

# The Ring

THE JOURNAL OF THE CAMBRIDGE COMPUTER LAB RING

Issue XXXVI — May 2014

Who's Who	2	Sirish Reddi	3
Hall of Fame news	10	<i>'Political Siri' to reach the world's largest electorate</i>	
Computer Laboratory news	12		
		Espruino	4
		<i>The first JavaScript Microcontroller jump started by Kickstarter</i>	
		Sunil Shah	6
		<i>Graduate Story</i>	
		Evolve a pet	8
		<i>Part IB Group Projects</i>	
		Certifying Algorithms:	11
		Designing Convincing Algorithms	

# Who's Who

**Toby Austin** (JE MA05) has started another two companies, Bantr and Beauhurst. Bantr is a social experience for football fans that lets them follow live scores and commentary, share their opinions and vote on match events all in real-time. Beauhurst provides deep data on thousands of growing companies, the deals they are doing and the investors that are backing them.

**Jonathan Ayres** (R MA92), CFO at C Hoare & Co, has been appointed Chair of BBA Small Banks Panel.

**Harry Barman** (K PhD86) is working at JP Morgan where he is an Executive Director.

**Dave Budenberg** (JN BA83) is a Senior Software Developer at Symplectic Limited.

**David Cleevely** (PhD82) has rejoined the Board of Cambridge Network.

**Peter Cowley** (F MA77) has been appointed Fellow in Entrepreneurship at the Judge Business School, Cambridge.

**Geoffrey Cross** (CHU MEng96) is Head of Data Quality at a UK Hedge Fund.

**Steven Davidson** (G BA98 MSci 99) is a Back End Developer at TrialReach, a search engine for clinical trials, covering all medical conditions from all over the world.

**Tony Gould** (F BA91) is taking a sabbatical and has recently started Master of Mathematics at the Open University.

**Matt Grimwade** (TH BA98) is working as a quant developer in the City of London.

**Amir Hajizamani** (JN BA11) is a software developer at Rangespan.

**Robin Harrison** (R BA03) is a software developer at CF Partners (UK) LLP.

**Colin Howe** (Q BA06) is CTO at Conver-social.

**Wenjun Hu** (T BA03 PhD07) is Assistant Professor at Yale University.

**Tom Jacques** (CHR MA98 Dip99) is working at Julius Baer in London.

**Murray Williams** (MA98) is working at i-nexus where he is senior software architect.

**Assel Zhiyenbayeva** (F BA10) is Group Head of IT Security at Eurasian Natural Resources Corporation Limited.

**Hui Li** (G BA02) is working for Merrill Lynch in Hong Kong.

**Ian McDowall** (CTH MA81) is Tech Lead Manager at Google.

**Joel Moss** (F BA13) is an analyst at Planlogic Partners LLP.

**Amir Nathoo** (JN MA02) is a product manager at Square in the Bay Area, San Francisco.

**Radek Nespor** (CHU MA06) is working at Daiwa Capital Markets in London.

**Christian Richardt** (CAI BA07 PhD12) is a postdoctoral researcher at the Max Planck Institute for Informatics.

**Matthew Rowen** (CC MA04) has joined Bromium.

**Mark Stringer** (RA04) is a senior consultant at Sopra Group.

**Bjarne Stroustrup** (CHU PhD79) is now a Managing Director at Morgan Stanley in New York City. He is also a Visiting Professor at Columbia University and a Research Distinguished Professor at Texas A&M University.

**James Sutherland** (JN BA02) is doing a PhD at Abertay University in Scotland.

**Christopher Sutton** (CAI BA05) has recently started a monthly ear training column for Total Guitar Magazine. The series covers interval ear training for modern guitar players, and is included in both the print and digital editions of the magazine.

---

*If you have some news please email it to [cam-ring@cl.cam.ac.uk](mailto:cam-ring@cl.cam.ac.uk)*

*The Ring*, Issue XXXVI May 2014

Web: <http://www.cl.cam.ac.uk/ring/>

E-mail: [cam-ring@cl.cam.ac.uk](mailto:cam-ring@cl.cam.ac.uk)

Tel: +44 1223 763585

Post: William Gates Building, Cambridge CB3 0FD

Published three times a year. Copy deadline for the September 2014 issue is August 5th 2014. All content is copyright ©The Cambridge Computer Lab Ring 2014 unless otherwise noted.

