The Royal Society of Edinburgh

Franco–Scottish Science Seminar Series

The Royal Society of Edinburgh and the French Embassy in London are collaborating in a three-year programme of science events designed to explore and publicly present areas of science where both Scotland and France have a powerful presence. These events are intended to stimulate Franco–Scottish collaboration in science, present new scientific ideas and their social and commercial implications to the public, and increase awareness of French and Scottish science in each other's country.

In this first year of the programme, the events focused on information technologies and how these are changing the ways we understand and construct the world, by providing new ways of sensing, communicating and analysing data. Informatics is the new science of information that underpins this revolution. While playing an ever-increasing role in the foundations of our new world, Informatics remains largely unknown to the general audience.

The events in 2010 were presented by the Royal Society of Edinburgh, the Collège de France and the University of Edinburgh, supported by the French Embassy, l'Institut National de Recherche en Informatique et Automatique (INRIA), and the Scottish Informatics and Computer Science Alliance (SICSA).

- The inaugural lecture described the heart of Informatics as a science and explained why it has such a huge impact on society, industry and other sciences.

- The second lecture was devoted to models of computation, key concepts of informatics needed to express, analyse and implement any kind of algorithm.

- The third lecture was concern with digital circuits and modern systems on chips, which are the engines of Informatics.

- The fourth lecture analysed finite-state systems, which are used in a very wide variety of applications and led to a wonderfully elegant theory developed over the last 60 years.

- A short fifth lecture introduced synchronous programming, a programming model used in applications ranging from major safety-critical industrial systems to musical composition; this introduction preceded a music seminar and concert featuring human–computer interaction.

- The sixth lecture explained the theoretical and practical foundations of networking, one of the most visible application fields of Informatics.

- The seventh lecture studied embedded systems, i.e. Informatics in objects, and revisited synchronous programming.

- The following event summary report covers the eighth and final lecture in this year's series of events.

The Royal Society of Edinburgh

Franco–Scottish Science Seminar Series:
Seven Keys to the Digital Future

Professor Gérard Berry (INRIA and Collège de France)
and Professor Chris Bishop FRSE (Microsoft Research Ltd)

7 October 2010

Reports by Peter Barr

Getting rid of bugs?

Professor Gérard Berry *is determined to exterminate bugs – and in his quest for error-free software, he is interested in many very different applications, from electronic circuits to flight-control systems for airplanes and space probes.*

The problem with computers, said Berry right at the start of his talk, is the gulf between computers and the people who program them. While people are clever, only semi-rigorous and slow at performing computations, computers are superfast, superexact and superstupid – and this leads to a "titanic fight" between machines and humans. Bugs are the collateral effect of this fight. Bugs are only found to that scale in computing, he added, not in other disciplines such as physics, because computers are "the best error amplifier ever invented" – sometimes leading to human as well as financial disasters. A large part of computer science, he explained, is about developing models and theorems to bridge the gap and reduce or avoid bugs.

Later, one of the audience questioned the value of testing software to the limit when "the vast majority" of software bugs were not mission-critical or even expensive, but Berry responded by saying that certification pays off in the middle term, so it is always "economical to work in better ways." Bugs are not just numerous but very expensive to fix and customers are less tolerant than they used to be when they experience failures – even when it comes to basic applications such as form filling and word processing.

To describe the history of bugs and the scale of the problem, Berry then quoted computing scientist pioneer Maurice Wilkes, who said in 1949 that writing a program was not quite as easy as he had thought, in terms of potential for error. In fact, the history of computing is infested with errors, and Berry gave several examples:

- The Zune MP3 bug (2008), which made the device unusable for 24 hours, caused by a problem with leap years.
- The similar Sony PS3 bug (March 2010), which created havoc amongst gamers.
- The Patriot missile failure during the Gulf War (Dharan, 1991), which enabled an Iraqi Scud missile to penetrate Allied defences, leading to 28 deaths; it was caused by accumulated time measurement rounding errors.
- A buffer overflow problem (1993); this kind of bug is a major headache for Microsoft and others, since it provides an entry point for malicious programs (i.e. worms), but it also provides a major incentive for formal verification of programs.

