Grand Challenges

in Computing

Education

Andrew McGettrick, Roger Boyle,
Roland Ibbett, John Lloyd,
Gillian Lovegrove and Keith Mander

Grand Challenges in Computing –

Education

Andrew McGettrick, Roger Boyle,
Roland Ibbett, John Lloyd, Gillian Lovegrove,
Keith Mander

BCS

Contents

The British Computer Society

The British Computer Society (BCS) is the leading professional body for the IT industry. With members in over 100 countries, the BCS is the professional and learned Society in the field of computers and information systems.

The BCS is responsible for setting standards for the IT profession. It is also leading the change in public perception and appreciation of the economic and social importance of professionally managed IT projects and programmes. In this capacity, the Society advises, informs and persuades industry and government on successful IT implementation.

IT is affecting every part of our lives and that is why the BCS is determined to promote IT as *the* profession of the 21st century.

Joining the BCS

BCS qualifications, products and services are designed with your career plans in mind. We not only provide essential recognition through professional qualifications but also offer many other useful benefits to our members at every level.

Membership of the BCS demonstrates your commitment to professional development. It helps to set you apart from other IT practitioners and provides industry recognition of your skills and experience. Employers and customers increasingly require proof of professional qualifications and competence. Professional membership confirms your competence and integrity and sets an independent standard that people can trust. www.bcs.org/membership

Further information

Further information about the British Computer Society can be obtained from: The British Computer Society, 1 Sanford Street, Swindon, Wiltshire, SN1 1HJ.

Telephone +44 (0)1793 417424
Email: bcs@hq.bcs.org.uk
Web: www.bcs.org

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 that it has become one of the largest areas of study within 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. For some, the discipline somehow does not seem to be so appealing. There are problems of retaining students within computing programmes, the study of the discipline seems to be difficult and complex, and jobs are no longer currently in such plentiful supply (perhaps partly caused by the phenomenon of outsourcing and off-shoring). The media do not help the situation, as they tend to focus on disasters of various kinds rather than on the innumerable success stories, and thereby foster uncertainty about the discipline.

A perhaps somewhat dramatic perspective is that computing may be viewed as a discipline in crisis. In reality, the popularity of the field has just returned to the 'pre-dot com boom' situation. Yet society is ever more dependent on computers and computing and this dependence will only increase in importance over time.

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; that is, it is time to question aspects of the existing situation, to ask what to do and to seek remedies. Almost inevitably, this involves focusing 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 30–31 March 2004, occurred at a particularly opportune time. This was not a conventional conference usually held to address solutions to problems, but instead it was about identifying problems. That is a significantly different task. 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, although 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 in 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.

For clarification, we use the term 'computing' in a generic sense to embrace traditional titles such as computer science, software engineering, information systems, artificial intelligence, computer hardware, communications, internet computing, and even aspects of cognitive science. In summary, it involves the technical aspects of computing systems (and often this involves considerations about users). It is important also to recognize that the community here is the computing community; the challenges therefore have to be phrased in such a way that they appeal to this group who should feel that it has the interest and the ability to develop the solutions.

This conference then presented an opportunity, not to solve problems, but to identify and articulate the priorities and the challenges that face educators in computing. An infrequent formal opportunity such as this brings with it certain responsibilities that need to be respected and recognized. In particular, it was important to articulate the challenges carefully and precisely and to try not to omit significant and important matters. It was also important to address carefully the level of challenge and to ensure that it is appropriate, while at the same time, not allowing purely local concerns to have major influences. We hope that the result will contribute to the scholarship agenda for the community.

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 for 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 elaboration. 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 of perception, social change, and changing attitudes. The effort involved in bringing about such change 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, even change at a 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 achieving the intended goal.

1.3 Approach adopted

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.

During the conference itself, six groups debated particular issues arising from the submissions and they were to come forward with significant challenges. Each group had to produce (at least) one grand challenge in the following format: title of challenge, motivation for that challenge, challenge explained and articulated, and sub-challenges to amplify the full range of issues. The resulting documents (suitably edited to address overlap, format, and other issues) form the basis of this document.

1.4 Contributions

The papers accepted for presentation at the conference included:

- Tom Boyle: The development of a systematic discipline of e-learning based on a synthesis of software engineering and pedagogy.

- Janet E. Carter: Students may be capable of formal reasoning, but they have yet to try.

- Lillian Cassel: An ontology of computing and information.

- Martyn Clark: Teaching computing in the liquid modern world.

- Fintan Culwin: Living with Pandora: Computing education in the internet age.

- Sharon Curtis and David Lightfoot: Teaching scientific techniques of software development.

- Peter Denning (invited speaker): (1) Great principles of computing; (2) Innovation and value; (3) Programming as a practice.

- Maria Fasli: Teaching agents and multi-agent systems.

- Neil Gordon: Mathematics and the computing curriculum.

- Gary Griffiths and Mike Lockyer: (1) Early life cycle emphasis. (2) Tools everywhere.

- Richard Hebblewhite, Stuart Cunningham and Vic Grout: Computing for the new generation.

- Mike Holcombe: Connecting the teaching of computer science with reality.

- M. Huggard: Programming trauma: Can it be avoided?

- Adrian Jackson: The public understanding of computing.

- Colin Johnson: (1) What do students want from computer science and should we give it to them? (2) A truly scientific computing education.

- Greg Michaelson: Accurate identification of potential computer programmers.

- Margaret Ross (for BCS): Realizing socially responsible computing.

- John Sargeant and Mary McGee Wood: Human computer collaborative assessment.

- Aaron Sloman: A new kind of liberal education.

- Mike Stannett: Can theory be made more attractive?

- Ian Utting: Mind the gap!