*The Ring* is the journal of the Computer Lab Ring, which is the graduate association of the University of Cambridge Computer Laboratory.

# Sirish Reddi

## *‘Political Siri’ to reach the world’s largest electorate using Voxta’s Hindi speech recognition*

“Mr Modi is talking back to me...I said ‘mehangai’ (inflation), he understood, and is telling me what he will do about it”. The look of amazement on the Indian auto driver’s face was enough reward, well almost, for our team’s 5 years of work on one of the hardest technology problems around — speech recognition for English and Indian languages.

After launching our speech recognition line, dubbed the political ‘Siri’, in Hindi for callers to listen to Modi’s views on different issues, we went and observed a few callers to see how they liked or disliked the experience. To see our Hindi speech recognition used in the context of the world’s largest election — a first for Voxta and the world — was very satisfying.



We founded Voxta in 2009 with the objective of giving access to dynamic information to the hundreds of millions of Indians without internet access. Mobile penetration is above 800m but few have any data access to the millions of websites and thousands of API’s that exist. So the voice channel and speech recognition for access to dynamic information was a natural progression. But to launch voice access to thousands of API’s, a contextual dialog manager a la Siri, but better, is another key component. Statistical, rule based or something else....all became key design and business choices.

But as most people in computer science know, and we did not at the beginning, is that speech recognition was not, and is not, a solved problem. After reading Engineering & Law at St. John’s in the 1980’s, I had put Engineering firmly behind me, unable to see it’s value in the UK of the 1980’s. It is now central to my existence!

Dealing with accents, dialects and varying quality of mobile networks is an enormous engineering challenge — something that even Google has side stepped by only targeting users with smartphone applications. So we have had a long haul, and an increasing focus on data, more data, even more data and ...algorithms. The team and many advisors, formally engaged and informal, have been critical to help us identify and address these issues. But the biggest personal challenge, was moving to Hyderabad from London with family, even with a family home here. As an experienced entrepreneur, setting up a new venture, was relatively speaking, a breeze!

However UI and understanding how users interact with technology is the guard dog that constantly bites us, and brings us back to reality. A smartphone user in London is very comfortable using devices to navigate around concepts and information. In India, many lack formal education and cannot save contacts on phones — some are even uncomfortable using numeric keypads. However they are very focused on anything that improves their lives.

We have several other services including language testing and training, customer service applications, surveys and interactive marketing campaigns. In the West, all major (and minor!) decisions are informed through web and mobile applications. For most Indians and a few other billion people around the world, currently not online, voice recognition can play the role of dial up internet access in the nineties. For instance services for job seekers or dating would be very interesting, but require scale. So having developed a product that is capable of supporting thousands of concurrent phone calls and decent speech recognition, we are keenly looking for partners to work with us on pioneering services that have impact — whether in India, the UK or US.

# Espruino

Gordon Williams needed £10k to get the party started but raised 10x that amount for his open source project Espruino.

*TR: Gordon, you worked as a software engineer for a few years after graduation. What made you take the decision to run your own company?*

GW: Like a lot of Compscis, I'd done odd bits of work even before I started my degree. This continued while I was studying, and some of the software I'd made sold well enough to allow me to finish University without any debt.

After university I got a job, real life intervened and I ended up ignoring the software. I always regretted not following it up, and about 18 months later I started contracting. Suddenly I found I had a company, more money and more free time, and was able to start developing my own software again.

