasm and knowledge that comes with the depth of a higher educational qualification in computing itself.

There is a significant influence on computing in higher education from the 'demand' side, namely students and employers. Part of this influence is on the choice patterns of entrants, but there is also an effect on the curriculum, both implicitly from university staff knowing (or thinking that they know) the requirements of employers, and explicitly via a number of organizations and government agencies enumerating 'what they want' [3].

2.7.1. Challenge

In the light of the discussion described for the situation above, the computing challenge for this area is as follows:

> Rationalize the situation at the pre-university level directed towards the promotion of computing to would-be students of computing. Create for students a smooth transition from school to university by enthusing and informing potential students and by creating a positive influence affecting pre-university computing.

2.7.2. Sub-challenges

Within the grand challenge for this section, it is possible to identify a number of interim challenges or sub-challenges that can inform and testify to progress towards meeting the main challenge.

(i) Engender in potential students a sensible understanding of the range of possibilities of advanced study of computing and, where appropriate, ensure that they possess the skills needed to undertake successful study of (some aspect of) computing in higher education.

(ii) Communicate to teachers and career advisors what university computing and computing careers are, and what computing degrees might lead to.

(iii) Understand/verify reasons for the low percentage of some population sectors (notably women) among applicants and, in a complementary manner, the reasons for the disproportionate appeal of the discipline to some other groups.

(iv) Ensure that those entering higher education to study computing form a representative cross section of society in terms of gender, ethnic origin, etc. and yet also compensate properly for diversity in intake, expectation, ability, maturity and experience.

(v) Establish resource (time) for teachers and university staff to develop a mutual understanding of one another's problems. Encourage university staff to appreciate that students' expectations are coloured by their life experience and not by the history of computing. Understand the prerequisites required to support successful study of the different flavours of computing in higher education, bearing in mind that introducing new prerequisites conflicts with aims of widening participation.

(vi) Have pre-university delivery of high quality computing courses informed directly by and consistently performed by appropriate active academic computing graduates.

(vii) Develop a clear view of the issue of programming at the pre-university level and have a positive influence on the development of appropriate curricula.

3. CONCLUDING COMMENTS

It would be a mistake to believe that this list of challenges is complete. At best, one should view them as an initial attempt to draw attention to some significant issues whose resolution, at the time of writing, has the capacity to improve significantly aspects of education in computing.

It would also be a mistake to believe that the challenges summarized in this paper are all independent of one another; indeed, there are interesting relationships between several of them.

An additional important observation is that grand challenges of this sort should have a review process at some stage, perhaps after several years. Should the concept prove effective, there will be a need to refocus and perhaps to reformulate the challenges with a view to ensuring that education in computing moves forward and remains in a healthy and vibrant state. The educational challenges addressed by the community must themselves remain current and capture the important concerns of the day.

REFERENCES

[1] McGettrick, A., Boyle, R., Ibbett, R., Lloyd, J., Lovegrove, G. and Mander, K. (2004) *Grand Challenges in Computing: Education*, full report of the educational strand of the Grand Challenges Conference. British Computer Society, Swindon, England.

[2] Krathwohl, D. R., Bloom, B. S. and Masia, B. B. (1956) *Taxonomy of Educational Objectives: The Classification of Educational Goals*. Longmans Green and Co. Ltd, London.

[3] Career Space (2004) http://www.careerspace.com/project_desc/index.htm