- The Pentium Floating Point division bug (1993), which cost Intel $475 million.
- A bug which caused AT&T's telephone network to crash for 10 hours in 1990.
- An outage at Facebook (September 2010), which caused "self-denial" of service.
- The explosion of the Ariane 5 rocket in 1996, when code originally written for the Ariane 4 led to failure. "The code was actually useless!" said Berry.
- The Pathfinder Mars probe (1997), which almost failed because of "task priority inversion" – a known problem for which a solution had already been published but was ignored.
- The near–failure of the Spirit Mars Rover (2004), due to a flash memory handling problem.
- The Mars Orbiter crash (1999), due to a confusion in the software between metric and imperial measurements
- The Therac-25 bug (1985–87) led to cancer patients being given fatal radiation overdoses, attributed to "poor error management and a sequence of bad fixes."
- A wireless security flaw which enabled remote-access researchers to take full remote control on commercial Pacemaker (2008).

After this impressive list of failures, Berry moved on to identify the major causes of bugs – for example, not viewing software engineering as a central activity, using reasoning valid in one sphere which may not be valid in software, performing tests to verify what *should* work instead of finding what will *not* work, and "forgetting to verify verification."

The solutions, said Berry, include the use of the appropriate design tools, making everything visible ("you can see a hole in the wing of a plane but you can't see a hole in software"), independent reviews, systematic testing by simulation or using real targets, certification, and formal verification. Full verification is "very ambitious," he added, because it means "complete understanding." Partial verification, or focusing on the most important properties of a system, can be very helpful when it comes to issues like safety, he said. But system compositional verification (verifying parts and then part composition separately) will be mandatory, but is hard to achieve.

Berry's specialist areas are the formal design and verification of circuits and software. He was Chief Scientist of the French company Esterel Technologies, developing Esterel Studio and SCADE, a family of products which helps with the design and development of mission- and safety-critical embedded hardware or software applications – for example, avionics, aerospace and nuclear energy. In avionics, the embedded software should be certified as part of the plane. In all such applications, traceability is vital to certification, Berry added, from mapping of requirements through design and coding to integration. Berry claimed that SCADE can speed up application certification from three weeks to only one day, from the fact that the SCADE compiler is itself certified at the same level as the plane. He would very much like to transfer similar results to the medical field.

Berry then discussed the problems of formal software verification, with the examples of sorting and factorial functions, using induction and termination proofs. He described the formal tools required to identify and eradicate errors – including languages, static analysis, model checking using backward and forward system transition exploration, and theorem proving. With model checking, explicit engines are good at checking applications such as communications protocols, while implicit formulae-based engines are good with applications such as circuit synthesis. "Proving Boolean logic was thought to be impossible," but recent techniques have proved efficient for actual applications. However, the subject is not yet well understood, and computer scientists sometimes use “alchemistry” rather than maths. For theorem proving, “Proofs in maths should be verifiable by stupid machines,” he said, and computers are good at that. Major applications such as compiler and operating systems kernel have been recently verified by theorem proving. To illustrate some of his points, Berry talked about verification of programs written to solve Sudoku problems – “taking one second to prove it's not magic but real” – and quoted the famous Curry-Howard principle which states that “computing is the same thing as proving.”

To summarise, Berry shared his shopping list of “bug food” – not paying enough attention to software engineering, loose or ever-changing specifications, poor documentation, programming and verification, and poor code maintenance. His solutions were first to “starve” the bugs, using the appropriate design processes and tools, and secondly to use powerful formal tools. But even though the ultimate target is no more errors, he also hoped programmers would continue “entertaining us with the fun bugs that we love.”

The lecture was entitled “Getting rid of bugs?” and Berry was careful to point out the importance of the question mark, because it suggested the quest to exterminate bugs was a process that would probably go on forever...

Embracing uncertainty

Professor Chris Bishop *promised to make his talk “maths-free” and largely succeeded (smuggling in only a couple of formulae) – in the process managing to illuminate several key mathematical and computational problems, including probabilities, loopy belief propagation, and how to help search engines make lots of money...*

If Professor Berry dealt with certainty, Bishop’s focus was the opposite. Computers carry out their tasks in what can be described as “certain” ways, based on the rules of logic, but everyday information is all about uncertainty.

To illustrate this central point, Bishop used four very different examples: how graphics software learns how to recognise different elements of photos, how to rank the members of a chess club, which ad on a page of search results will be clicked by the user, and how to understand the complex causes of childhood asthma, including genetic and environmental factors.

Many processes may appear random, said Bishop, but it is also possible to make precise predictions about uncertain events by using clever mathematical tools. Bishop's “party trick” to prove this is the Galton Machine – a kind of mathematical Pachinko, using tiny glass beads and a series of pinball-like buffers which steer the beads into a row of glass tubes at the bottom. It is almost impossible to predict where any one bead will go, but if you pour thousands of beads in, they follow a very clear pattern – conforming to the laws of probability according to what is called a “binomial distribution.”