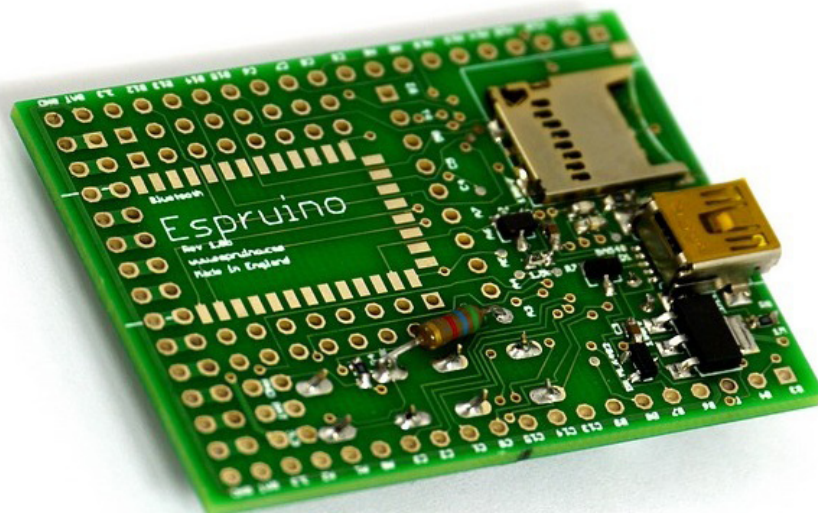
The decision to focus on a product was really because of the companies I came into contact with. Many of them appeared to have really bad products or to make shockingly bad business decisions — and yet they still seemed to survive and in some cases even to do quite well. It made me think “how hard can it be?”

I spent a few years putting my own projects second to consultancy work. However, finally I had a really bad year. One of my main clients went bust, and another wasn't paid by their customer and so refused to pay me. I ended up losing a lot of money, but it gave me the motivation I needed to focus on my own things!

*TR: Espruino is the company's second product. Can you tell me about it and what prompted its development?*

GW: Espruino is a JavaScript interpreter that runs on cheap, single-chip microcontrollers. You just plug the board in via USB and can program it with nothing more than a serial terminal (although there's a shiny web-based IDE too). It's designed to make microcontroller development significantly easier, faster, and a lot more fun.

I'd been programming microcontrollers for fun for a while, but as soon as I moved past writing PIC assembly, the (free) tools were useless,



especially for ARM. I bought several development boards and found the same thing over and over — it'd take two or three hours to enter my details, download 500mb of drivers and crippled software, install, register, wade through useless documentation, set up paths, and finally find, compile and load the blinky LED example. By that point I was so disheartened I'd just put it back in its box and forget about it — and that was on Windows — if you had Linux or a Mac the situation was far worse.

It seemed that there must be a better way and I thought about putting all the tools on the device itself. However, the only self-contained solutions still seemed to use BASIC or languages that most developers would never have heard of.

Given the popularity of web apps and the rise of node.js, JavaScript seemed the obvious choice. The way it encourages you to write your code as a series of functions that are called when things happen means that you get cooperative multitasking and great power saving for free.

*TR: Why is Espruino different from other available JavaScript-capable devices?*

GW: Espruino has significantly lower memory requirements, so it can run on a single chip without external RAM or Flash. For instance Tessel has 32MB of RAM, and realistically Node.js with Linux will need 64MB or more. Espruino can run in just 8kB.

That's got a few really big benefits: The chips can cost as little as £1.50, they're packed full of peripherals (USARTS, I2C, SPI, timers, RTC, etc) and they're extremely low power, drawing just microamps when sleeping. Because you don't have to attach memory, you can use small, simple packages, and every single pin is still available for use by the developer.

*TR: Why did you decide to go via the Kickstarter route?*

GW: I actually tried a freemium route first. I provided the Espruino software for a variety of cheap development boards, and if you wanted to save your code to the chip's internal flash you'd have to pay £3 to register your board.

It was a complete disaster. I'd tried very hard to raise awareness, but over the entire year I made about £50, and for that I had a massive amount of support to do.

I contacted many board manufacturers about including it on their boards, but didn't get much interest. They would have used it if it was free, but weren't interested in paying anything towards supporting the software's development.

So developing my own board seemed the only thing left to try, and for that kind of thing, Kickstarter was perfect.

*TR: How much did you need to raise and how much did you raise?*

GW: I couldn't have had many less than 500 boards made in one go, so I absolutely needed £10,000. I put the funding level at £20,000 as I thought below that, I should just go and get a proper job again.

In the end, I raised £100,710 from 1692 backers.

*TR: What did this allow you to do?*

GW: A few things I guess. It's definitely given me some time to find a way to get an ongoing income from Espruino without giving up equity. It's given me a lot of publicity, and it's shown to distributors and users that Espruino is something people want. Probably most importantly it's almost instantly built a community who are passionate about Espruino, and because they supported it right at the start, they have a real interest in seeing it succeed.

