Grand Challenges in Computing

Research

Edited by
Tony Hoare and Robin Milner

Organised by:

BCS THE BRITISH COMPUTER SOCIETY

CPHC Council of Professors and Heads of Computing

EPSRC Engineering and Physical Sciences Research Council

The IEE

northumbria UNIVERSITY

UK COMPUTING RESEARCH COMMITTEE

UNIVERSITY OF NEWCASTLE UPON TYNE

Grand Challenges in Computing –

Research

edited by Tony Hoare and Robin Milner

BCS
THE BRITISH COMPUTER SOCIETY

The British Computer Society
1 Sanford Street
Swindon, Wiltshire SN1 1HJ
UK
www.bcs.org

ISBN 1-902505-62-X
British Cataloguing in Publication Data.
A CIP catalogue record for this book is available at the British Library.

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

Introduction

Tony Hoare and Robin Milner

A Grand Challenge for research in science or engineering pursues a goal that is recognized one or two decades in advance; its achievement is a major milestone in the advance of knowledge or technology, celebrated not only by the researchers themselves but by the wider scientific community and the general public. It has no guarantee of success; an essential part of the Challenge is that it should define clear criteria for success or failure. A Challenge that failed was Hilbert's programme, formulated around 1900, for the foundations of mathematics; nevertheless its failure inspired previously unimagined advances in logic and mathematics by Church, Gödel and Turing. A Challenge that recently succeeded was the mapping of the human genome; it set directions and provided methodology for biology in the twenty-first century.

The computing discipline is distinguished by two factors: the breakneck progress of computing technology, and the world's voracious appetite for new applications. Against this background it is vitally important for the research community themselves to define the long-term aims and opportunities for their disciplines, independently of the push of technology and the short-term pull of demand.

With this purpose, and with stimulus from a similar initiative of the Computing Research Association in the USA, the recently constituted UK Computing Research Committee (UKCRC) has mounted a Grand Challenges Exercise. It has the support of the Engineering and Physical Sciences Research Council (EPSRC); the British Computer Society (BCS) and the Institution of Electrical Engineers (IEE) representing the computing profession; and the Conference of Professors and Heads of Computing (CPHC) representing the academic community. To ensure that the title Grand Challenge is reserved for truly far-sighted and ambitious goals, a first step in the Exercise was to agree a list of stringent criteria for judging the maturity of a Grand Challenge proposal. Not all the criteria have to be met, but it is remarkable how well they match the qualities and history of the Human Genome project.

Full details of the Exercise, including these criteria, can be found on the Grand Challenges website.[1] In the remainder of this introduction we summarize the history of the Exercise so far, list the current seven Challenges with some comments on the balance and synergy among them, and propose briefly the next steps to be undertaken.

History of the Grand Challenges Exercise

The UKCRC began the Grand Challenges Exercise in 2002 by appointing a Programme Committee to organize and conduct it, beginning with a Grand Challenges Workshop. The Committee members were Malcolm Atkinson, Alan Bundy, Jon Crowcroft, Tony Hoare (chair), John McDermid, Robin Milner, Johanna Moore, Tom Rodden and Martyn Thomas. An open call was issued for ideas that would meet the criteria. The Workshop was held in Edinburgh on November 2002, and discussed 109 submissions from the UK computing research community.

The details of the call for submissions, together with the planning and conduct of the Workshop and its follow-up, are all reported on the website. In summary, a set of possible topics for Grand Challenges was identified at the Workshop, and one or two champions for each were chosen to carry forward its development. A drafting phase followed the Workshop, leading to seven draft proposals. In January 2003 these were published on the website, each to be the subject of a public email Discussion, moderated by the champions and archived on the website. The Discussion was advertised to the research community via the CPHC mailing list, and is still open.

A principle of the Grand Challenges Exercise is that no submission from the community is or ever will be rejected by the Committee; the UKCRC has no desire to usurp the selective functions of the fund-granting agencies. Indeed the seven topics surveyed in this Report arose, via Panel discussions, from the original submissions to the Workshop, and all these submissions (except those withdrawn by authors) are still accessible on the website. Further submissions may be made at any time.

[1] http://www.nesc.ac.uk/esi/events/Grand_Challenges/

A second principle is that in formulating the Grand Challenge proposals no questions are raised about sources of funding. In this way the independence of purely scientific judgement is maintained.

A third principle is that the Exercise does not attribute higher importance to research aimed at a Grand Challenge than to exploratory research or to research aimed at more immediate goals; rather it seeks to unite suitable research directions, where appropriate, when they can be seen to contribute to long-term aspirations.

The next coordinated step in the Exercise was a Conference on Grand Challenges in Computing Research, held jointly in Newcastle from 29 to 31 March 2004, with a Conference on Grand Challenges for Computing Education. The stated aims of the Research Conference were:

- to encourage UK researchers in computing to articulate their views about long-term prospects and progress in their academic discipline;
- to discuss the possibility of speeding progress by broader collaboration, both nationally and with the international community;
- to facilitate the pursuit of more ambitious scientific and engineering goals;
- to work towards the definition of a Grand Challenge project, where this is an appropriate means to achieve the goals.

Again an open call was issued for submissions, either related to the existing seven Challenges or proposing new topics for Challenges. Some 50 submissions were received (remarkably, almost all linked to an existing Challenge) and these were again the subject of Panel discussions, chaired by a members of an extended Programme Committee; new members were Mike Denham, Ian Horrocks, David May and Vic Rayward-Smith. The two Conferences, sponsored by CPHC and the BCS, jointly attracted over 250 attendees.

The Grand Challenge proposals

As explained above, seven draft proposals were defined after the Workshop in 2002 and were the subject of further discussion and refinement, assisted by the website and especially by the Conference in 2004. Their titles are:

GC1 *In Vivo–in Silico* (iViS): the virtual worm, weed and bug;
GC2 Science for global ubiquitous computing;
GC3 Memories for life: managing information over a human lifetime;
GC4 Scalable ubiquitous computing systems;
GC5 The architecture of brain and mind;
GC6 Dependable systems evolution;
GC7 Journeys in non-classical computation.

We defer more detailed descriptions of these Challenges to the following sections; here we briefly assess them collectively. As will be seen, the goals of each Challenge are stated with precision and are distinct; yet on close analysis it emerges that many research topics will be shared among them. Each Challenge is a mountain peak; many routes to the peaks are in common. For example, abstract computational models (or virtual informatic machines) will arise in GC1, GC2, GC5 and GC6; they will represent the development of organisms in the first case, agent populations in the second, cerebral processes in the third, and language interpreters in the fourth. Software engineering principles are essential to both GC4 and GC6, though with different emphasis. GC7 will share biology-inspired computing with GC1 and GC5, and the non-sequential computing paradigm with GC2.

A related point is the extent to which exploratory research, while continuing for its own sake, will also receive stimulus from involvement in a Grand Challenge. For example, databases and information retrieval, including retrieval on aural and visual keys, will feature strongly in GC3; and hardware design will be crucial to the design of low-powered devices, whether sensors or informatic agents, in GC4.

These examples show that the Grand Challenges Exercise can exert a unifying influence upon computing research, even though conducted with a variety of motives and aspirations.

Next steps

It is intrinsic to the idea of a Grand Challenge that a large amount of work is needed to reach a point at which a ‘road map’ can be constructed. Such a road map consists of the definition of a sequence of projects by which the goals of the Challenge may be approached and achieved. Typically a road map will define initial projects in some detail, and later projects more conjecturally, depending on the outcome of earlier ones. The champion(s) for each Challenge have therefore each gathered a small team, which has typically organized workshops and discussion groups with such a road map as their goal. In some cases a road map is in sight; in other cases it is quite clear that exploratory projects are needed before even the definition of a road map is attempted. Such projects will typically have achievable goals, will bring together researchers in subdisciplines that do not normally interact, and will involve international collaboration.

The decision on the precise nature of the next steps lies firmly with the champion(s) of each Challenge and their own steering groups. The UKCRC expects to provide overall coordination where, and only where, appropriate. One form it may take is a conference every few years based on the Newcastle model; this provides a valuable way to keep the Grand Challenges Exercise in the sight of the whole community. A communal website will also serve this aim.

Finally, no Grand Challenge proposal is guaranteed to develop a road map and so mature into a Grand Challenge project. Even if it fails to do so, or succeeds in defining a road map but fails in achieving its ultimate goals, it will provide a valuable opportunity for discussion of long-term aspirations for computing research. It will attempt to look forward up to 15 years, identify the most valuable directions of research, and the most promising methods with which to pursue them. It may even identify milestones in the progress, at which the results of one line of research may become available for exploitation by another. Many of the scientists who have devoted their efforts to this exercise have welcomed the opportunity and encouragement to think deeply about such questions, and to take responsibility for the good health of future computing research.

The ensuing sections of this report present a résumé for each Challenge written by one or more of its champions, who are coordinating its development. The champions gratefully acknowledge the help of many collaborators in this Exercise. On the website readers can find a more detailed manifesto for each Challenge.

GC1: *In Vivo–in Silico* (iViS): the virtual worm, weed and bug

Ronan Sleep

We routinely use massively powerful computer simulations and visualizations to design aeroplanes, build bridges and predict the weather. With computer power and biological knowledge increasing daily, perhaps we can apply advanced computer simulation techniques to realize computer embodiments of living systems. This is the futuristic proposition of a research challenge proposed by UK computer scientists. The project, called *in Vivo–in Silico* (iViS), aims to produce fully detailed, accurate and predictive computer embodiments of plants, animals and unicellular organisms.

Initially the aims will be restricted to simple and much studied life forms such as the nematode worm, the humble weed Arabidopsis and single-cell organisms such as streptomyces and bakers' yeast: hence the subtitle 'the virtual worm, weed and bug'. These model organisms, apparently so different, have much in common:

> *As we trace the increase of complexity from single cell creatures, through small animals like worms and flies … evolution is not so much adding new genes performing wholly new functions – what it's chiefly doing is to increase the variety and subtlety of genes that control other genes.*[2]

Further, the human and worm have a common ancestor from which we jointly inherit many similar genes. An example is the recent discovery that there is a gene in the worm that is similar to the human breast and ovarian cancer gene BRCA1 (Boulton *et al.*, *Current Biology*, 14, 1, pp. 33–39). So there is considerable hope that studies of the simpler life forms will have real relevance to humans.

Possible benefits of iViS include an understanding of regeneration processes in plants and animals, with potentially dramatic implications for disease and accident victims. But iViS may also lead to revolutionary ways of realizing complex systems: instead of designing and programming such systems in excruciating detail, perhaps we can just *grow* them from compact initial descriptions in a suitable medium. We know it's possible, because that's exactly what nature does with the worm, the weed and the bug.