1.5 Additional contributors

Through participation in discussions and otherwise, the following made contributions to this conference in different ways:

Terry Carr, John Chapman, Richard Connor, Aline Cumming, Douglas Dankel II, Gordon Davies, Jenny Davies, Pierre-Arnoul de Marneffe, Enda Dunican, Clive Fencott, Sally Fincher, Mike Hender, Martin Henson, Tony Hoare, Matthew Jadud, Alan Jones, Paul Kelly, Myungsook Klassen, Pete Lee, Ian Marshall, Richard Millar, Les Neal, John Nealon, Stanley Oldfield, David Parsons, Geoffrey Sampson, Yuh-Huei Shyu, Andy Sloane, Peter Smith, Gill Sugden, Deborah Trayhurn, Des Traynor, Eileen Web, Yau Yip.

Important additional editorial input was also received from John Impagliazzo and Lillian (Boots) Cassel.

2 Challenge areas

2.1 Perception of computing

2.1.1 Situation

Public image and public understanding of any discipline will 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 computing career.

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 news. 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 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, where these developments 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.

For computing, the current public image tends to be less than ideal. The media will often focus on disasters of various kinds, on stories about new viruses and the damage these can do, about security breaches, about intrusions from ugly and unwanted spam, and generally on incidents that are criminal or border on the irresponsible. Those who use computers are often frustrated by curious software bugs, by software that will not perform as required, by systems that are too complex to understand and too difficult to use. Of course, much of this relates to the nature of software itself; nevertheless, the image portrayed is not kind.

2.1.2 Challenge

Given the situation described 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 practise within it.

2.1.3 Motivation

As advances unfold and future progress towards ubiquitous computing emerges, the unsavoury matters (associated with spam, security violations etc.) could have the potential to strangle developments in computing. In all of this there are medical analogies, reflected in the use of terms such as 'virus' and 'epidemic', where the solutions can also be couched in medical terminology.

To address these concerns a number of initiatives have been undertaken. For example, in security and related matters, grand research challenges from Computer Research Associates (CRA, 2003 a and b) include (but are not restricted to) the elimination of epidemic-style attacks (from viruses, worms, email spam etc) within 10 years and the development of tools and principles that allow construction of large-scale systems for important societal applications, such as medical records systems, that are trustworthy despite being attractive targets. Another is to build distributed computer systems, which in the event of some major disaster can be relied on to continue to function effectively and efficiently (Jones *et al.*, 2003).

Looking at the scene from a different perspective, IBM has initiated the concept of autonomic computing (Ganek and Corbi, 2003). The aims of autonomic computing are to produce systems that are

> intelligent in the sense that they understand themselves and are able to manage themselves so that they adapt to both predictable as well as

> unpredictable situations or occurrences, even tune themselves and recover
> from failure.

The intended result is that there is a decrease in the need for tedious maintenance and a consequent reduction in the frustration of users. The resulting systems can be described as self-managing, self-healing, self-protecting, and generally are able to self-diagnose, to predict and to adapt to dynamically changing environments and situations. There are, of course, implications here for the safety and for the security of such systems.

In the area of usability concerns, one of the grand challenges from Jones *et al.* (2003) concerns building systems in which users, software agents, and robots can work in harmony to enhance the productivity and capability of an individual. This is particularly relevant when dealing with such matters as natural language translation in real time in a variety of user contexts and situations, compensating for disability, and so on.

These are only examples of the possibilities and they hint at a considerable cross-fertilization of ideas. For instance, maturity models have their origins in studies of the software development process. In the case of autonomic computing, for example, a similar kind of maturity model can be associated with the development of such systems: thus, there is a basic level, a managed level, a predictive level, an adaptive level and a fully autonomic level. There are mechanisms and steps that we can define and describe to aid transition from one level to the next. This idea of levels is significant and has wider applicability.

Another important aspect of the required image of computing is that it should be outward looking and inclusive. In part this is to reflect a discipline that remains in a rapid state of evolution and consequently there is merit in seeking ideas and concepts from other disciplines and elsewhere. These can aid and encourage (perhaps through abstraction, absorption and/or cross-fertilization) the future development of computing. Therefore, in creating an image of computing, it is important to project a discipline with a strong central body of knowledge with sound underpinning etc, but for which it is important to illuminate the beneficial relationships with other disciplines.

What are the factors that influence the public image? Certainly, there are the matters reported in the press and in the media generally; mechanisms exist for influencing these and they should be used. In these endeavours it is important to draw attention to the ever-increasing range of the important applications and positive images involving computers coupled with the benefits these bring to individuals, to society, to industry, to commerce, and so on. Other influences include such matters as the reliability of many computer systems, the trust that the public can place in them, and the ease of use of such systems. The challenge is to place positive images before the public to balance and even negate the reports of disasters and failures.

The very concept of a grand challenge itself merits attention from another perspective. We can use the existence of these challenges to fire the imagination, to tell the public (as well as others, e.g. those in positions of influence) about those matters that the computing community sees as being important, and why. Apart from other considerations, this serves to reinforce the view that there remains much to do in computing and the products of this endeavour have the potential to bring huge benefit and advantage.

2.1.4 Sub-challenges

Within this grand challenge 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:

- 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.

- 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 extent of computing as a discipline and to further the development of computing as that discipline.

- 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.

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

2.1.5 Measures of success

We wish to associate various metrics with this challenge as noted in the sub-challenges above.

2.2 Innovation

2.2.1 Situation

In his invited address to the Grand Challenges Conference, Peter Denning drew attention to a number of important matters:

- 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, for example those sponsored by the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers Computer Society (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.

- People view 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 per cent.

- 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. Therefore, 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 simpler systems.

An additional important observation is that over the last thirty years or so, there has been a dramatic shift in emphasis towards interaction; the original emphasis on the concept of and the properties of algorithms are now less pronounced, although 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 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. There has to exist a rich variety of programmes of study with different kinds of prerequisites and differing kinds of emphasis.