To gauge the probability of repeatable events, there are basically two schools of thought, said Bishop: frequentist and Bayesian. Frequentist probabilities are used for problems such as predicting the flip of a coin, because it deals with the “limit of an infinite number of trials,” so that after a few million flips, you are likely to see heads and tails close to 50:50. But if you wish to describe the uncertainty associated with a single non-repeatable event, the problem gets harder. This is where Bayesian probabilities come to the rescue. They describe the prior probability of an event, and then update this probability in the light of new relevant data. This can be viewed as learning from experience. There has been something of a “religious war” in statistics between these two approaches, said Bishop, but he operates primarily in the Bayesian camp, using Bayesian methods to develop new commercial applications.

For example, companies try to learn what people like and dislike, using a matrix of ratings – e.g. of movies – to anticipate what they are likely to buy. By comparing someone’s preferences with those of a large population of other people, it is possible to identify similar patterns of preferences in others, and hence to make predictions of what other items that person will like. Bishop explained that the system used to measure likes and dislikes does not need to “understand” the products involved. It simply bases its analysis on data from millions of other consumers to recognise patterns.

Photographs may not seem good examples of uncertainty, but Bishop proved otherwise by showing how a graphics program uses Bayesian probabilities to cut and paste a human figure from one landscape to another – using a combination of colour information together with “prior knowledge” about the structure of natural images. “There is more to the world than just data,” he said. Colour information would not be enough on its own to recognise and isolate the figure, he explained, but the system can combine the colour information with the prior knowledge to determine the boundaries of the figure and hence remove the background, thereby allowing the figure to be placed into a new scene.

The data revolution, he continued, is gaining momentum. In 2007, the world's computers stored an estimated 280 exabytes of data (280 billion gigabytes), and this amount is doubling every 18 months. Between now and March 2012, we will have created as much information as the sum total of all data gathered from the dawn of humanity up to the present day. But there is more to the data revolution than just the sheer quantity of data. There is a transformation from desktop computers to cloud computing, from applications to services, from isolated data to the fusion of multiple diverse data sources, and from hand-crafted solutions to solutions that are learned (by machines).

To exploit this opportunity requires the invention of new tools to process and store all the data, including new machine intelligence based on Bayesian learning. The other key elements include probabilistic graphical models, which factorise probabilities into simpler sub-sets, and efficient methods for inference, which allow predictions based on probabilities to be updated quickly once new data is observed.

Bishop was driving at one of the critical problems faced by computing today – the fact that we don't have enough computer power to solve all the problems that arise. In other words, there's too much data and too many tasks to perform, and current methods and systems don't scale, despite the use of clever methods like “Monte Carlo sampling.” We need to get smarter – and this means using strange-sounding methods such as “loopy belief propagation.”

Bishop then focused on practical problems to illustrate some of the methods required. First, he discussed the problem of ranking members of a chess club – a “noisy” problem because the best players don't always win. To measure the probability of one player beating another, various factors have to be taken into account, assigning different strengths to different players then adjusting the rankings based on the results of games, using Bayesian inference methods such as “expectation propagation.” Microsoft applies these methods to ranking and matchmaking amongst the 24 million users of the Xbox 360 games console, achieving what Bishop described as the world's first “planet-scale application of Bayesian methods,” with Bayesian inference running in real time, all day, every day, processing millions of games results.

Turning to search engines, Bishop discussed adPredictor – a system used to predict where someone visiting a web site will click on the page. This is important, because although search engines may seem to be free to the user, web search is funded by advertisers who pay for every click on their ads. The key problem, said Bishop, is how to measure the potential value of an ad (determined by the probability that someone will click on it) by weighing all the relevant factors involved, collecting billions of pieces of data by counting clicks and non-clicks on specific locations, then quantifying the uncertainty in the probability of being clicked by using Bayesian inference.

Finally, Bishop briefly touched upon his work in the study of childhood asthma in collaboration with the University of Manchester. He is applying Bayesian methods to a large data set collected from 1,000 children over an 11-year period, combining environmental with genetic information, and trying to understand which factors are relevant for determining the onset of asthma, and how these factors interact with each other – a fascinating example of the widespread benefits which can follow from *embracing uncertainty*.

Opinions expressed here do not necessarily represent the views of the RSE, nor of its Fellows