The vision

iViS offers a powerful vision in which future life scientists can take virtual reality fly-through tours of a plant, animal or colony of cells, studying what is happening at scales ranging from a whole life form to what goes on inside an individual cell, and stretching or shrinking time. Filters control what is seen by the observer, allowing concentration on specific aspects such as cell division, motility or chemical potential.

This is an attractive way of exploring our knowledge about a life form. But iViS may offer more: with sufficient effort, it might be possible to raise the faithfulness of the underlying model to the point where it becomes predictive as well as descriptive. If this happens, it will become possible to perform meaningful observations and experiments *in silico*. And we want to cover a wide range of phenomena: specifically, we include development from an initial fertilized cell to a full adult, cell function and interaction, motility and sensory behaviour, including interactions with other life forms. Virtual experiments (e.g. moving a virtual cell during development) should lead to the same outcomes as real life.

iViS and the life science data mountain

Computers are vital to the life sciences: they record, process, visualize and automatically distribute data, and even design and run experiments. They analyse the results in terms of large statistical and other

[2] http://www.sanger.ac.uk/HGP/publication2001/facts.shtml

mathematical models of biological processes.

But what do all the numbers, graphs and spaghetti-like diagrams that emerge from the latest experiments all mean, and what can we do with this data? Making it all fit together into a coherent and useful picture presents a major challenge – many biologists would say the major challenge – facing the life sciences.

This problem is now so pressing that the UK's Biotechnology and Biological Science Research Council (BBSRC) is establishing a number of Centres for Integrative Systems Biology. These Centres will need the vision, breadth of intellectual leadership and research resources to integrate traditionally separate disciplines in a programme of international quality research in quantitative and predictive systems biology. iViS offers a challenging focus of attention for such centres.

iViS as a driver for global knowledge integration

Part of the answer to the data mountain may lie in the way in which the web is revolutionizing our approach to knowledge organization. The web is already a vital window on the world for scientists wishing to remain up to date. Groups of laboratories that previously worked at arm's length and communicated infrequently only via journals and the conference circuit now converse via the web within seconds, swapping massive datasets to compare results. Scientists have begun to exploit the web in its own right by establishing global Virtual Knowledge Repositories to share data, theories and models.

A particularly relevant example is Physiome,[3] which supports

> *the databasing of physiological, pharmacological and pathological information on humans and other organisms and integration through computational modelling. 'Models' include everything from diagrammatic schema, suggesting relationships among elements composing a system, to fully quantitative, computational models describing the behaviour of physiological systems and an organism's response to environmental change.*

Virtual Knowledge Repositories like Physiome will help reduce the proliferation of models and theories that explain parts of the global mountain of life science data. But this in turn will create a deeper challenge: instead of fitting raw data pieces together, we will be faced with the problem of making the models fit into a consistent larger model. Sometimes this will be easy, for example when there is a simple input–output relationship between subsystems. More often – perhaps the rule rather than the exception – combining two models will show unexpected interactions inconsistent with *in vivo* data.

Part of the problem is that mathematical models deal in numbers, devoid of meaning. The latest evolution of web technology – the semantic web – may help fix this. There is now provision for the web to enhance raw data with additional information called metadata. This can tell the recipient what the data means, how it is represented and the way in which it was generated. Models, which often come in the form of a computer program, can be tagged with metadata describing their assumptions and use: effectively an inbuilt instruction manual.

There are already over 40 metadata dictionaries[4] (called ontologies) for the life sciences. So the drive and energy to create bio-ontologies is already very active. But there is not the same drive to draw them together into a unified whole. The iViS Challenge provides just such drive because the *in silico* modelling of a *complete* life form will require harmonious working across all relevant ontology boundaries.

Even if we can build a simulation of a life form that successfully integrates all known data, we need to take care in choosing our models. If they follow all the raw data too closely, the models may lack any predictive power. For example, we can always find a polynomial of degree $(n-1)$ to fit n data points exactly. This is a strong reason for complementing data-driven modelling work on iViS with more abstract top-down approaches. If we take care there will be at least some domains which succumb to iViS's whole life form modelling approach: developmental biology looks a good bet.

[3] http://www.physiome.org/
[4] http://obo.sourceforge.net/

Meeting the Challenge: iViS research themes

The obvious targets for iViS models are the organisms selected for special attention by biologists for over a century. These range from single-cell life forms such as yeast or streptomyces, through model plants such as Arabidopsis and maize, to creatures such as the nematode worm, the fruit fly and the squid.

But how can we ‘breathe life into data’ via computer simulation? This is not simply a question of computing speed or memory, but how to represent the mass of known data as a set of interacting computational processes. We can get computers to simulate massively complex aircraft or bridges, but getting them to ‘grow a worm, weed or bug’ is significantly beyond the current state of the art.

Nevertheless, it may not be impossible if we build determinedly on the considerable body of work underway to explore ways of organizing life science data. One example is the Edinburgh Mouse Atlas Project: [5]

> *The EMAP Atlas is a digital atlas of mouse embryonic development. It is based on the definitive books of mouse embryonic development ... yet extends these studies by creating a series of interactive three-dimensional computer models of mouse embryos at successive stages of development with defined anatomical domains linked to a stage-by-stage ontology of anatomical names.*

It can be expected that growing numbers of life science virtual knowledge centres will follow EMAP in adopting some form of spatio-temporal framework. The role of iViS is to expand this vision to a dynamic three-dimensional working model, initially targeting much simpler life forms.

There are a number of research strands in the computing sciences needed to support the aspirations of iViS. We might bundle them under the heading ‘Computational Models and Scalable Architectures for *in silico* Life Sciences’. Some strands will work bottom-up, paying great attention to biological data. Other strands will work top-down, studying minimal abstractions capable of generating the phenomena exhibited *in vivo*. Many will work ‘middle-out’, balancing the desire to be simple, elegant and general with the desire to be faithful to the data.

Key to success will be the development of a new breed of computer languages for representing and manipulating biological data in a meaningful way, and using it to drive a realistic, highly detailed simulation, which can be explored using advanced interfaces.

Groups of computer scientists are already exploring new languages, architectures, and system design and analysis tools for the life sciences. Luca Cardelli of Microsoft[6] gives a good picture of this work. Cardelli and others are tackling the complexities of life science head on, developing industrial-quality models aimed at handling the masses of detail in a living system, and – of critical importance if the results of iViS are to be trusted – validating the resulting models.

The impact of such work on the life sciences could be as dramatic as the discovery of structured programming was for computing in the late 1960s. Prior to structured programming, even short programs looked like a mass of spaghetti, just as much of our knowledge of living systems does now. If biological analogues of the compositional primitives of structured programming (sequencing, alternation and repetition) could be discovered, the prospects for integrated systems biology would be very bright indeed.

Such direct attacks on the complexity of biological detail are complemented by more abstract top-down approaches. These begin by asking what sort of computational systems have emergent lifelike properties. Such abstract models can be useful when viewing some particular aspect of a plant, animal or cell. For example, Prusinkiewicz[7] has almost created a new art form for constructing good-looking pictures of plant growth from remarkably simple abstract models called L-systems. These models capture only a tiny part of the truth, but iViS may need such simplifying ideas to help structure the great mass of detail.

What about raw computer power and advances in haptic and other interface technologies? Certainly

[5] http://genex.hgu.mrc.ac.uk/
[6] http://www.luca.demon.co.uk/
[7] http://www.cpsc.ucalgary.ca/Research/bmv

they will be needed, but some will emerge anyway from the computer industry without special prompting. The critical problem is to get the underlying computational models right. Once we have these, we can begin, as it were, serious work on designing the building and sorting out of exactly which contemporary technologies we should use to actually construct an iViS prototype.

Demonstrators and outline road map

iViS progress will surface as a number of key demonstrators, culminating in whole life form models covering a wide range of phenomena. Intermediate demonstrators will cover a narrower range. Modelling the development of form during development is one example, motivated by the following quote:

> *Perhaps no area of embryology is so poorly understood, yet so fascinating, as how the embryo develops form. Certainly the efforts in understanding gene regulation have occupied embryologists, and it has always been an assumption that once we understand what building blocks are made, we will be able to attack the question of how they are used. Mutations and gene manipulations have given insight into what components are employed for morphogenesis, but surely this is one example where we need to use dynamic imaging to assess how cells behave, and what components are interacting to drive cell movements and shape changes. (*Scott E. Fraser and Richard M. Harland, *Cell*, 100, pp. 41–55, 7 January 2000)

A speculative timeline is:

Within 5 years: Early results on developmental phenomena in plants and animals, and first unicellular demonstrations. Prototype modelling frameworks and validation methodologies.

Within 10 years: First prediction of a textbook result from an assembly of component models; models of meristem growth; models of simple animal development; reliable unicellular models; mature iViS modelling environments.

2017, 100 years after the publication of D'Arcy Thompson's paper 'On Growth and Form': First substantial demonstration of iViS whole life model.

Within 20 years: iViS models in common use.

What happens next?

A website[8] for the iViS Grand Challenge has been established, and funding is being sought to help build the community of cross-disciplinary scientists needed to elaborate and vigorously address this very exciting challenge.

[8]http://www.cmp.uea.ac.uk/ivis

GC2 Science for global ubiquitous computing

Marta Kwiatkowska and Vladimiro Sassone

How many computers will you be using, wearing, or have installed in your body in 2020? How many other computers will they be talking to? What will they be saying about you, doing for you or to you? By that time computers will be ubiquitous and globally connected. It is better not to count them individually, but to regard them collectively as a single Global Ubiquitous Computer (GUC). Who will then program the GUC and how? Shall we be in control of it or even understand it?

Let us imagine what life will be like in 2020. Smart homes will be equipped with autonomous, self-aware sensors, which will be capable of gathering information about their physical and digital environment in order to recognize events, identify threats and take appropriate actions. Intelligent spaces will be created by positioning sensors in the environment, all wirelessly connected and capable of monitoring working conditions and access to restricted parts of the buildings by automatically sensing radio tags worn on clothing. Medical treatment will be personalized, based on our genetic make-up and factors such as age, and even delivered directly to our bodies by wearable devices.

Does this seem too futuristic? Not so, if one considers the degree to which our lives are today dependent on the internet, with access to grid and web services, and involve electronic devices that are ever smaller, wireless and mobile. Already there are hundreds of embedded processors in modern cars, wireless 'motes' are deployed in environmental monitoring and industrial control, intelligent buildings are a commercial reality and first successes have been reported with 'medibots' (nanoscale DNA computers capable of fighting cancer cells in a test tube).