2.2.2 Challenge

Given the situation described 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 accommodates a broad spectrum of student ability and interest.

2.2.3 Motivation

There is a natural tendency to emphasize undergraduate programmes. In fact, postgraduate programmes also need to be scrutinized to ensure that they are well founded and have real appeal. Issues of the number of student applications, and issues of successful completion are just as valid and important in these contexts.

Certain questions arise about fundamental issues. In a thought provoking private communication, Tony Hoare writes:

> The fundamental question in any branch of engineering is 'how does it work?' The fundamental question in any branch of engineering science is 'why does it work?' These are exactly the fundamental questions of software engineering and computer science. We must therefore inspire our students with curiosity about these questions, and then satisfy it. In most branches of engineering, this requires some mathematics, and the students often object. We just have to live with that.
>
> Of course, there will always be students who have little curiosity, and only want to learn 'how to do it' so that they can earn lots of money. They are the taxi drivers of the world, not the automobile engineers. We need them, and they need vocational courses in how to set up and maintain computer networks, databases, websites etc. Perhaps they don't even need to know how these systems work, but in a university they should be taught all the same, because one day it might help.

As a discipline, we must still view computing as being in a state of evolution. In seeking simpler models of computing, it is desirable to:

- have models that are simpler than the current possibilities, and are attractive and interesting;

- have new models that reflect insights into the deep structures of the discipline, and reflect fundamentals; in the process of their development there is scope for combining knowledge areas where appropriate, recognizing and drawing out relationships and even identifying new structures;

- ensure that the new models accommodate and indeed are unaltered by changes in technology and changes in fashion (e.g. choice of programming language, choice of tools) and yet accommodate fundamental advances;

- reflect a theoretically sound and socially responsible and professional approach to the discipline and yet recognize the importance of 'learning by doing'.

In seeking to address these important matters, there is a need to stand back and to rethink radically many issues associated with the computing curriculum and its delivery. Existing approaches are proving to be inadequate.

Peter Denning (2003) puts forward a new view of how to present and describe this discipline of computing. Of course, he is fully aware of the need for comparisons and welcomes alternative views aimed at making computing simpler as a discipline as well as more attractive, more appealing and more alluring.

The breadth and the importance of computing are now such that there must exist degree programmes to accommodate a wide spectrum of students with diverse interests and diverse backgrounds. This strongly suggests the need for a rich variety of well-designed and appealing degree programmes that exhibit different characteristics, impose different demands, and open up different future possibilities. There is a need to recognize that, for example, not everyone needs to be an expert programmer and that there can exist various other extremely useful practical skills that are equally demanding, interesting and useful and around which degree programmes can be structured.

A number of rather specific requirements have emerged from major studies. For example, the National Science Foundation (NSF) in the USA stated:

> **A new interdisciplinary work force:** The need for a new workforce - a new flavor of mixed science and technology professional - is emerging. These individuals have expertise in a particular domain science area, as well as considerable expertise in computer science and mathematics. Also needed in this interdisciplinary mix are professionals who are trained to understand and address the human factors dimensions of working across disciplines, cultures, and institutions using technology-mediated collaborative tools. Prior work on computer-supported collaborative work and social dimensions of collaboratories needs to be better codified, disseminated, and applied in the design and refinement of new knowledge environments for science based on cyber infrastructure.
>
> (NSF, 2003)

Similarly, The Royal Academy of Engineering (RAE) and The British Computer Society (BCS), published a study and concluded:

> Education in many universities and management schools . . . is not producing IT practitioners with the IT application and project skills they need
>
> . . .
>
> The importance of project management is not well understood and usually underrated, and senior managers are often ill-qualified to handle issues relating to complex IT projects
>
> . . .
>
> Risk management is critical to success in complex projects but is seldom applied effectively in the case of IT and software
>
> . . .
>
> The vital role of the systems architect in major IT projects is frequently not appreciated and there is a shortage of appropriately skilled individuals.
>
> (RAE, 2004)

For institutions such as the ACM or the IEEE-CS, which offer valuable curricular advice to the community, this presents a possible difficulty. The concept of a very small number of well-defined degree programmes from which we can distil advice runs counter to the idea of a rich diversity of interesting programmes of study. Of course, this raises questions about standards of degrees. However, we can address these issues by setting criteria as described by the UK Quality Assurance Agency (QAA, 2000).

The issue of curricular advice is a separate matter. One could address this by developing an ontology for computing, ideally based around some simple model for the discipline. This would then form the basis for a system that we can populate with curricular materials, advice and general resources that facilitate the whole process of course and curricular development.

To help the learning and teaching processes it is desirable to find better ways of presenting the curriculum. It would be a mistake to believe that the best approaches are those currently used. For example, there is considerable scope for finding new approaches to design that are better and more natural than existing methods (e.g. methods derived from natural phenomena). We can find new and better ways of teaching (e.g. finding approaches that have greater visual appeal and/or that fire the imagination of current students) and of engaging *all* students in active learning. There are numerous other possibilities.

2.2.4 Sub-challenges

Within this grand challenge 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.

- 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 (at all levels, undergraduate as well as postgraduate) must meet the needs and the interests of students.

- 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.

- Develop new models of the discipline of computing that present it as simple and appealing and yet ensure that it is easy to teach and that 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, for example by being able to accommodate new advances.

- 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.

- Develop new and more effective ways of delivering the curriculum, and so focus on ideas and basic approaches to computing that excite students.

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

- In these matters to remember the vital importance of links with other disciplines, the light that can be shed and the possibilities for innovative learning and teaching.

2.2.5 Measures of success

We can associate various metrics with the challenge and sub-challenges as follows:

- Notwithstanding the range of academic backgrounds of entrants, they successfully complete their first year of study of computing seeing it as interesting and challenging, and by the end of that year are involved in some novel application development which is at the forefront of technological change.

- During their first year, entrants experience at least one novel use of technology in curriculum delivery and are able to assess its effectiveness.

