Journeys in Non-Classical Computation

A Grand Challenge for Computing Research

Notes from the York Workshop, 30 September 2003

Susan Stepney

Introduction

A workshop was held at the Department of Computer Science, University of York, on Tuesday 30 September 2003, to discuss ways to progress the nascent Grand Challenge, including submitting a Network proposal to EPSRC. Below are my record of the workshop, produced from notes taken at the time, and some subsequent thoughts, with a fairly arbitrary structure imposed after the event. This is a record of a group discussion – I have made no attempt to attribute the individual sources of ideas.

Attendees

Andy Adamatsky – U. West of England – CAs, reaction-diffusion computers
Tetsuya Asai – Hokkaido University – reaction-diffusion computers, nanotechnology
Jim Austin – U. York (CS) – neural-inspired architectures
Sam Braunstein – U. York (CS) – quantum computation, foundations of information theory
John Clark – U. York (CS) – evolutionary computation in crypto/security/testing
Richard Greaves – U. York (Biology) – protein structure
Mark Hylton – EPSRC – Novel Computation
Colin Johnson – U. Kent – bio-inspired computing
Julian Miller – U. York (Electronics) – emergent computation, bio-inspired, quantum
Mark Neal – U. Wales Aberystwyth
Derek Partridge – U. Exeter – precise approximate computing
Vic Rayward-Smith – U. East Anglia – optimisation, robustness, emergence
Rob Smith – U. West of England – complex systems, evolutionary computing
Susan Stepney – U. York (CS) – emergence, self-organisation, nanotechnology
Jon Timmis – U. Kent – bio-inspired computing, immune systems
Andy Tyrrell – U. York (Electronics) – adaptive/evolving/healing/replicating hw
Klaus-Peter Zauner – U. Southampton – protein structure, molecular computation

Complex Systems and Novel Computation

- There is a need to open up the narrow concept of CS
  - Early CS was very broad, then narrowed to the "classical" view of Universal Turing Machines that merely need more and more speed
  - Even the CS notion of Formal Methods is very limited
    - mathematics is a much wider subject area
    - statistics
  - We need a clear definition of "computation", a general model encompassing all forms
    - Otherwise our domain is *everything*
    - Protein folding is not effectively computable – but proteins do it!
    - Three bodies can solve the Three Body problem
    - Need an analytic, not synthetic, definition – currently we talk about things we build – it's difficult to point at something and say "it's computing"
    - What about non-halting Turing machines, where you look at the tapes as they run?

        - If we exclude the trivial *self*-computation, and have some encoding/decoding of inputs/outputs, and let the laws of physics "just do it" (not forcing the trajectory) – that's computation – analogue computation
            - There also needs to be a compressed form, a representation, some idea of "programming"
        - Maybe a series of articles on "computing without brains or computers"
    - We need to move away from TMs. People invent a novel paradigm, then show it is possible to implement a TM with it. We don't need another TM!
        - We can be interested in special-purpose devices, they don't all have to be Universal.
        - Time is not part of the Turing paradigm – interaction with real time unpredictable environment is important – you cannot pay the price of a Virtual Machine
        - But it can be fun to build a TM out of weird stuff
        - TMs "converge" – continual innovation / change *needs* to go outside the paradigm
        - Even within Turing computation there are lots of hard problems
        - AIS, GAs, NNs – may still be TMs underneath, but they are still interesting – because we think about them differently

- Classical CS is good at "computing ballistic tables", but not good at soft, AI tasks – and more speed just won't help
    - But there is an external perception that more speed *is* all we need
    - Sub-challenges of things we can't do at the moment
        - Smaller, focussed challenges, like the "RSA challenges", with prizes? (Money, or just GC "credits") – eg "factor 100 digit numbers using a GA", "design a homeostatic system"
        - There are Data Mining competitions
        - A "most interesting computation" challenge – like "The Great Egg Race"
        - "Autonomous Robot Wars"
    - Not only do we need it, we're getting it because it can't be stopped! The Web has a substantial descriptive element
    - NY and Italian power cuts – emergence happens!

- Need recognition that the classical approach is seriously incomplete
    - the idea that "if you do it correctly, that's all you need" will *never* happen in practice
    - the paradigm of "what we want is the right answer" is the *problem*
        - I don't expect my human companions to be 100% reliable/accurate
        - We want fuzzier ideas, of "competence"
    - biology is never clean or optimal, but is self-sustaining
        - Google is teaching that optimality isn't necessary – as long as the answer is there ion the first page or two, it doesn't have to be the "Am I feeling lucky" one
    - requires an acceptance of constant level of "light" failures

- With Neural computing – it was very exciting in the early days – we solved all the easy problems – now we are left with the *hard* problems – they are fundamental, but need hard work