What are the essential features of an infrastructure to support such a scenario? Firstly, the internet already enables *global connectivity*, by means of wired, radio and satellite communications; this implies the huge *scale* and *complexity* of the system, its highly *distributed* nature, *mobility* and continued *evolution*. Secondly, each node on the network, either sensor or device, is capable of *computation*, *communication* and *information processing*, as it shrinks in size to the microscale, possibly nanoscale. Thirdly, the devices are increasingly *self-aware*, space and location conscious, able to interact with the surrounding environment and exhibiting introspective behaviour. The size of the network, and the complexity of the interactions within it, demand that they be capable of *cooperation*, *self-organization*, *self-diagnosis* and *self-repair*. Finally, *trust*, *privacy*, *security* and *dependability* must be assured, as the cost of malfunction or breach of contract can be very high.

The challenges this raises for both computer engineers and software designers are enormous. The risks in the GUC are great: confidential medical records must never be leaked out to unauthorized parties, drive-by-wire sensors must respond timely, and 'medibots' must be dispatched only if it can be guaranteed they work correctly. This is in sharp contrast with the frequent failures of current computer systems, which are orders of magnitude simpler than those we envisage for the GUC.

So, how can we engineer a better GUC? Just as engineers rely on *sciences* of the physical world when building bridges, and use toolkits and theories from Physics and Mathematics to model and evaluate their designs, we need to develop a Science for Global Ubiquitous Computing, a fundamental theory describing ubiquitous computational phenomena. This will involve *conceptual*, *mathematical* and *software tools* to inform and support the design process for ubiquitous systems, via models, programming languages, protocols, analysis techniques, verification tools and simulation. For example, the 'medibots' have been modelled using automata theory, temporal logic has been used to specify the security of communication protocols and verification tools to detect their flaws, the pi-calculus models *ad hoc* networks, and simulation is often used to analyse system performance. But we shall need many more new concepts and tools. For instance, the theory must provide logics for trust and resource, and tools for the analysis of crypto-protocols and routing algorithms. It must also span several abstraction levels to provide a flexible framework to harness complexity.

How can science make a difference?

Is there a challenge for science at all? The scenarios we envision are, as illustrated above, typical futuristic, technological scenarios. Why cannot they be conquered as purely engineering, 'hands-on' tasks? The difficulty is that, without rigorous analysis of all the possible interactions between system components in the design phase, it is all too easy to build systems that exhibit erroneous behaviour. The recent discovery of security flaws in the 802.11 and Bluetooth® protocols can serve as a pertinent example.

Of course, new technologies are bound to be experimental and, therefore, they run an intrinsic risk of being flawed. There is nothing wrong with that, and it would be naive for us to advocate a future in which progress and market forces wait for scientists to give the green light. On the other hand, the scientific analysis of technological innovations allows us to understand solutions and their limitations, and is irreplaceable in a field like the GUC, which aims at applications with the potential to radically change the way we live. Enormous wealth, both in moral and physical terms, is currently entrusted to the internet, and this can hardly be expected to change in the GUC future.

Such analysis is the purpose of our challenge. It will underpin the design and engineering that is the focus of our sister challenge Scalable Ubiquitous Computing Systems. Science is not design itself; the two are distinct, but sisters. They proceed with different timescales, tools, principles and milestones. But they only make sense as Grand Challenges when coupled together. We describe below two groups of relevant issues where the scientific, foundational approach will make a difference. The first group is concerned with models.

System and software architectures: We need models that inform the design of large software-intensive systems formed by *ad hoc* networks of heterogeneous components. The models must support evolution, adaptive behaviour, loose coupling, autonomy, context-awareness, learning, security and more.

Mobility, self- and context-awareness: We need calculuses and logics to formalize these notions. For example, how will systems attach a semantic meaning to information received from the contexts met while roaming the global network, and how will they validate such information?

Discrete/continuous models: We need to extend the current limited logics and formalisms to provide predictive theories for hybrid systems (e.g. sensor networks) that feature continuous inputs, for example position and temperature, as well as discrete ones, for example room number.

Stochastic models: We need to adapt the current stochastic models (increasingly important for network protocols) to provide compositional probabilistic analysis of the systems and subsystems that make up the GUC.

Cognition and interaction models: We need to model the cognitive aspects of the GUC, whereby software devices learn during their lifetime, and learn how best to interact with humans inside a smart home.

Knowledge, trust, security and privacy: Migrating devices will acquire information on which they will base action, including interaction with potentially dangerous environments. We need to build models for the acquisition, distribution, management and sharing of such knowledge, and in particular how trust may be based upon it.

The second group of issues is concerned with languages and tools.

Programming language design: We need to isolate language features appropriate to the GUC, especially for complex disciplines of interaction. New data structures, for example based on XML, are of equal importance.

Ubiquitous data: We need to understand how best to embed semi-structured data in programming languages and applications. This entails new type systems (for safety) and theories to underpin the notions of 'certified origin' and 'relevance' of data items.

Protocol design: We need to develop decentralized protocols for information exchange in *ad hoc* wireless and sensor networks. These will likely contain elements of randomization to break the symmetry between the nodes, achieve scalability and improve performance under changing conditions.

Algorithms for coordination, cooperation and autonomy: Interactions in the GUC invite a vision of autonomous, mistrustful, selfish components, with no notion of common purpose and a rather faint notion of common welfare, running in some sort of equilibrium. We need to discover the algorithms to underpin all this.

Software technology and design support tools: We need to upgrade and advance the models which underlie our current design technologies. They will involve new type systems, new forms of static analysis (including topographical analysis) and new validation/testing suites equipped for the GUC.

Verification techniques and technology: A marked success of computation theory has been the acceptance in industry of verification software tools, chiefly model checkers and automated theorem provers. We need to further advance research in this field so that every safety or business critical system can be fully verified before deployment.

Related activities

There are already a number of international and UK activities and projects that can provide support for the research outlined above. The EPSRC Network UK-UbiNet provides a forum for a broad spectrum of activities, from appliance design, through user experiences, to middleware and theory. It has already held two workshops and plans a summer school in the near future. The EU-FET Global Computing initiative is a Framework Sixth programme focusing on the foundations of the GUC. In the UK, the Next Wave initiative of the DTI focuses on products and technology transfer for ubiquitous computing; Mobile Bristol[9] aims at developing user experiences with digital devices. The Equator project[10] is an Interdisciplinary Research Collaboration which focuses on interleaving physical and digital interaction. In the USA, Berkeley's Smart Dust originated the concept of 'motes', cubic millimetre miniature computing devices, which are now being developed at Intel[11] and Crossbow. In a recently founded UK–US partnership, the Cambridge-MIT Institute (CMI), continues the research begun by project Oxygen.[12]

Science for Global Ubiquitous Computing is related to several other Grand Challenges discussed at the Conference in Newcastle (March 2004). The closest is Scalable Ubiquitous Computing Systems, a true sister challenge which shares our domain of interest. Indeed, the two held joint meetings in Newcastle and intend to collaborate in future, thus realizing what Christopher Strachey once famously said:

> *It has long been my personal view that the separation of practical and theoretical work is artificial and injurious. Much of the practical work done in computing ... is unsound and clumsy because the people who do it have not any clear understanding of the fundamental design principles of their work. Most of the abstract mathematical and theoretical work is sterile because it has no point of contact with real computing.*

There are also strong and useful relationships with at least three of the other challenges: Dependable Systems Evolution, since the Global Ubiquitous Computer is highly dynamic and dependability is an essential requirement; *In vivo–In silico*, which shares with our challenge the aim of predictive modelling for system dynamics and composition; and Memories for Life which aims to develop devices and recordings to become part of the physical GUC. We also note that the new National Science Foundation's Science of Design programme[13] for internet and large-scale distributed systems, the infrastructure that underlies the GUC, addresses issues very similar to ours.

First steps and ultimate goals

We would like to begin by defining a 'road map': a conjectured path or group of paths leading to the goals. The discussions at the Newcastle Conference led us to the conclusion that it is too early for this,

[9]http://www.mobilebristol.co.uk/flash.html
[10]http://machen.mrl.nott.ac.uk/home.html
[11]http://www.intel.com/research/exploratory/motes.htm
[12]http://oxygen.lcs.mit.edu
[13]http://www.nsf.gov/pubs/2004/nsf04552/nsf04552.htm

largely because the GUC is an artefact that is still developing, and in unpredictable directions. Instead, we decided to focus on initial collaborative experiments, which may be called foothill projects, to help us establish a case for such a road map. The projects arising from the panel discussions include:

Assuring privacy, authentication and identity of medical records: An exemplar that enables anytime/anywhere accessibility to medical records to doctors or patients in a provably secure manner will be developed. It will be based on sound models of trust- and privacy-ensuring technologies, and will consider human and digital communications of various kinds. Collaboration between security experts, researchers in human–computer interfaces, and the medical and legal professions will be needed.

Rigorous process models for web services: We are increasingly dependent on web services for information, but are often frustrated with their unreliability and untimely responses. A collaboration between researchers in process calculuses and middleware will address rigorous foundations of web services, and will identify key coordination primitives for the GUC software architecture.

Communications infrastructure protocols: As the GUC continues to evolve, we will see more and more reliance on sensor and wireless networks in addition to the conventional internet (TCP/IP) protocols. New communication and routing protocols will have to be designed, implemented, verified and analysed. This project will involve networking experts, middleware developers and verification researchers.

We aim to devise a *science* of the GUC: a set of concepts and theories that underpin and play a driving role in the design and engineering of the GUC and its components, which will assist in the modelling, analysis, diagnosis, evaluation and validation of the design. Our goals are:

> *to develop a coherent informatic science whose concepts, calculuses, theories and automated tools allow descriptive and predictive analysis of the GUC at many levels of abstraction; and that every system and software construction, including languages, for the GUC shall employ only these concepts and calculuses, and be analysed and justified by these theories and tools.*

We envisage a 15-year time frame for this challenge to come to fruition, and much of the research will require global effort. It is, of course, possible that our challenge will never be entirely met. However, even partial successes will yield strong advantages. And it is ultimately the responsibility of computing research to place the GUC on as rigorous a basis as possible.

GC3 Memories for life: managing information over a human lifetime

Andrew Fitzgibbon and Ehud Reiter

When computers first appeared, they were used to help people perform numerical calculations and clerical tasks. More recently they have also been used to help people communicate. But there is another potential use of computers which is now emerging, namely as a tool to help people store, search and interpret their 'memories', by which we mean digital personal information such as emails, digital photographs and internet telephone calls. People are capturing and storing an ever-increasing amount of such memories, with new types of information constantly being added such as GPS location data, heart-rate monitors, TV viewing logs etc. The vision of Memories for Life is to help people manage and use their digital memories across their entire lifetime.

Memories for Life is a challenge for computer science because only computer science can solve the scientific problems of searching, storing, securing and even (selectively) 'forgetting' this digital data. It is a Grand Challenge because existing solutions, based on manually annotating every document, image and sound, will not scale to the volumes of data being collected by and about people today. It is also a challenge which is important to society as we can see for example by the frequent mention of 'information overload'; society badly needs tools that help people manage and use information better.