However, ultimately quantitative measures relating to students successfully completing their degree provide better focus.

2.3 Competencies

2.3.1 Situation

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 their 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. The situation could easily change in the UK as issues associated with globalization and the global nature of the workforce become even more relevant. Graduates need to be aware of the possibilities and the implications of such changes.

One can address such matters within courses on professional issues. Unfortunately, students often see many of these courses to be of marginal relevance to their technical development. Indeed, if presented in the wrong fashion, then the legal restrictions of certain kinds of data protection, confidentiality and privacy rules etc, can act as a deterrent and inhibitor to certain kinds of development. We need to take steps to approach the teaching of these matters in a more positive fashion. We might observe, for example, that courses on human computer interaction could address the legality of asking users to supply personal information in certain contexts. The ethical and professional issues associated with outsourcing and off-shoring, as well as issues associated with software patents, have significant relevance to students and their future careers. Furthermore, standards play an important role in computing and are highly relevant.

The 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, on 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 (for instance, see Krathwohl, Bloom and Masia, 1956). These different levels, as well as the associated degrees of commitment required to achieve these levels, need recognition and the consequences need understanding.

2.3.2 Challenge

Given the situation described 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.

2.3.3 Motivation

Embarking on learning initially requires planning, and this benefits from guidance and advice, and even some vision. Certain kinds of learning will bring about fundamental changes in approach and in attitude and take time, commitment and even on occasion require real courage. During a lifetime, one would likely require such major change on several occasions.

As mentioned earlier, the concept of lifelong learning is important. It is also important to see that a phased or staged approach to the development of the lifelong learner is required. Fellows, Culver, Ruggieri and Benson (2002) identify four stages, namely the *dependent* learner who relies on others for guidance and support, the *interested* learner who is showing enthusiasm and commitment, the *involved* learner, and finally the *self-directed* learner. Thus, a first year in higher education might address the first stage in a carefully planned manner. The second year might address the development of the interested learner, and so on throughout the years of a programme of study.

These ideas apply more generally. A clear requirement of higher education is to identify the set of personal and transferable skills as well as the technical skills that a student has to acquire in their programme of study. Then, a planned or phased development must be put in place for each year of their course and this should culminate in genuinely skilful, confident, and mature graduates.

One of the important matters raised by Peter Denning in his invited talk is the importance in the modern world of innovation. This is similar to, but different from, entrepreneurship. Innovation is not solely about the discovery of a new idea or the development of new software. It is a skill that can be taught in the

context of computing and it benefits from a phased approach of the kind discussed above in relation to lifelong learning. However, we rarely teach this: and that needs to change. The elements of innovation need consideration and we must provide a structure for the staged approach that will ensure its proper place and proper integration into the computing curriculum.

We can regard innovation as one instance of a new skill. It is also important to recognize that with technological development, new skills will be a requirement. For instance, the RAE and BCS provide hints that perhaps risk analysis and risk assessment in their many forms should become essential (RAE, 2004). The academic community needs to be alert to these shifts and changes.

It is certain that students must become acutely aware of the currency of their repertoire of skills. This is something that they will need to be alive to throughout their working life – something related to the rate of change. They will need to both refresh and refine it. However, they need support for this just as they need support throughout their earlier education. Mechanisms need to exist to provide that support. Thus, a certain infrastructure should exist to guide and help them in their future careers.

An important audience for the lifelong learning agenda is the young (even not so young) professional on whom much of a country's wealth depends. These professionals need to be highly skilled, highly motivated, and highly effective in their roles. They need to be kept up-to-date, but because of the pressures and demands on them, every attempt needs to be made to ensure that learning is enjoyable, rewarding, effective, efficient and beneficial in terms of career development. A technically-based infrastructure ought to exist to support and guide them in their careers. The challenge here is considerable, and very important.

2.3.4 Sub-challenges
Within this grand challenge 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.

- 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.

- 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, and have acquired a sound basis for lifelong learning (so maintaining and improving their ability to learn and thereby the currency of their skills).

- 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.

- 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).

- 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.

2.3.5 Measures of success
We can associate various metrics with the challenge and sub-challenges as follows:

- Provide an infrastructure to support lifelong learning of professional engineers within five years and ensure its wide use within organizations within ten years.

- Provide evidence of a systematic and phased approach to the development of a range of competences for differing computing programmes recognizing the issue of levels.

2.4 Programming issues

2.4.1 Situation
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.

However, major concerns exist among 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. Further, 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 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 appropriately advised;

- using the right methods by choosing between different approaches: for example 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 ... ;

- using the right teachers who can convey knowledge, skills and, above all, enthusiasm.

2.4.2 Challenge

Given the situation described 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.

2.4.3 Motivation

The particular concern is that, after more than forty years of teaching an essential aspect of our discipline to would-be computing professionals, we cannot do so reliably. Indeed, there are perceptions that the situation has become worse with time. The situation appears compounded by the broader profile of current students and perhaps by the growing complexity of programming languages, development environments and other tools, the pressure to use commercially relevant languages as a vehicle for teaching, and the reduced familiarity of our students with mathematical concepts as well as their reduced confidence with systems of notation.

A symptom of the problem is that much of the above 'analysis' comes from anecdotal evidence; there are very few rigorous evaluations of pedagogy in this area. There is also a feeling that, in one-to-one situations between teacher and student - a master-apprentice environment - we could achieve a much higher success rate, not least by differentiation of approach. From this perspective, the grand challenge is to identify what is 'right' in each of the six contexts mentioned in section 2.4.1.

The costs of not meeting this grand challenge are significant. First, there will continue to be unacceptable failure rates in programming courses and programmes that depend on them. Second, students and staff involved in such courses and programmes will continue to experience frustration and demoralization. Finally, the expenditure of educational resources will continue only to achieve poor results.