- Problems of monocultures

- CS is taught as an engineering subject – it needs to be something else
    - CS is problem driven – we spend very little time doing "science"
    - There is a clear distinction between Chemistry (science) and Chemical Engineering – CS is rather like alchemy – we need an analytic science to backup the synthetic engineering

- Major problem – we don't know what techniques work where
    - No Free Lunch theorems
    - Find conditions under which the NFL assumptions don't hold – where Bayesian learning occurs
    - Characterise the problems that bio-inspired algorithms are good at

- Classical sw development : start with a spec, then target a “big enough” platform – what about starting with very limited resources – what’s the best you can do?
  - Very low power (ubiquitous devices)
  - Very few qbits (small quantum computers)
  - What if I have something doing a computation, and I keep removing bits – can it still do the same computation (cf the scene removing chunks from HAL in *2001: a Space Odyssey*)
- (Simulating) these techniques can eat all the classical computing power there is
  - we need to “soften the interface” to High Performance Computing
- Importance of the environment
  - Robot simulations have a very restricted environment – real robots behave differently – wheels skid, etc – have to cope with water, leaves, mud
    - Robots should be able to exploit the environment, not just “avoid” it
  - The Internet is a sufficiently complex environment for sw robots
    - Can we use properties of the Internet to do a computation? [Someone has already done calculations using packet header information]
- We will need the techniques of the GC to solve, or “do”
  - We are a service GC!

A Network Proposal

Input from EPSRC

EPSRC support Network grants, for ~£60k/2 years. These grants can cover coordinator salary, workshops, travel, etc, but are not for doing research.

A Network grant proposal should cover (at least):

- Aims and objectives
- Relationship to other activities – especially Novel Computation clusters, Quantum IRC
- Timing – why now?
- Deliverables

There is also other funding, for Public Awareness of Science, for PhD summer schools, …

We should decide whether the Network is pure CS, or inter-disciplinary. [It was agreed that we definitely want an interdisciplinary activity.]

Discussion

- There is a need for an umbrella network
  - Capture the knowledge from the Novel Computation projects as they run
  - Also covers Quantum Computation, and other areas
  - Links to European initiatives, SFI, etc – and raise out international profile
  - Clear statement of what we intend to do, like
    - “to understand and develop a science of non-Turing computation”
    - “to broaden the definition of computation beyond the Turing Machine paradigm”
    - “to restore computing to the broadest vision of Turing and Von Neumann” (away from the impoverished models of today)
- What do we want to achieve? We can all get together and brainstorm, but what next?
  - Identify and understand the big problems
  - Establish a common language for CS and biologists

- Educational aspect
  - Resource sharing
    - developing Open Source simulations, toolsets
    - identification of resources, things already done, things needed
      - develop “responsive mode”-style bids to plug the gaps (infrastructure projects need to have a research component, to mollify referees, but backing from a large diverse community should also help)
  - Teaching CS people about real science practice (experimental design, etc)
  - Showing the next generation that CS is a fun area – bring the excitement back
    - There is no public awareness of advances in CS – it’s “finished” – it used to be rocket science, now you get better graphics on a home games console
    - People now expect so *little* of computers – where is the “intelligent assistant” that I can talk to and solve problems with?
    - The CS school syllabus is *very* dull! – an ‘A’ Level in Complexity?
    - As are many undergraduate CS courses
  - Populist book
    - Seriously explain the benefits and potential of a true understanding of complexity and emergence
  - Promote Non-Classical Computation as a valid area of study
    - Currently, people just “blunder” into it from random backgrounds
  - Do something for the British Association for the Advancement of Science
- People
  - We need the right people to contribute to the proposal, not just CS
    - Biologists, physicists, mathematicians, philosophers, sociologists, psychologists, …
    - Industrialists
  - Go for an initial team, with activities planned to cover all the issues
    - Stream at suitable International Conferences
    - Workshops, annual, 30–40 attendees
      - Focussed objective for each workshop
      - “Dagstuhl”-like, eg Cumberland Lodge, Newton Institute – need to book well in advance
      - sessions timetabled 8–12, 5–8 to allow the afternoons for networking, discussions, etc
    - More frequent, smaller scale, more highly focussed research meetings, 5–6 attendees
      - Need to commit a reasonable amount of time, 3–4 days, to get full benefit
    - travel and accommodation for “mini-sabbaticals”, a week at another institution
- Infrastructure
  - Technical support / web site / workshop admin
- Ways to encourage radical thinking
  - “Thinking outside the box *outside* the box”
  - Host a series of “green papers”, ideas that cant get published conventionally (where “green” is a reference to the colour of the ink used…)
  - “Concepts” workshops
  - Develop a community that supports whacky thinking in this CS area
    - Referees who believe Non-Classical Computation is worth funding
    - Natural Sciences foundations change very slowly, but CS runs on a “memetic timescale”