Memories for Life is not a single scientific challenge, but rather a set of scientific challenges which are associated with the technological goal of helping people manage and use their electronic 'memories'. Success at meeting the challenge will be measured by successful applications as well as theoretical advances.

Vision: applications

There are numerous applications of Memories for Life. In the next 5–10 years, we expect that the most progress may be made in systems that help people retrieve and organize their memories. For example, such a system might help a person find all memories, regardless of type, about his or her holiday in Germany two years ago; and also help organize these memories by time, location or topic.

In the longer term, perhaps 10 or 20 years in the future, Memories for Life systems would interpret as well as retrieve memories. Such systems might be incorporated into personal companions, which use information extracted from memories to help people. For example, a Senior Companion might help an elderly person in various ways, including reminding him, or contacting his aide, when he deviates from his normal schedule as observed by the companion on previous days; helping him organize his digital photos, videos etc. for presentation to grandchildren and other relatives; and perhaps even serving as an archive of the person's life which persists after his death and can be consulted by friends and family to refresh their memory of the departed. A Medical Companion might use memory data to detect potential medical problems and otherwise help doctors. For example, such a system might detect potential autism in a young child, thus enabling early and hence more effective treatment. Other companions could include Learning Companions which present lessons in the context of the learner's memories and experiences, using language the learner is comfortable with; and Administrative Companions which process routine paperwork by observing how similar paperwork was processed in the past.

Scientific challenges

In this section we list a few of the many scientific challenges involved in Memories for Life. We do not discuss here scientific challenges related to ubiquitous computing. While these are very important for Memories for Life, they have been presented in two other Grand Challenges: Science for Global Ubiquitous Computing and Scalable Ubiquitous Computing Systems.

Security and privacy

How should privacy be protected, especially when an individual is present in someone else's memory? For example, if Joe has a digital picture of Mary and he holding hands, what rights does Mary have over the use and distribution of this picture? And how can these rights be enforced if Joe physically controls, or has copied, the image file, especially if Mary subsequently wishes to change the rights she initially gave to Joe (for example, if she and Joe were married but got divorced). What rights does the legal system have to access memories, and can the accuracy of digital data be confirmed? There are a whole host of both technical and public-policy security issues which need to be addressed before Memories for Life can become a reality. It may be technically easier to tackle these issues if memories are stored in shared central servers, but many people are likely to resist this, so we need solutions that work when data is stored on personal PCs as well as on servers.

In our experience, whenever Memories for Life is discussed with the general public or the non-scientific media, privacy and security issues are the ones most frequently raised. We need solutions that not only work technically, but that also can be explained and justified to the general public so that they have confidence that their data will indeed be private.

Data and databases

How should a diverse range of memories be stored, indexed and searched in an integrated and seamless fashion? How can we ensure that data is still accessible in 50 years time, despite inevitable changes in software, hardware and formats? How do we check that data is accurate, and how do we fix mistakes? What annotations and other contextual information should be stored along with the actual memories, and can these be acquired automatically or do they need to be manually entered? Should we store just the raw data, or also interpretations?

Indeed, what exactly do we mean by 'personal data'? For example, to support applications such as medical diagnosis, a person's memories should contain some kind of medical record. But a full medical record contains information that a patient may not be aware of and indeed may not be authorized to see (for example, some psychiatric assessments). Should such information be included in a person's digital memory?

Clearly what data should be stored and how it should be organized depend on the applications being supported. The central scientific challenge is to understand what data is needed for what types of applications, and to devise methods of acquiring this data accurately and with minimal additional cost.

Information retrieval

How can we easily and effectively retrieve useful information from the sort of multimedia data we expect to find in an archive of personal digital data? How can queries that require sophisticated interpretation be handled, such as 'find a picture of me playing with Peter when he was a toddler'? The web is a rich collection of digital data in various formats which is not completely dissimilar to the sorts of data we expect to find in personal memories.

Information retrieval, for example Google™, has been one of the keys to making the web useful and exploitable, and undoubtedly information retrieval will play a similar role in making collections of personal memories useful.

Artificial intelligence

How can information from a diverse assortment of memories be interpreted, combined and reasoned with? These questions bring in an enormous range of artificial intelligence topics, including computer vision (e.g. figuring out what people and objects are in a digital photo); speech recognition and natural language processing (e.g. understanding the topic of a recorded internet telephone call); knowledge representation and reasoning (e.g. bringing together information extracted from diverse media types into an integrated model and filling in 'holes' where data is unavailable); machine learning and data mining (e.g. building general models of an individual's behaviour based on specific incidents); and so forth. Indeed, almost every branch of artificial intelligence is relevant to Memories for Life.

Machine learning

The key to medical diagnosis is gaining enough data about the patient. For example, we have for decades been able to build medical diagnosis systems that outperform most human doctors, if they have access to sufficient data about the patient. Unfortunately the data needed is almost never available because it includes information that currently must be entered manually, such as the patient's visual appearance, behaviour, environment and history. Even a GP with a good relationship with a patient may have the equivalent of no more than one hour's conversation with the patient over 10 years, and will be far from knowing everything that might help a given diagnosis. If the physician were to be allowed (temporary and confidential) access to the patient's stored photos, emails and even conversations, he or she would be able to quickly search for lifestyle hints which might aid patient care – 'I see you holidayed in Australia last September, at which time there was an outbreak of Legionnaires' disease'. Some of this technology is available now, but the provision of powerful tools to query the data, while protecting privacy, requires significant research effort.

Human–computer interaction

How should people interact with (view, update, delete, annotate) memories and associated knowledge? Can we provide a wide variety of tools suitable for people of very different backgrounds and preferences (virtual reality? smart paper? speech dialogue?). Memories for Life is intended to benefit everyone, not just computer-philes, and this means we need interfaces that work for children who hate reading and love computer games; adults, from socially deprived areas, who can barely read or add, and have almost no self-confidence; elderly people with cognitive impairments; etc, etc.

Also, can we use information extracted from memories to help people use computer systems? One of the visions of human–computer interaction is to personalize interfaces to the skills, preferences and abilities of individual users, but this has been held back by the difficulty of building accurate models of individuals. Can accurate user models be built by analysing an individual's memories?

Digital memories and real memories

The above challenges focus on the computational problems of managing and using computer archives of personal digital data. But, of course, 'memory' in the broader sense is fundamental to human cognition, and indeed information storage (that is, memory) is fundamental to the functioning of society. Are there similarities between digital memories and these other types of memories? Such questions raise issues for neuroscientists, psychologists, sociologists and philosophers, as well as for computer scientists.

For example, one of the main puzzles of the neuroscience of human memory is forgetting. What does forgetting mean: is data gone or just hard to find? How and why does it happen etc? It will be extremely interesting to see in what manner computer-based Memories for Life systems 'forget', and if there are any similarities between this and human forgetting.

Past, current and future activities

The first mention of computer 'memories' in our sense was probably Vannevar Bush's Memex in 1945. This was extremely speculative and indeed Memories for Life remained mostly speculation for the next 50 years. Recently, though, the topic has moved from science fiction to the research agenda because of advances in hardware (processing power, disk capacity, digital cameras and other sensors) coupled with a clear interest among many people in storing large amounts of personal data on their computers. Also important has been the increasing importance of data-based techniques in many fields of computer science. In particular, while 10 years ago many of the above challenges seemed to require solving the complete artificial intelligence problem, now we know that many problems that were once thought to require deep reasoning can in fact be solved by using machine learning techniques on large data sets.

There are various projects underway which are related to Memories for Life, of which the best known is Microsoft's MyLifeBits.[14] MyLifeBits focuses on collecting, organizing and using the memories of one individual, Gordon Bell. Memory for Life concepts are also emerging in the blog community, for example Nokia's Lifeblog.[15] Within the UK, Memories for Life has been discussed by the Foresight Cognitive Systems project.

For the near future, we have proposed to EPSRC that it supports a Network of Excellence on Memories for Life. If EPSRC agrees, this network would support road-mapping workshops, which would bring together interested people to attempt to define concrete short-, medium- and long-term research objectives. The network would also investigate whether it is sensible and possible to create some

[14]http://research.microsoft.com/barc/mediapresence/MyLifeBits.aspx
[15]http:www.nokia.com/lifeblog

example memory corpora, that is example datasets of personal digital information, which could be used by the research community. The network would also organize discussions with interested scientists from other disciplines, especially people studying other types of memory.

Conclusion

We believe that the accumulation of personal digital memories is inevitable; people are going to be storing their emails, digital photos etc, regardless of what computer scientists do. The general challenge for computer science is to develop ideas and techniques that help people get the maximum benefit from these memories, while at the same time giving people complete control over memories so as to preserve their privacy.

GC4 Scalable ubiquitous computing systems

Jon Crowcroft

We are all acquainted with the personal computer. We are less aware of the burgeoning numbers of invisible computers around us, embedded in the fabric of our homes, shops, vehicles and even farms. They help us command, control, communicate, do business, travel and entertain ourselves, and these invisible computers are far more numerous than their desktop cousins.

The visible face of computing, the ubiquitous PC, is nowadays generally networked. In contrast, embedded computing systems have largely been stand-alone (e.g. in the brakes of a car), replacing analogue control systems. Increasingly, however, they will become integrated into a whole.

This will lead to a challenge in the form of system complexity. For as the systems become integrated they will also become even more numerous, smaller and more deeply embedded, for example in our clothes and even our bodies. Faced with designing such populous systems, computing engineers can conquer complexity only by evolving *scalable design principles*, applicable at each level of population magnitude. The core of this Challenge is therefore to abstract out these engineering design principles, via a process of 'build and learn'.

A ubiquitous system scenario

Many different scenarios can be chosen to illustrate how the engineering problem will escalate with different stages of development; for example, an instrumented home environment responding to occupants' needs or a driverless travel network.

We choose a medical scenario. Envisage that the entire population of the UK is to be permanently instrumented for systematic monitoring of metrics such as heartbeat, skin conductivity, blood sugar etc. and that this data is to be logged in a secure way via wireless communication.

In the short term this would be done mainly for people at risk, allowing medical staff to take prompt remedial action; in the longer term it could be applied to all, with obvious benefits to medical research. In a second stage intervention might be introduced, at first via remote diagnostics and human advice. Later the intervention might be via autonomic responses; for example triggering defibrillators for cardiac patients undergoing unattended attack, or even replacing faulty natural autonomic processes with a distributed artificial system having limited intelligence. A third stage might be to synchronize information, making sure that it is both up to date and consistent, between patient, hospital and an arriving emergency service.