The benefits of meeting this grand challenge are also significant. First, we may identify students who are not suited to programming and give them better educational guidance at an earlier stage. Furthermore, we might teach programming more effectively, making better use of resources and with greater student and staff satisfaction. Finally, we will have a better understanding of a major human cognitive ability.

We believe that this challenge requires a series of systematic cross-disciplinary studies engaging not only computer specialists, but also anthropologists, psychologists, educationalists, as well as practitioners from other disciplines, so that in the long term, we can supplement the present anecdotal observations with evidence. In particular, substantial studies are required to identify current best and worst practice in programming education and locate or construct accurate quantitative indicators of potential programming skill. Furthermore, it is important to characterize cognitive features of acquisition of programming skills and articulate a constructive path from school to university level study of programming.

2.4.4 Sub-challenges

Within this grand challenge 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.

- There appear to be *layers of programming attainment*, from simple sequential web 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 between them, and how best to teach them both individually and in relation to each other.

- 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 challenge to develop a 'Programmer Quotient' to give a measure of programming ability, relative to the whole population, that would remain the same independent of programming experience.

- A more specific, technically focused 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.

2.4.5 Measures of success

We can associate various metrics with the challenge and sub-challenges as follows:

- Within five years, students should not be dropping out of computing programmes because of lack of ability in programming.

- Programming courses should be seen to be attractive and enjoyable.

2.5 Formalism

2.5.1 Situation

Some level of mathematical thinking is important for all 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 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 typically include basic ideas associated with structuring and they highlight abstract concepts that commonly occur in a computing context including an investigation of their properties.

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 on firm 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 birth date). Mental age is determined by the tasks a student can perform at a certain age and an age scale.

2.5.2 Challenge

Given the situation described 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.

2.5.3 Motivation

In the current state of the art, mathematical ideas are important for the provision of formal specifications, for model checking, for the formal verification of the correctness (as well as other properties) of code and for the equivalence of program transformations as well as for the relevance of situations related to the applicability of these transformations. In these and in other cases, mathematical manipulation is employed and much of this can be accomplished automatically using appropriate tools, accompanied by suitable reuse of proofs.

There are many other areas where mathematical concepts and techniques can be used to beneficial effect in computing. There is reasoning in artificial intelligence systems; there is the role of number theory and the theory of elliptic functions in cryptography; there is the role of combinatorics in complexity theory, and so on. Mathematics also has roles in information theory, error-correcting and error-detecting codes, in automata, in graphics and visualization, in probability and statistics, and so on. In practice, mathematical reasoning benefits from the provision of suitable (user-friendly) tools that can undertake much of the mathematical manipulation automatically. Indeed, from a computing perspective, there is merit in relegating as much of this mathematical manipulation as possible to background activity so freeing the human to focus on the difficult but important matter of design.

At a more general level, mathematics typically encourages facility in problem solving via the development of analytical skills. Through familiarizing students with many discrete structures, it provides support for the important ability to become comfortable with the notion of abstraction and to employ it appropriately in computing.

At the more advanced levels, researchers provide deep arguments explaining the great relevance and importance of mathematical concepts such as categories and bi-graphs. Yet much of this material is highly formal, providing mechanisms for describing deep structures and relationships. Much of this is currently inaccessible to almost all involved in computing. If it truly is important, then we need to take steps to uncover the secrets, make its relevance apparent, and show that its utility and applicability are readily and easily accessible. Since formality itself is the key to automation, it should be readily possible to devise steps whereby this can occur.

Formalisms as employed in computing rely heavily on mathematics: on mathematical notations, on mathematical techniques, on mathematical manipulation and on mathematical insights. In some sense, the true impact and power of these methods needs to carry greater conviction and they should demonstrate significant impact and benefits across the range of computing. This suggests encouragement for the development of new formalisms, which are easier to understand and use and are more powerful than existing possibilities. We need to establish firmly formal methods themselves as providing a sound engineering approach to software engineering by showing significant benefit in areas such as specification, design, and reuse. Furthermore, these methods need to be better supported and far more 'user friendly'.

In the world of education, great problems exist in the mathematical area. Students are less confident than ever in their mathematical abilities and there are threats that the situation is unstable; it may get worse or it may get better. Many would claim that the matter is exacerbated at the point when students embark on higher education and this is caused by a gulf that exists between the expectation of academic staff and the reality of student ability; the latter is often caused by changes to school curricula that seem to be invisible to university staff. The situation on this interface can change from year to year and so it is necessary to be vigilant in this regard. In short, claims are made that general higher education is not sufficiently well attuned to the changing abilities. This needs addressing.

2.5.4 Sub-challenges

Within this grand challenge 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.

- 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.

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

- 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.

- Encourage 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.

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

2.5.5 Measures of success
We can associate various metrics with the challenge and sub-challenges as follows:

- Diagnostic testing can be employed to evaluate the mathematical ability of students (e.g. at university entry) so that they can be properly guided in their programme of study. In addition, carry out diagnostic tests at a later stage to allow comparisons about the effectiveness of formalisms.

- Degree programmes should possess appropriate sound mathematical underpinning as well as the elements of appropriate formalisms. Within five years, students should not be failing or having to withdraw from programmes of study based on their inadequate mathematical ability.

2.6 About e-learning

2.6.1 Situation
'The challenge of *e-*' refers to the challenge of making effective use of information and communications technology (ICT) in the twenty-first century rather than transferring ineffective techniques from the twentieth 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 interchanges and transactions within 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, *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 the 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 'modern' 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 its own sake. How do we develop and support such online communities?

