Computing with People

Jon Crowcroft
University of Cambridge
15 JJ Thomson Avenue
Cambridge, CB3 0FD, UK
jon.crowcroft@cl.cam.ac.uk

Computing is ubiquitous. However, much of the growth in computing is hidden from sight, leading to a common assumption that it isn't interesting. Yet today's transport, homes, power, sports, health and entertainment depend increasingly on complex and interesting interconnected computing systems, running a wide variety of software. In these five lectures, I will explore five aspects of these systems, through interactive demonstrations with the audience, and, *crucially* without the use of gadgets, devices or technology that may obscure the fundamentals of what is going on.

Systematic Problem Solving Computing is the science of problem solving. We proceed by breaking a task down into components, and then abstracting away the detail, and then writing down an unambiguous set of instructions on how to solve a problem. The solution, a program, can be run by a digital computer (PC, desktop, smart phone) or by people. I will illustrate this through a small classic example, of the Fibonacci sequence, pausing to note that this also occurs, in the way that winding plants and shells grow, in nature. In the second half of the lecture, I will illustrate recursion through audience participation in self-sorting by age.

From Data to Information to Knowledge Let's play a game like *20 Questions*, but we'll change it just a little bit. I will pick a number between 1 and 32, and I'll write it down. Your task is to figure out which number I have chosen, by asking me a series of 'yes/no' questions, and I promise that I will answer every question truthfully with a yes or a no. How many questions do you think you need to ask me in order to be guaranteed of finding out which number I have chosen? Maybe 31? Would you believe that you can always do it in at most 5 questions? What sorts of questions might those be? Would it be useful to ask questions like "Is it 17? Is it 26? ... In the second part of this lecture, we will try out the Three Ballot voting system, which is a clever mechanism that allows a voter in an election to check if her vote has been counted properly, but without letting anyone else check who she voted for!

Networks, Naturally Computers are connected. Web pages are linked. Humans form social groups. Networks show up in nature in many many ways. Humans gossip, diseases spread in epidemics, species form herds of predators and prey, your cells signal each other, proteins and genes relate in different structures. I will ask the audience to discover the ways in which they are related through schools, families and interests. I will then conduct an experiment in gossiping, to see how quickly and accurately a rumour can spread or percolate from one side of the room to the other. In the second part of the lecture, I will construct a game where misinformation and information are both spread at different rates, by giving audience participants different roles as agents and counter-agents, to see what the resulting emergent states are for differing levels of misbehaviour.

Graphics and Virtual Worlds 3D scenes that never existed in the real world show up more and more in every day life. In this lecture, I will give the audience a description of a room and a light source, and a set of simple to follow instructions, that end up with them holding up a particular coloured card: a camera will then show that they have created an image of the scene with (say) specular reflection from a cup of tea. [The organisation of people in the room to carry out this task continues from the previous lecture]. And that's what happens in the graphics card in your XBoX or in Pixar's cluster of workstations when they make one frame of a computer animated movie! When there is motion, objects must follow the right physical laws: hence VR systems contain simulations of the real world.

Hacking, Debugging and Scientific Method When we include humans in the picture with computers, things often go wrong in wonderful ways. There are many examples of bad design that doesn't allow for how people really behave. In this lecture, I will use the example of the Heathrow T5 car park system which frequently fails to help people find their cars. The audience will be asked to try and *debug* the system, as an example of the scientific method. In the second part of the lecture, I will talk about the relationship between the psychology of debugging and hacking. This links back to social networking again.

Each lecture will contain brief video in the middle, of a well known public figure relating an experience: I will invite Jeremy Clarkson to talk about his experience of publishing personal bank account information, Lily Allen on social networks, and Stephen Fry to talk about gadgets.

At the end of the lectures, the audience should now have a clear understanding of the fascinating science that underlies computing, and how this science underpins and relates to the science of the natural world.