At every stage of development, we should be able to determine the level of risk of using a system, and in the event of failure it should be possible to provide an automatically derived explanation. This would explain complex actions that might otherwise seem mysterious. For example, our autonomic cardiologist might excuse its behaviour with the explanation that: 'the defibrillation was run despite an *x* per cent risk that a heart failure had not occurred, because ...'.

Rationale for the Challenge

There is a popular view that the technology underlying the internet, the web and the PC (hardware and operating systems) has solved the problems of system design for large-scale distributed computing. The PC appears to provide hardware that scales up exponentially with time in performance, while scaling down in size, cost and power requirements. The internet appears to support twice as many users at ever-doubling performance each year. The web supports an information-sharing system that has evolved from a tool for a few research physicists, to become for some purposes a replacement for physical libraries, shops and banks.

However, existing methods of system design fail in critical ways to address the requirements we might reasonably expect from our medical scenario. In the next two sections we first discuss difficult properties of ubiquitous systems and then describe the platform from which we hope to tackle the associated problems. Many of these were identified by contributions to the Grand Challenges Workshop in 2002, followed by the Grand Challenges Conference in 2004. In the final section we outline steps that are being taken and propose criteria by which we may measure how well the Challenge has been met after 15 years.

Challenging properties of ubiquitous systems

A crucial property of ubiquitous computing systems is their invisibility. They are embedded in the environment. Their purpose is to serve people, and this certainly entails occasional interaction with users and guidance from them. However, most of their work will be done autonomously and humans will not be aware of it.

In this section we isolate many system properties needed to allow populations of 'entities', which may be hardware devices, software agents or infrastructure servers, to deploy themselves flexibly and responsibly in this work.

Context awareness: Entities need to know about their surroundings, including their own location, the presence of neighbouring devices and agents, and the available resources such as power and processing capability. Also, a ubiquitous system cannot assist users without being able to determine what they are doing.

Trust, security and privacy: A ubiquitous application may involve collaborations between *ad hoc* groups of entities. It may require migration (mobile agents) or downloading of code. New encounters occur, and there are complex issues of knowing what entities to trust. Does a server trust an agent enough to allocate processing resource to it? Does a device trust a neighbour to send message packets (which could be a 'denial of service attack' aiming to deplete the device's battery) for onward routing? An entity needs an authorization policy, based on trust, for what access it will permit to its resources, and for what services it should refrain from using itself. It may need to validate credentials without access to certification authorities. For this purpose a ubiquitous system may need to track users or other entities and determine patterns of their activity.

Relevance: Recent work proposes that ubiquitous systems use a 'publish-subscribe' and 'event-notification' paradigm. This model risks a tendency to broadcast information to all and sundry. Systems for filtering events based on interest or relevance to the consumer are needed, and they must scale (i.e. must themselves be distributed). So-called 'area-of-interest' management is currently an important component in multiplayer mixed reality games. We need a similar service in the pervasive computing paradigm.

Seamless communication: Ubiquitous systems need to interact with mobile users without wired infrastructure. This generally involves some form of wireless communication. Mobile users may need to seamlessly switch between different modes of communication, for example in a healthcare system when a doctor moves from a house to the street. Models of mobility must support the design of such systems.

Low-powered devices: A person with thousands of devices cannot keep changing batteries. The challenge is to design very low-powered devices, replacing the traditional emphasis on faster chips. Use of solar cells, fuel cells, heat converters and motion converters may all be possible. Operating systems and languages should themselves be designed to avoid squandering power.

Self-configuration: New approaches are needed to provide flexible and adaptable software and hardware both for mobile devices and the intelligent environment. The scale of these ubiquitous systems necessitates 'autonomic' (self-organizing, self-managing, self-healing) systems which can dynamically update software to cater for new services and applications. Problems of context-awareness, trust and security (mentioned above) arise again: as soon as a system can reconfigure itself, how can a migrating agent discover, in its new location, the resources and services it needs, and on what authority should they be allocated?

Information overload: A vast number of sensors can generate an equally vast quantity of data. To avoid overwhelming traffic, this data should be filtered, aggregated, reduced, ..., as close to source as possible. Sensors may have to be programmed to look for specific events. It may be better to send queries to sensor networks rather than to assume that they send information to databases.

Information provenance: We need to be told the authoritative source of information, that is its provenance. Provenance must be fine-grained; we need to know not only the document source but the source of each modification; even a single sentence. The problem, already familiar on the web, will be multiplied by orders of magnitude in a ubiquitous system, where both humans and agents need to assess the reliability of data.

Support tools and theories: All the familiar tools for design, programming and analysis, need to be rethought to work in these new environments. This includes software design tools, analytical tools for performance or correctness, and theoretical models, for example for trust, context, security, locality, mobility etc. There is no obvious way to adapt current tools and theories to the new context.

Human factors: Design of ubiquitous systems must cater for new modes of interaction and must enable 'invisible' ambient technology to be used by non-technical people. There is a need to support virtual organizations and mobile work arrangements. Social issues arise with new forms of interaction based on ambient technologies, for example deployed in the home, the workplace, museums etc.

Business models: Ubiquitous applications demand a support infrastructure which provides storage, processing, network connectivity and various services in the home, office, on planes, trains and even in the street. How is this paid for? What is the business model that will make sure this expensive infrastructure is available everywhere?

The platform: base camp for the Challenge

The UK's computing research has strengths in a number of areas that will allow us to mount an attack on some of the problems mentioned above.

Architecture: Research in communications and distributed systems has been a strength in the UK for three decades. It has increasingly involved networked systems. However, the properties of ubiquity listed above put a whole new set of pressures on processing, communications and data placement, especially in terms of the scaling and self-organization of ubiquitous systems.

Agents: Several UK research groups are expert in agent technologies, which have been very successful in the context of the web. However, this has typically involved only fixed networks. In mobile, wireless, pervasive and faulty environments, agents are a possible part of the solution but we need much more experience with building systems with agents in practice. We also need a cogent theory.

Models and languages: The UK has for over two decades been a leader in models for distributed computing, including mobility. Languages and analytical tools (such as model checking) have emerged, and the models have addressed networked systems and security. It remains to be seen whether they adapt to larger populations, to probabilistic analysis and to large hybrid systems.

Each of these communities has strong international connections so they may form a UK nucleus at the heart of an international attack on the Challenge. The EPSRC Network UK-UbiNet is providing a forum for these and other relevant communities to interact, often for the first time. It has already held two workshops with this purpose. The early contributions to our Challenge included augmented memory and physique, annotated recording of humans' experience and the interface questions that arise with physical/wearable/endurable computing systems.

At the UK-CRC Grand Challenges Conference in Newcastle (March 2004) a panel with some thirty attendees discussed ways to begin work on our Challenge, in collaboration with our sister Challenge Science for Global Ubiquitous Computing. We also established that there are strong and useful relationships with at least two of the other Challenges: Dependable Systems Evolution and Memories for Life. All the Grand Challenges are clearly distinguished by their ultimate goals; but in reaching these goals there is room for much synergy in projects that serve more than one of them.

First steps and ultimate goals

To mount our attack on the Challenge we would like to define a 'road map': a conjectured path or group of paths leading to the goals. At the Newcastle Conference our panel agreed that with so many new concerns, surveyed as 'challenging properties' above, we are not ready for that. We first need some initial experiments, which may be called 'foothill projects', to lead us to a situation where we can more clearly express a road map. A number of projects are already up and running. They will be joined by more to explore the problem space. Foothill projects should have well-defined accessible goals and should involve relevant research communities that have not normally collaborated. Such projects include:

Exemplar: Build a prototype of the medical scenario we described above.

Stochastic model and self-organization: Proving some useful properties of some self-stabilizing and self-organizing systems with a stochastic model checker.

Simulation/check of real Ubicomp code: The ability to simulate and check systems before running them in the jungle.

Performance analysis of resource management: Verification of some decentralized mechanism for resource management (e.g. power).

Correctness of performance: Correctness of hierarchical failure detectors to meet a quality-of-service requirement.

What then are our goals? They are certainly *engineering design principles*, and they must *scale*, that is they must be applicable to a subsystem of a 1000 agents just as much as to a system containing a 1000 such subsystems. It only remains for us to express this in a form against which success after 15 years can be measured:

> *To define a set of design principles that pertain to all aspects of ubiquitous systems, that are agreed by both the academic and professional engineers, that are taught regularly in Master's Degree courses, and that are instantiated in the design and rigorous documentation of several ubiquitous systems with a successful operational history.*

GC5 The architecture of brain and mind

Aaron Sloman

What is the most powerful and most complicated computer on the planet? Wrong! It is not a machine you can buy for millions of dollars; it is the amazing system that we all own, the few kilograms of grey and white mush in our heads... .

Brains are the most impressive products of biological evolution. We are still far from understanding what they do and how they do it, although neuroscience is advancing rapidly. Biological evolution, in combination with social evolution and individual learning, also produced our minds, which are intimately connected with our brains. Despite much progress in psychology, we are also far from understanding many of the things minds do and how they do them.

- **Brains:** The contents of our skulls are composed of extraordinarily intricate physical structures performing many tasks in parallel at many scales from individual molecules to large collections of cooperating neurones or chemical transport systems. In some ways brains are like computers, in so far as they have vast numbers of components that can change their states, and this allows them to perform a huge variety of different tasks. In other ways brains are very different from computers, though the study of brains has already begun to inspire the design of computers of the future.

- **Minds:** These are abstract machines whose existence and ability to do things depend on all the 'wetware' components that make up brains. But whereas brains contain visible and tangible things like nerve cells and blood vessels, minds contain invisible, intangible things like ideas, perceptions, thoughts, desires, emotions, memories, knowledge and skills of many kinds. Minds and their contents cannot be seen by opening up skulls, though their effects are visible all around us in many physical behaviours of humans and other animals and in enormous changes in the physical world brought about by mental processes, including the development of computers.

Explaining the connection between minds and brains was once thought the province of philosophy. We now see this as a scientific problem, which is inherently about relations between abstract and concrete information processing systems.

Gaps in our knowledge

Many processes in brains and minds are not yet understood, including how we:

- see many kinds of things around us;
- understand language, like the language you are now reading;
- learn new concepts;
- decide what to do;
- control our actions;
- remember things;
- enjoy or dislike things;
- become aware of our thoughts and emotions;
- learn about and take account of the mental states of others;
- appreciate music and jokes;
- sense the passage of time.

These all involve both abstract mental processes and concrete physical processes. We aim to explain how both kinds of processes work, and to demonstrate this in robots with a large collection of human-like capabilities, unlike current robots which are very limited in what they can do.

The historical roots of the project

This research addresses two quests that have fired the imagination of many inventors, scientists, philosophers and storytellers for centuries: the attempt to understand what we are and the attempt to make artificial human-like systems, whether entertaining toys, surrogate humans to work in inhospitable environments or intelligent robot helpers for the aged and the infirm.