2.6.2 Challenge
Given the situation described above, the computing challenge for this area is as follows:

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

2.6.3 Motivation

In a seminal study, Peter Lyman and Hal Varian (2000-3) estimated that in 1999, 250 MB of new information was published for every single person on the planet. That is about 1.5×10^{18} bytes of information, which is a very great deal of information indeed – and an amount that can only be expected to grow in the future. In 2003, they estimated that the figure of 250 MB had already risen to 800 MB. In an age of such vast quantities of information where much of it is available on or through the internet, we can now expect that a significant proportion of the sum of human factual knowledge will be encoded and readable from a computer within ten years.

Key issues arise such as the skills needed to access this information and the way we can use this information to support learning and education. Setting aside broadcasting and telephone, the internet includes surface web, deep web, email, messaging and files. These facilities provide opportunities for learning and, if appropriately harnessed, can underpin e-learning systems.

The whole process of engaging with e-learning raises a number of fundamental questions. They include the following:

- How can we take locality and context into account and use it to create effective systems?

- What are the business models that will determine supply, billing and access, and on-demand systems?

- How can we incorporate effectively entertainment-based systems, simulation, and the power of the visual and multimedia systems?

- How can we incorporate effectively explanation systems?

- How can we incorporate effectively learning systems?

- How is the notion of walk-up usability and interoperability achieved?

- How do we devise intelligent personalized e-learning environments through interfaces other than keyboard, mouse and screen?

- What metrics can we use to support evaluation and how can we embed these metrics into systems so that they can increasingly become more effective?

Despite this lengthy catalogue, it is also important to take account of the additional complexity that arises from taking the presence of disability in its various forms into account.

The design and creation of e-learning systems that meet the different stipulations must itself be a manageable task. There are important separations of concerns here. Designers of e-learning material should be free from the concerns regarding the underlying technological realities. We should encourage and support them in the task of developing imaginative and challenging activities that fundamentally address learning. The technological implications of this are profound. There are implications for entities such as design languages, appropriate software tools, systems and reusable components that support the designer (and yet do not compromise the privacy of learners).

While certain individuals learn best when they are alone, many people benefit from interaction and discussion within a supportive setting. In brief, they benefit from *group learning* and technology can provide support for this. In this context, it is important that individuals know the elements of such group learning and are able to create or become part of online communities that they find supportive and beneficial. We can capture the fundamental challenge facing education in this context through the following suggestions.

- Design interfaces for online communities and in the process establish e-learning as a credible alternative to face-to-face education.

- Provide interfaces with offline communities across the digital divide.

- Create learning organizations (two meanings).

With all these possibilities, the idea of quality (e-quality) needs addressing. More precisely, we need to re-interpret the concept of quality in the context of e-learning systems and environments.

Another important component of e-learning relates to assessment. In e-learning situations, individuals will require someone to assess and challenge them to ensure that they have mastered material. Various aspects of this span a whole range of complexity. These include multiple-choice questions that provide a regime in which assessment is relatively straightforward; on the other hand there is the time consuming and complex task of assessing essays or theses of, say 200 pages. Between these extremes, there are myriads of alternative possibilities with, for example, Bloom's taxonomy providing one set of possibilities

(see Krathwohl, Bloom and Masia, 1956). Providing online assessment for all these approaches is a considerable challenge.

This discussion of assessment need not be tied solely to e-learning. With the increase in numbers in higher education and the increased possibilities for plagiarism, there are serious problems in guaranteeing the reliability and the safety of assessments of student work generally. There are attendant dangers in assuming that every assessor will be equally vigilant and conscientious with every assessment. One way of addressing this is to use automatic methods. It is desirable that important advances are made in this arena, in part to relieve the colossal workload associated with much assessment, in part to open up possibilities of more assessment and more helpful feedback to students, but also to address the issue of consistency of treatment and fairness.

To conclude, it is worth drawing attention to three observations. First, in the United States (Jones *et al.*, 2003) a research-based grand challenge states, 'Provide a teacher for every learner'. Jones expresses this as

> Provide learning environments that approach the effectiveness of one teacher for every learner. Such systems, properly used, can produce a significantly better-educated populace by combining advances in learning science with advances in information technology.

In many ways, this captures the spirit of this challenge.

In a second observation, Bloom's taxonomy and variants of it highlight a hierarchy of skills that form part of educational accomplishment. Many of the current learning environments support the development or acquisition of only a very limited range of such skills, e.g. only the first two basic levels, namely *knowledge* and *comprehension*. There is, therefore, scope for developing learning environments to enable learners to support the development of skills at all taxonomic levels such as shown in Bloom's taxonomy; the other levels would address *application*, *analysis*, *synthesis* and *evaluation*.

In a third observation, Tony Hoare, Robin Milner and Wendy Hall (2004) present a further grand challenge, which is referred to as *memories for life.* This addresses the requirement to store all the lifetime experiences of an individual. It will typically contain a multitude of multimedia items, many of them possibly forgotten. The challenge is to extract or retrieve items as people stored them, potentially by a variety of methods and in a variety of user contexts.

2.6.4 Sub-challenges

Within this grand challenge 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.

- 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.

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

- 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 as well as online communities. 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.

- 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.

2.6.5 Measures of success

We can associate various metrics with the challenge and sub-challenges as follows:

- Uptake of and integration of e-learning into educational contexts and demonstrate the benefits that this brings.

- Contrast different learning environments and evaluate the rate at which students learn and the nature of that learning.

2.7 Pre-university issues

2.7.1 Situation

Universities recruit students from high schools and, in UK terminology, colleges of further education. They then aim to convert 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 increasingly 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 is likely to continue.

The vocational nature of important aspects 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 on high school staffs is very low. Teachers with a first speciality in another subject usually purvey computing. There is no suggestion that computing is inadequately 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 education qualification in computing itself.

There is 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 they know) the requirements of employers, and explicitly via a number of organizations and government agencies enumerating 'what they want' (e.g. Career Space, 2004).

2.7.2 Challenge

Given the situation described above, the computing challenge for this area is as follows:

> Rationalize the situation at the pre-university level and direct it 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.