Minds are machines: virtual machines

Minds are in some ways like software systems running on computers; both minds and software systems depend on complex physical machinery to enable them to exist and to perform their tasks. Yet neither minds nor running software systems can be observed or measured by opening up the physical machinery and using methods of the physical sciences, and in both cases the mappings between the abstract components and the physical components are very subtle, complex and constantly changing.

This strange relationship between invisible, intangible mental things and our solid flesh and bones has intrigued and mystified mankind for centuries. Despite the efforts of many philosophers and scientists it remained unexplained for centuries. However, around 1842 Ada Lovelace wrote some notes on the 'analytical engine' designed by the British mathematician and inventor Charles Babbage in which she showed a clear recognition of something that only began to become clear to others more than a century later. She noted that the analytical engine, like other calculating machines, acted on numbers, which are abstract entities, and observed that it

> *might act upon other things besides number, were objects found whose mutual fundamental relations could be expressed by those of the abstract science of operations, and which should be also susceptible of adaptations to the action of the operating notation and mechanism of the engine ... .*

She suggested, for instance, that using this abstract science of operations '... the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent'. This idea of a virtual (abstract) machine running on a physical (concrete) machine, and performing many abstract operations, was expanded and refined through developments in computer science and software engineering in the 20th century, following the pioneering work of Alan Turing[16] and many others. We now know how to design, make, debug, maintain and sell many kinds of virtual machines including word processors, email systems, internet browsers, operating systems, planning systems, spelling correctors, teaching packages, data mining packages, plant control systems etc. Many of them outperform humans on specific tasks (e.g. playing chess, statistical analysis), yet there is nothing that combines the capabilities of a three-year-old child, or even a squirrel or a nest-building bird. The most powerful chess machines cannot discuss or explain their strategies.

Animal brains in general and human brains in particular have many capabilities as yet unmatched by anything we know how to build. Can we ever replicate all that functionality? The availability of more powerful computers than ever before, advances in many areas of artificial intelligence and computer science, and an ever-growing body of research results from neuroscience and psychology provide the inspiration for a new concerted attempt at a major leap forward by developing new deep theories, tested in a succession of increasingly ambitious working models: namely robots combining more and more human abilities, for example an ability to see, manipulate objects, move around, learn, talk about what they are doing, develop self-understanding, think creatively, engage in social interactions and provide advice and physical help to others when appropriate.

Building on many disciplines

Much research in computer science departments is already strongly interdisciplinary but, like many major breakthroughs, this project will require new ways of combining results from researchers in several subjects, including:

- neuroscientists studying brain mechanisms and architectures;
- psychologists, linguists, social scientists, ethologists and philosophers studying what minds can and cannot do;
- researchers in computer science and artificial intelligence developing techniques for specifying and implementing many kinds of abstract mechanisms and processes in present and future physical machines;

[16]http://www.turing.org.uk/turing/

- researchers in mechanical engineering, materials science and electronic engineering extending materials and mechanisms available for robot bodies and brains.

Inspired by both past and future advances in neuroscience, the project will attempt to build machines that simulate as much as is known about how brain mechanisms work. In parallel with that, the project will attempt to implement many kinds of abstract mental processes and mechanisms in physical machines, initially without requiring biological realism in the implementation mechanisms, but gradually adding more realism as our understanding increases.

How will it be done?

Several mutually informing tasks will be pursued concurrently:

Task 1: Bottom-up specification, design and construction of a succession of computational models of brain function, at various levels of abstraction, designed to support as many as possible of the higher-level functions identified in other tasks.

Task 2: Codification and analysis, partly from a software engineering viewpoint, of many typical, widely shared human capabilities, for instance those shared by young children, including perceptual, motor, communicative, emotional and learning capabilities, and using them:
a) to specify a succession of increasingly ambitious design goals for a fully functioning (partially) human-like system;
b) to generate questions for researchers studying humans and other animals, which may generate new empirical research leading to new design goals.

Task 3: Top-down development of a new theory of the kinds of architectures capable of combining all the many information-processing mechanisms operating at different levels of abstraction, and testing of the theory by designing and implementing a succession of increasingly sophisticated working models, each version adding more detail.

Different research groups will study different parts of the problem, but always in the context of the need to put the pieces together in working systems. Analysing ways in which the models produced in task 3 succeed and fail, and why, will feed back information to the other two tasks.

More detailed information about the Architecture of Brain and Mind proposal can be found on its website.[17] The Grand Challenges website also shows the email discussion archives[18] leading up to the development of this proposal.

Targets, evaluation and applications

The primary goal is to increase our understanding of the nature and variety of natural and artificial information-processing systems. This is likely to influence many areas of research including psychology, psychiatry, ethology, linguistics, social science and philosophy. It could transform ideas about education.

As a 15- to 20-year target we propose demonstration of a robot with some of the general intelligence of a young child, able to learn to navigate a typical home and perform a subset of domestic tasks, including some collaborative and communicative tasks. Unlike current robots it should know what it is doing and why, and be able to cooperate with or help others, including discussing alternative ways of doing things. Linguistic skills should include understanding and discussing simple narratives about things that can happen in its world, and their implications, including some events involving capabilities, motives and feelings of humans. The robot could be tested in various practical tasks, including helping a disabled or blind person to cope without human help.

This long-term target will be broken down into a large collection of sub-goals, with different subsets used to define intermediate milestones for judging progress. Achieving all this will require major scientific advances in the aforementioned disciplines, especially if one of the sub-goals is the production of biologically plausible mechanisms capable of supporting the robot's functionality.

Success could also provide the foundation for a variety of practical applications in many industries, in unmanned space exploration, in education and in the ever-growing problem of caring for disabled or

[17]http://www.cs.bham.ac.uk/research/cogaff/gc/
[18]http://archives.nesc.ac.uk/gcproposal-5/

blind persons wishing to lead an active life without being totally dependent on human helpers. Perhaps some people reading this will welcome such a helper one day.

There is no guarantee of success. The project could fail if it turns out, for instance, that the problems of understanding how brains work are far deeper than anyone imagines, or if replicating their capabilities turns out to require much greater computer power than will be available in the next few decades. However, there is a distinct possibility that this research will eventually lead to the design of new kinds of computers that are far more like brains than current machines are.

International collaboration

Several international research programmes, including initiatives in cognitive systems in Europe, the USA and Japan are now supporting related research: international collaboration is essential for success in such a demanding project. Work inspired by the UK Foresight Cognitive Systems initiative will help with the foundations of the project. Several projects funded by the €25 million EC Framework 6 Cognitive Systems initiative starting in 2004, including a UK partner,[19] will provide significant steps towards the achievement of the Grand Challenge.

[19] http://www.cs.bham.ac.uk/research/projects/cosy/

GC6 Dependable systems evolution

Jim Woodcock

Background

A new technological product or system is dependable if, from its initial delivery, it justifies the confidence of its users in its soundness, security and serviceability. When the design of a system evolves to meet new or widening or changing market needs, it should remain as dependable as on initial delivery. Dependability in the face of evolution is currently a significant challenge for software systems, on which society increasingly depends for its entertainment, comfort, health, wealth and survival.

To meet this challenge we propose to elucidate and develop the basic scientific principles that justify confidence in the correct behaviour of software systems, even in the face of extreme threats. We propose to exploit these principles in the cooperative development of a strong software engineering toolset to assist in design, development, analysis, testing and formal verification of computer software systems of any size throughout their evolutionary life. We propose to evaluate and improve the tools by competitive testing against a representative range of challenge problems taken from real computer applications. And we propose to assist the producers of commercial programming toolsets to adopt the improved verification technology in their widely marketed products. We hope to achieve these goals in a 15-year international project.

If the project is successful, it could lead to a revolutionary long-term change in the culture of the producers and users of software. The users of software will expect and demand guarantees of correctness of software against agreed specifications, and perhaps will be entitled by law to do so. The technology of verification will be incorporated in commercial toolsets, and their use will enable software producers to meet their guarantees. Software will be developed and evolve faster and at significantly less cost than at present, because verification will reduce the need for lengthy debugging and regression testing of each change. A recent estimate has put the potential savings in 2003 to the US economy alone at up to $60 billion dollars per year.

Our ultimate goal is to meet the long-standing challenge of the verifying compiler. Its design will be based on answers to the basic questions that underlie any branch of engineering or engineering science, which ask of any product: what does it do (its specification), how does it work (its internal structure and interfaces) and why does it work (the basic scientific principles)? We will be inspired by the ideal of total correctness of programs, in the same way as physicists are inspired by the ideal of accurate measurement and chemists by the ideal of purity of materials, often far beyond the current needs of the marketplace. We hope that these ideals will be attractive to young scientists entering the field, and that each step in their achievement will be welcomed by professionals, teachers and scientists alike as a milestone in the development of the discipline of computer science.

Research agenda

Our Grand Challenge has two central principles: theory should be embodied in tools; and tools should be tested against real systems. These principles are embodied in our two main current activities: the creation of a strong software engineering toolset, the *Verifying Compiler*; and the collection of an appropriate range of examples, the *Challenge Repository*.

The verifying compiler

A verifying compiler is a tool that proves automatically that a program is correct before allowing it to run. Program correctness is defined by placing assertions at strategic points in the program text, particularly at the interfaces between its components. These assertions are simply executable truth-valued expressions that are evaluated when control reaches a specific point in a program. If the assertion evaluates to false, then the program is incorrect; if it evaluates to true, then no error has been detected. If it can be proved that it will always evaluate to true, then the program is correct, with respect to this assertion, for all possible executions. These ideas have a long history. In 1950, Turing first proposed using assertions in reasoning about program correctness; in 1967, Floyd proposed the idea of a verifying compiler; and in 1968, Dijkstra proposed writing the assertions even before writing the program.

Early attempts at constructing a verifying compiler were frustrated by the inherent difficulties of automatic theorem proving. These difficulties have inspired productive research on a number of projects, and with the massive increases in computer power and capacity, considerable progress has been made. A second problem is that meaningful assertions are notoriously difficult to write. This means that work on a verifying compiler must be matched by a systematic attempt to attach assertions to the great body of existing software. The ultimate goal is that all the major interfaces in freely available software should be fully documented by assertions approaching in expressive power a full specification of its intended functionality.

But would the results of these related research projects ever be exploited in the production of software products used throughout the world? That depends on the answers to three further questions. First, will programmers use assertions? The answer is yes, but they will use them for many other purposes besides verification; in particular, they are used to detect errors in program testing. Second, will they ever be used for verification of program correctness? Here, the answer is conditional: it depends on wider and deeper education in the principles and practice of programming, and on integration of verification and analysis tools in the standard software development process. Finally, is there a market demand for the extra assurances of reliability that can be offered by a verifying compiler? We think that the answer is yes: the remaining obstacle to the integration of computers into the life of society is a widespread and well-justified reluctance to trust software.