2.7.3 Motivation

A number of surveys have occurred and these highlight significant issues. On being asked what they are expecting to study, entrants to university computing programmes usually are unaware of what they are going to study in a way that entrants to other disciplines are not. This can lead to disappointment and dissatisfaction. Surveys of dropouts reveal this as a major cause of high attrition.

Surveys of universities report that a primary source of information to entrants is teachers: computing or careers teachers. Hence, it is a contributory factor that those guiding students are often unaware of what higher education study of computing might involve. This problem extends to a lack of awareness of what careers in computing might be, or careers in industry and commerce that lead on from computing degrees. It also has to be recognized that the problems are often exacerbated by a lack of awareness among university staff of what is taking place within schools and colleges and their curricula. Generally the gender imbalance of university intake to computing programmes in certain countries is well known, as is the unevenness across many other population groups.

It seems *probable* that in ten years, the penetration of computers into society, coupled with government enthusiasm for ICT in all its guises, will lead to the emergence of a qualification among entrants that computing understands and respects. If this is the case, it implies a threshold point at which it becomes true, and has consequences for equipment in schools (or at home) to support the technologies used. This is not to suggest or to encourage the establishment of some advanced computing qualification as a prerequisite. Rather, the intention would be to identify an agreed suitable foundation that would be of value to those who did not wish to specialize in computing in higher education as much as to those that do.

Issues of resource, in any event, could easily have a backlash on any desired strategy. These issues are more difficult in the UK than in the USA[2], for instance, because of the traditional inflexibility of certain UK degree structures. The intention is not to decrease the amount of material in computing degrees, but to provide more space within them for breadth and depth.

There is no doubt that computing is currently unusual in having relatively few of its graduates active in secondary and tertiary education; this is at variance with a number of other disciplines (such as psychology, sociology etc.) which do not stipulate discipline prerequisites, but whose graduates frequently do choose to go into teaching. This means that pre-university computing students rarely benefit from the disciplinary input that comes from teachers and lecturers who are able to take them beyond the formal curriculum.

There is no doubt that programming is a major issue in the teaching of computing, often dominating the student experience in higher education (discussed in section 2.4). There is evidence beyond the merely anecdotal that programming (as understood within universities) is rarely encountered in schools and colleges of further education. In a five to ten year period, this state of affairs is likely to evolve. For example, already more than 40 per cent of homes in the UK have domestic computers – a proportion that will grow – and it seems plausible that the incidence of the desire to learn programming among the pre-university community will grow. Should this happen it would be desirable to influence its nature, while not intending to dictate *what* language is taught[3]. Circumstances would improve enormously if there was some basic agreement on that issue.

Fundamentally, one should view an education in computing as an extremely valuable commodity for the technological society of today. Apart from the knowledge and understanding, as well as the associated confidence and marketability, an education in computing imparts a range of valuable skills. These include: the ability to structure and organize information; to take disciplined approaches to problems and to complex situations; to recognize the power of life cycle models, of the concept of process and the associated improvement and enhancement possibilities; to recognize the power of abstraction including the different levels of such abstraction; to appreciate the elements of good design in all its forms; to recognize the need to carry out risk assessment in sensitive situations; to value and strive for innovation; to recognize the professional, legal and ethical issues associated with a profession.

2.7.4 Sub-challenges

Within this grand challenge 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.

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

- Communicate to teachers and careers advisors what university computing and computing careers are, and what computing degrees might lead to.

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

- 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.

- 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 the aims of widening participation.

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

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

2.   We note that in the USA the involvement of the ACM with school activity is considerable: note in particular the Java Engagement for Teacher Training project (see http://satchmo.cs.columbia.edu/jett/).
3.   Language wars within higher education are, in any event, rife.

2.7.5 Measures of success

We can associate various metrics with the challenge and sub-challenges as follows. In five to ten years, it might be reasonable to expect a prior computing qualification. This could include explicitly programming to some level and some understanding of 'algorithm'. So metrics might include:

- The incidence of the home-ownership of properly and easily programmable computers.

- The incidence of appropriately qualified computing graduates on the staff of high schools and colleges.

- The establishment and widespread acceptance of proper mechanisms to compensate for entrants with demonstrable prior qualification.

- Demonstrable understanding by university staff of school/college activity and syllabuses.

- The incidence of higher-education-based continuing professional development for computing teachers in schools/colleges.

- Average standards of those entering to study computing are no worse (and ideally better) than those entering to study related disciplines.

Ultimately success in improving such metrics should lead to:

- Dramatic improvements in the proportion of women and other under-represented sectors in intake.

- Improvements in retention rates so that, for instance, at most 10 per cent of university entrants drop out at the end of first year.

3 Addressing the challenges

In providing a set of challenges of this kind, it seems essential to have an eye to the existence or otherwise of mechanisms whereby we can address the challenges, monitor progress towards their resolution, and disseminate information. Hence, this 'final challenge' seeks to put in place such a mechanism.

3.1 Pending need

An initial observation is that the challenges presented in section 2 are important and require addressing by the computing community. Addressing them is vital work and we must ensure the relevant communities recognize them. These communities include university administrators and funding agencies. Generally, this implies changes in the status attributed to high quality activity in learning and teaching as well as clear support for developments of the kind implied by these challenges. If the challenges are to be addressed purposefully, then there needs to be an infrastructure in place that includes appropriate levels of resources and other support. There are considerable implications involved in these observations.

Given the important social, economic, and other impacts of computing, it must be important for the educational scene to be healthy and vibrant. This interpretation implies that the best students are attracted into the discipline and we must ensure they are enthused by the programmes of study that they follow and so remain within the discipline. It is our obligation to treat graduates as important and vital agents of technology transfer and of innovation.

In the USA, for instance, there is the National Science Foundation (NSF). It has divisions dealing with research. However, it also deals with high school, undergraduate and postgraduate teaching levels, education reform, and the development of national human resources. Its Division of Undergraduate Education has a mission that reads:

> To promote excellence in undergraduate science, technology, engineering,
> and mathematics education for all students.

The NSF seeks to accomplish this via a set of goals that includes providing leadership (encouraging innovation, identifying national priorities etc.), supporting novel curriculum development, preparing the workforce, and importantly fostering connections with other relevant disciplines.

3.2 A way forward

Different communities will need to decide how best to respond to these challenges. Much will depend on the relevance and the importance of the different challenges but also on existing institutions and organizations and existing infrastructures. The following suggestions are made in the context of the UK, but may find relevance on a wider front.

Within the UK there exists a variety of groups well positioned to address aspects of these challenges. The intention is not to usurp any one of these groups but to build on them and provide a coordinating impetus that ensures the challenges are addressed in a meaningful manner.

3.3 Meta-challenge

In light of the discussion described above, the computing meta-challenge to address the seven challenges presented in section 2 is as follows:

> Introduce a new *Computing Council* whose role is to coordinate activities
> that will lead to the challenges being addressed in an efficient and
> meaningful manner and in the process to further and promote the well-
> being of computing.

This is not the place to go into details such as constitution, remit, or funding. However, it is important to make certain stipulations.

- The *Computing Council* must not be a threat to any agency or group.

- Its existence should act as a focus for strengthening the position of computing.

- It should provide an ongoing sense of vision, purpose, and leadership and have an important role in setting national priorities.

- It should create initiatives that strengthen the discipline but also initiatives that foster relevant connections with other disciplines.

- It should provide a monitoring and dissemination role, and maintain a set of significant challenges.

- Importantly, it should take an international perspective on these and related matters.

An obvious initial step would be for all interested parties to come together.

There are some important observations to consider in relation to this. First, the *Computing Council* should not see a distinct separation between learning, teaching, and research. These activities are all intimately related with research informing learning and teaching, and the latter giving rise to important activity for the research community. An important observation made earlier is that appropriately crafted research-based grand challenges provide a sense of vision and purpose that serve to motivate and inspire potential students of computing. Second, the *Computing Council* should address the undergraduate, the master's, and research student levels. However, it must also address the high school level issues and the issues associated with the maintenance of a strong workforce.

4 Concluding comments

It would be a mistake to believe that the list of challenges presented here is complete. At best, one should view them as an initial attempt to draw attention to some significant issues the resolution of which 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 set out in this document are all independent of one another; indeed, there are interesting relationships between several of them.

The sincere hope is that the computing community will recognize these as expressing valid and important concerns. Individuals and/or groups of different kinds may choose to engage in the resolution of the problems (or part of the problem) and to disseminate the resulting insights.

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.

In some countries, quality regimes exist and these quality communities seek usually to bring about improvement. Such activity is often not disciplined-based, often absorbs vast amounts of valuable time, and has a focus on process and on bureaucratic issues that claim to provide evidence of quality. These often serve to mask important issues such as many of the issues identified within these challenges. Perhaps there is a message here.

Further reading

Career Space (2004) *Career Space*, http://www.career-space.com/project_desc/index.htm.

CRA (2003a) *Grand Research Challenges in Information Systems and Assurance*, published by Computer Research Associates, USA.

CRA (2003b) *Grand Research Challenges in Computer Science and Engineering*, published by Computer Research Associates, USA.

CUPM (2004) *Undergraduate programs and courses in the mathematical sciences: CUPM curriculum guide 2004*, a report produced by the Committee on the Undergraduate program in Mathematics of The Mathematical Association of America, published by The Mathematical Association of America.

Denning, P.J. (2003) Great principles of computing, published in *Communications of the ACM*, **46** (11), 15-20 November.

Fellows, S., Culver, R., Ruggieri, P. and Benson, W. (2002) *Instructional tools for promoting self-directed skills in freshmen*, FIE 2002, Boston, November.

Ganek, A.G. and Corbi, T.A. (2003) The dawning of the autonomic computing era, published in *IBM Systems Journal*, **42** (1), 5-18.

Hoare, T., Milner, R. and Hall, W. (2004) Report on grand challenges in research, *The Computer Journal*.

Jones, A. *et al.* (2003) *Grand Research Challenges in Information Systems*, published by Computing Research Associates, 2003.

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

Lyman, P. and Varian, H.R. (2000-3) http://www.sims.berkeley.edu/how-much-info and http://www.sims.-berkeley.edu/how-much-info-2003/.

McKeown, K., Clarke, L. and Stankovic, J. (organizers) (2003) CRA workshop on research related to national security: Report and recommendations, published in *Computing Research Review News*, **15** (4), September.

NSF (2003) *Revolutionizing Science and Engineering Through Cyberinfrastructure*, report of the National Science Foundation Blue-Ribbon Advisory Panel on Cyberinfrastructure, published by the National Science Foundation, January.

QAA (2000) *Academic Standards - Computing*, published by The UK Quality Assurance Agency for Higher Education, Southgate House Gloucester, England.

RAE (2004) *The challenges of complex IT projects*, a report of a working group from The Royal Academy of Engineering and The British Computer Society, published by The Royal Academy of Engineering, 29 Great St Peters Street, Westminster, London SW1P 3LW, April.

Smith, A. (2004) *Making mathematics count*, the report of Prof. Smith's inquiry into Post-14 Mathematics Education, published by the Stationery Office, London, February.

Appendix:
Summary of challenges

---

1. Perception of computing
Promote an improved and ultimately very positive public image of computing whereby the public gains respect for the field and the professionals who practise within it.

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

3. Competencies
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.

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

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

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

7. Pre-university issues
Rationalize the situation at the pre-university level and direct it towards the promotion of computing to would-be students of computing. Create for students a smooth transition from school to university by first enthusing and informing potential students and create a positive influence affecting pre-university computing.