The challenge repository

A scientific repository is a collection of scientific data constantly accumulating from experimental or observational sources, together with an evolving library of programs that process the data. Running one of these programs on the data is a kind of scientific experiment whose success, and sometimes failure, is reported in the refereed literature. If they prove to be useful, the results of the experiment are added to the repository for use in subsequent experiments.

Our challenge repository will complement the verifying compiler by providing the scientific data for its experiments. These will be *challenge codes*: realistic examples of varying size. These codes will not merely be executable programs, but will be documentation of all kinds: specifications, architectures, designs, assertions, test cases, theorems, proofs and conjectures. They will have been selected because they have proved their significance in past or present applications. The following are just two well-known experimental applications that may be used.

The recently constructed Waterkering storm-surge barrier is the final piece of the Dutch coastal flood defences. As part of Rotterdam's harbour is behind the barrier, it has to be movable. Since it takes about 11 hours to close, the decision to do so is based on weather forecasts and tide predictions. It is a computer that takes this decision: there is no manual override. This is a safety-critical application: if the barrier is closed unnecessarily, then the work of the harbour is badly disrupted; but if the gates are not closed when they are needed, then the severe floods of 1953 may be repeated.

Mondex is a smart card used for financial transactions: the value stored on a card is just like real money. An on-board computer manages transactions with other cards and devices, and all security measures have to be implemented on the card, without any real-time auditing. This application is security-critical: it must not be possible for a forger to counterfeit electronic value.

More ambitious examples may include crash-proofing selected open-source web services, perhaps leading eventually to crash-proofing the entire internet. Many of the challenge codes will be parts of real systems and so they will be evolving at the hands of their owners and users; it will be part of the challenge to deal with this evolution.

Extending the horizons

The basic idea underlying all techniques aimed at achieving high dependability is *consistency checking of useful redundancy*. A verifying compiler aims to check the consistency of a program with its specification once and for all, as early as possible, giving grounds for confidence that error will always be avoided at run-time. In contrast the dependability technique of *fault tolerance* delays the checking until just before it is needed, and shows how to reduce the effects of any detected errors to a tolerable level of inconvenience, or possibly even to mask the effects completely. This technique is used for different kinds of faults, including software, hardware and operator faults, but the error checking will have been derived from the same specification that the verifying compiler would use. The same kind of redundancy is exploited, no matter where the consistency is checked.

Verification technology is applicable only to properties that can be rigorously formalized in notations accessible to a computer. Some of the important requirements of dependability, fault tolerance, component architectures and associated design patterns have resisted formalization so far. The availability of verification tools will surely give a fillip to research in these important areas.

Current activities

Our first task is to define the scientific goals and objectives of the Grand Challenge and we expect that this will take about three years of serious discussion. We have launched a series of workshops to begin this process and to expand the range of research activities to ensure that we address the issues encompassed by dependable systems evolution.

Workshop 1: Introduction. (November 2003, Queen Mary University of London.) This was dedicated to getting a core group of interested researchers together to get to know each other. Work started on a comprehensive survey of the state of the art. This will contain a list of leading researchers and teams around the world, with an annotated bibliography as background for the Grand Challenge.

Workshop 2: Toolset. (March 2004, Gresham College.) This involved a significant number of international Grand Challenge partners. It was devoted to discussing the theory and practice of existing tools, and to plan the components of a future toolset for the Challenge.

Workshop 3: Applications. This will be devoted to establishing a collection of suitable benchmark problems for testing the progress of the Grand Challenge, and a collection of common software design patterns suitable for formalization. The workshop will have a significant number of practitioners from industry.

Workshop 4: First steps. The fourth workshop will continue the work on publishing a definitive survey of the field, and be concerned with the detailed planning of the first phase of the project.

Workshop 5: Landmark events. The final workshop will be devoted to the organization of landmark events in the progress of the project, which may include the following:

- a Royal Society meeting for discussion to interface with the general scientific community;
- the founding of an IFIP (International Federation for Information Processing) Working Group on strong software engineering toolsets;
- a programme of meetings at the Royal Academy of Engineering to discuss experimental and future industrial applications;
- A one-week workshop at Schloss Dagstuhl, the International Conference and Research Centre for Computer Science;
- a meeting at the Isaac Newton Institute for Mathematical Sciences at Cambridge to discuss gaps in the theoretical foundations.

We will advertise the activities of the Grand Challenge through articles in the popular and learned press. We will maintain a comprehensive website to promote and disseminate our activities. We will engage in discussions in relevant newsgroups and email lists. One of the ambitions of a Grand Challenge is to capture the public's awareness of its aims and objectives. To achieve this, accessible accounts will be published of the background to the Challenge, its scientific goals and a vision of future exploitation. This work will in no way compromise the scientific promise or integrity of the project to attract public attention or support. More technical articles will be published, particularly summaries of the proceedings of the workshops and final conference. Every article will also be available online on the Grand Challenge's website.

This period of planning will close with a final conference, which will provide a showcase for a summary of the Challenge and the public presentation of the scientific research programme. It will also provide a forum for the formation of teams to carry out the proposed research. The most important product of the initial programme of work is to produce a 'road map' of strategic directions: a list of long-term scientific goals. These will be dedicated to extending current scientific knowledge and understanding necessary engineering skills. The road map may contain the following: a survey of known effective methods that can contribute to our scientific goals; pathways from successful research to new fields for research and application; and a list of open research questions, preferably with simply verifiable answers. The road map will have independent value as a reference for subsequent research proposals.

In preparing this description the author was given valuable assistance by the GC6 Steering Committee: Keith Bennett. Juan Bicarregui, Tony Hoare, Cliff Jones, John McDermid, Peter O'Hearn, Brian Randell, Martyn Thomas and Jim Woodcock.

GC7 Journeys in non-classical computation

Susan Stepney

Challenging paradigms

Today's computing, classical computing, is a remarkable success story. However, there is a growing appreciation that it encompasses only a small subset of all computational possibilities. There are several paradigms that seem to define classical computing, but these may not be valid for all computation. As these paradigms are systematically challenged, the subject area is widened and enriched. The Grand Challenge Journeys in Non-classical Computation is nothing less than a reconceptualization of computation.

Physical embodiment

Classically, computation is viewed mathematically in terms of algorithms, complexity and such like, based on the underlying essentially mathematical model of the Turing Machine.

Computation is physical; it is necessarily embodied in a device whose behaviour is guided by the laws of physics and cannot be completely captured by a closed mathematical model. This fact of embodiment is becoming ever more apparent as we push the bounds of those physical laws. The mathematical model of the Turing Machine is not an adequate model of reality for all notions of computation.

For example, one of the most exciting (and seemingly weird) recent developments is the non-classical paradigm of quantum computing. This has emphasized the fundamental link between computation and its physical embodiment in the real world, in that quantum superposition and quantum entanglement can lead to results simply not possible classically. Still in relative infancy, it holds out the promise of massive increases in computation power, of untappable communication channels and of spooky effects such as quantum teleportation.

To emphasize the critical importance of the underlying physical laws on computation, it has been shown that the deep unsolved conundrum of whether $P = NP$ is in fact answerable, in the affirmative, if the laws of quantum mechanics are non-linear. The fact that the precise form of the laws of physics can have an impact on what is classically thought to be a purely mathematical question is considerable food for thought.

At the opposite extreme, back in the everyday classical physical world, the importance of embodiment can turn up in strange places. For example a computation, being embodied, takes time to execute and consumes power as it executes, usually in a data-dependent manner. These time and power consumption side channels can be measured and analysed, and such analyses have been used to attack the security mechanisms of certain smart cards. Side channels are ones outside the mathematical model of the computational system. Even if these channels are then explicitly added to the model, the world is open, and further side channels always exist: no mathematical model of the world is complete.

Bio-inspiration

Classical computations tend to be very susceptible to unexpected changes. We have to carefully manage their inputs and interactions, keeping them locked in a safe, controlled environment. As our ambitions grow and our computational devices interact more with the open, noisy, uncontrolled real world, it seems we need a radically new way of designing and implementing programs. So, how can we build complex computational systems – systems that are autonomous, adaptable and robust – from millions of less reliable and simpler components? How can we build them to perform correctly and safely in an unpredictable, changing and hostile environment, and over long periods of time? Such tasks are currently well beyond the state of our computational art, and as our technology moves towards ever smaller and ever more numerous nanoscale and quantum devices, these tasks will get only more difficult.

And yet biological processes manage to do such things routinely. Living creatures are remarkably robust and adaptable. They can survive injury, damage, wear and tear, continual attack from other creatures, and can adapt to their environment. Biology manages to take huge amounts of potentially unreliable matter and use self-checking, self-repair, self-reconfiguration, multiple levels of redundancy and multiple levels of defence, to develop adaptable complex biological organisms that continue to work for long periods of time in an extremely hostile, changing environment.

So in an attempt to cope with complexity, researchers are drawing inspiration from biology, which seems to have already discovered the answers, in order to develop a host of bio-inspired algorithms from evolution (genetic algorithms, genetic programming), neurology (artificial neural networks), immunology (artificial immune systems), plant growth (L-systems), social networks (ant colony optimization) and much more.

Much progress is being made in the area of bio-inspiration, yet the resulting algorithms do not fit well into a classical computational framework. At the very least, they are inspired by highly parallel systems of multiple autonomous interacting agents, yet are often unnaturally sequentialized to fit the classical von Neumann execution paradigm. Additionally, the algorithms exhibit emergent properties: properties at one level that are not expressed in terms of, and not directly mappable to, the concepts available at a lower level. Emergence does not fit within the classical development process of stepwise refinement and correctness arguments. The most important divergence may be the fact that biological organisms are themselves embodied in a physical, chemical, biological world. For example, evolutionary algorithms model DNA as a string of ACGT bases, and from this build metaphorical chromosomes of bits or numbers. Yet DNA encodes for proteins, proteins fold into complex three-dimensional shapes, and it is these physically embodied shapes that interact by chemical forces with other shapes to perform their functions.

Complex systems

Researchers are also beginning to explore open complex adaptive systems, where new resources, and new kinds of resources, can be added at any time, either by external agency or by the actions of the system itself. Such new resources can fundamentally alter the character of the system dynamics and so allow new possibilities and new adaptations. Our current computational systems are beginning to open themselves, for example, through the continuing dialogue between user and machine, through continued new connections to networks such as the internet and through robotic systems controlling their own energy sources.

Open systems do not fit well into a classical framework either. Turing machine computations are defined in terms of halting computations, not ongoing ones with continual interactions with an open environment. Classical specification and refinement paradigms assume a predetermined phase space, with all the component types known beforehand. Yet a biological evolutionary system, or even a complex chemical catalytic network, is constantly creating new species, is constant surprising.

Enriching computation

Classical physics did not disappear when modern physics came along: rather its restrictions and domains of applicability were made explicit and it was reconceptualized. Similarly, these various forms of non-classical computation – embodied computation, bio-inspired algorithms, open complex adaptive systems and more – will not supersede classical computation; they will augment and enrich it. In the process, classical computation will inevitably be reconceptualized. This Grand Challenge seeks to explore, generalize and unify all these many diverse non-classical computational paradigms, to produce a fully mature and rich science of all forms of computation that brings together the classical and non-classical computational paradigms. It is clearly closely linked with two sister Grand Challenges: *In vivo–In silico* and Science for Global Ubiquitous Computing.

Such a mature computational science will allow us to design and build robust, adaptable, powerful, safe, complex computational systems. It will help researchers to uncover deep biological truths: which features of biology are necessary for correct robust functioning and are therefore true of any living organism; which are necessary only because of the particular physical realization (carbon-based terrestrial life forms); which are merely contingent evolutionary aspects and so could be different if ‘the tape were played again’. It will help researchers to uncover deep physical truths: what is the relationship between logical information (bits) and physical reality?

Journeying hopefully

Many Grand Challenges are cast in terms of goals and end points, for example 'achieving the goal, before this decade is out, of landing a man on the moon and returning him safely to earth', mapping the human genome, proving whether $P = NP$ or not. We believe that a goal is not the best metaphor to use for this particular Grand Challenge, however, and prefer to use that of a journey. The metaphor of a journey emphasizes the importance of the entire process, rather than emphasizing the end point. Indeed, we do not even know the end point; ours is an open journey of exploration. On such a journey, it is not possible to predict what will happen: the purpose of the journey is discovery, and the discoveries along the journey suggest new directions to take. One can suggest starting steps, and maybe some intermediate way points, but not the detailed progress, and certainly not the end result.

First steps

In this light we have mapped out some initial journeys, the first steps, of the Challenge. These more detailed statements are necessarily influenced by the specific interests of the researchers involved, but their purpose is to indicate the kind of research that we believe needs to be done to start the Challenge. (See the GC7 website for the full statements of these journeys.) Each of these journeys is already underway, to a greater or lesser extent, in the UK and worldwide.

Non-classical philosophy – rethinking classifications: Recent philosophy, of Wittgenstein and Lakoff in particular, emphasizes that people do not classify the world into sets defined by necessary and sufficient conditions. Yet this is one of the artificial constraints we impose on our programs, which may be why they are inflexible and lack 'common sense'. The challenge is to construct computing based upon family resemblance rather than sets, on paradigms rather than concepts, on metaphor rather than deduction, and to devise systems that make context-sensitive judgements rather than take algorithmic decisions.

Non-classical physics – quantum software engineering: Quantum computation and quantum information processing are in their early days, yet the more we look at them, the weirder their consequences appear to be. The challenge is to rework and extend the whole of classical software engineering into the quantum domain so that programmers can manipulate quantum programs with the same ease and confidence that they manipulate today's classical programs.

Non-classical refinement – approximate computation: In interacting with the continuous, noisy real world, a radical departure from discrete 'correct versus incorrect' computation is required, a shift away from logics towards statistical foundations, such that meaningful estimates of confidence emerge with each approximate result.

Computing in non-linear media – reaction-diffusion and excitable processors: Chemical reactions can exhibit excitation waves. Such waves do not conserve energy; rather they conserve waveform and amplitude, do not interfere and generally do not reflect. These are properties that may be used to perform a novel form of embodied computation. The challenge is to develop a science of computation using spatio-temporal dynamics and propagating phenomena, in many-dimensional amorphous non-linear media.

Artificial immune systems (AIS): AIS are exciting new forms of bio-inspired computation. There is a broad and rich set of models available for investigation, from population-based selection models to dynamic network models. The challenge is to develop a corresponding computational theory of the variety of AIS-inspired algorithms and to extend and unify these with other bio-inspired results.

Non-classical interactivity – open dynamical networks: Dynamic reaction networks can have complex non-linear interactions and feedback where reaction products may themselves catalyse other reactions in the network. They exhibit the complexity, complex dynamics and self-organizing properties of many far-from-equilibrium systems. Computation can be viewed as a dynamical process. The challenge is to develop a pragmatic theory of dynamic, heterogeneous, unstructured, open networks, and thence to develop a computational paradigm in terms of dynamical attractors and trajectories.

Non-classical architectures – evolving hardware: Most bio-inspired algorithms are implemented in software, in disembodied simulations. But hardware can have bio-inspired properties too and can exploit unexpected properties of embodiment, such as the analogue characteristics of devices. The challenge is to use such properties to develop (bio-inspired) computing hardware that can adapt, evolve, grow, heal, replicate and learn.

Non-classical architectures – molecular nanotechnology: Molecular nanotechnology envisages the use of hordes of molecular-scale robots to construct macroscopic artefacts as guided by their programming. Centralized control is infeasible. The challenge is to find ways of designing and assembling artefacts in ways that can be described in terms of predominately local interactions, that is, in terms of the emergent properties of vast numbers of cooperating nanites. This requires analysis of emergent behaviour, of growth and of complex adaptive systems.

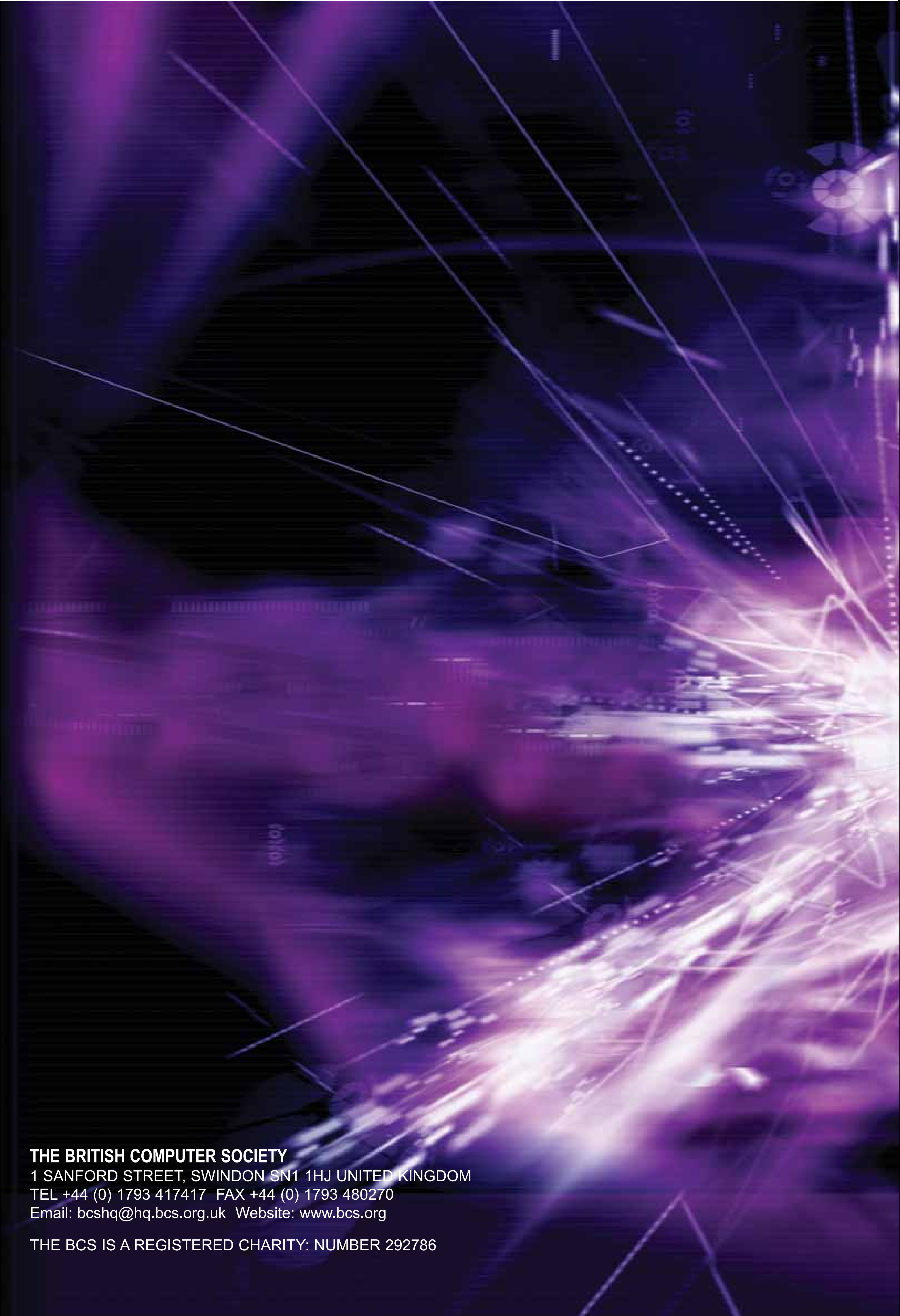
Non-von Neumann architectures – through the concurrency gateway: The classical von Neumann sequential execution paradigm forces computation into a sequential straitjacket. Classical approaches to overlaying a concurrency framework are conceptually very difficult to program and expensive in time and memory to implement. The challenge is to enable concurrency to be a fundamental modelling and programming concept with a simple conceptual model that can be efficiently implemented.

The Grand Challenge umbrella

These first-step challenges cover embodiment, bio-inspiration and complex systems, sometimes within the same journey. Such wide-ranging and novel research necessarily requires multidisciplinarity, with computer scientists working alongside biologists, chemists, physicists, mathematicians, materials scientists and hardware engineers.

The purpose of the overarching Grand Challenge[20] is to coordinate such separate journeys, to cast their research hypotheses and view their results in terms of the overall vision, and to abstract out common results into new paradigms. These individual exploratory journeys, and others yet to be defined, form part of the process of mapping out the wide-ranging new discipline of non-classical computation. The GC7 group is building a network of interested researchers to this end. If you would like to join us in our journey please visit the Grand Challenges website to subscribe to the GC7 mailing list, or contact the GC7 moderator.

[20] http://www.cs.york.ac.uk/nature/gc7/



THE BRITISH COMPUTER SOCIETY
1 SANFORD STREET, SWINDON SN1 1HJ UNITED KINGDOM
TEL +44 (0) 1793 417417 FAX +44 (0) 1793 480270
Email: bcshq@hq.bcs.org.uk Website: www.bcs.org

THE BCS IS A REGISTERED CHARITY: NUMBER 292786