*TR: Would you recommend the Kickstarter route to others?*

GW: Absolutely! It's a fantastic platform, and at the moment it seems to be basically risk free. The biggest issue for Kickstarter projects right now seems to be that they haven't budgeted for producing in far higher volumes than they'd expected!

Having said that, Kickstarter is quite a social platform and Kickstarter supporters want to feel part of the process. That'll mean you get an insane amount of email, and if you're doing it on your own then you probably won't get much time for anything else, both during the Kickstarter, and immediately afterwards!

*TR: What challenges do you now face and what are the plans for Espruino?*

GW: The main challenge now will be trying to keep the momentum going, keeping people talking about (and buying!) Espruino. I'll be getting some more distributors, improving the website and drumming up a bit more publicity, but also pushing Espruino into some new areas, like Education and IoT.

Finally, later this year I'll probably be heading to Kickstarter again, with a smaller, cheaper, more efficient Espruino board, and a much better video!

# Graduate Story

## Migrating to the Heartland of Technology: **Sunil Shah's** journey to California

At Cambridge, I was far from a model student and so my graduate school aspirations didn't really take hold until I wrote up my dissertation (under the supervision of Nishanth Sastry) during Easter of my final year. Entitled 'Distributed Twitter', it actually had little to do with Twitter and instead was some dataset analysis and simulation involving various delay tolerant networking routing algorithms.

However, on graduation I had little motivation to go straight into further studies so, having made extensive travel plans and used up my savings, I decided to enter the world of work for a few years.

After spending a year volunteering with One Laptop Per Child in India and cycling across Africa in the world's longest bicycle race, the Tour D'Afrique, I started as a graduate technology analyst at Barclays Capital. At the mercy of human resources, I found myself in a team not only antithetical to my technical interests but also to my interest in investment banking and level of motivation. So, I quickly booked a seat for the Graduate Record Examination (GRE), the graduate version of the SAT that is required for university applications, and began down the path to studying in the US.

For personal reasons, I stayed at Barclays until the end of the graduate scheme and then joined music recommendations company Last.fm, working as a software engineer looking after their catalogue systems and performing the odd data mining task.

While at Last.fm I applied to graduate schools in the US, first in 2011 and then again — when my first applications were unsuccessful — in 2012<sup>1</sup>. In the interest of brevity, suffice to say that the graduate school acceptance rate in the US is, for one reason or another, an order of magnitude lower than for the top universities in the UK. It's not enough to have achieved a first class degree. You also need to demonstrate an aptitude for professional/entrepreneurial competence for Masters programmes and it's extremely rare for undergraduates to get into any PhD program without some research experience already under their belt. There are more details about how to put in a competitive application on my blog <http://www.geekonabicycle.co.uk/blog/1029/> ).

I was accepted by six of the 12 schools to which I applied and eventually picked UC Berkeley (commonly known as Cal) for their Master of Engineering Program. Berkeley offers a traditional Master of Science degree in Computer Science but, presumably for funding reasons, takes very few students for this (I know just one). The Master of Engineering is a one year professional degree which is mostly technical and includes a team capstone design project and two engineering leadership courses — essentially a mini MBA.



The structure of postgraduate education in the US differs markedly from that in the UK. A PhD program in the US is typically at least five years and most students take six years to complete. Students pick up a Master of Science degree in the first two years, and during these first two years they will take a significant number of courses related to their

intended area of research. Typically funding for most PhD students will be contingent on working as a teaching or research assistant for at least a couple of years.

In the short time I've been at Cal, I've taken courses in Advanced Robotics, Computer Vision, Parallel Computing and Machine Learning. The way students are assessed is also different. Instead of pinning everything on a final exam, grades are split evenly between examinations, final projects and coursework.

Coursework tends to occupy most of my time. In addition to three hours of lectures per week, typically there'll be 20 hours homework per class every two weeks. However, homework can often take longer — one robotics homework took me 50 hours and I handed it in 70% complete! That said, such homework allows students to more fully understand a subject through practice, very valuable in a subject like computer science.

The capstone project that I've been working on is (unfortunately) titled "Drones: The Killer App". With the hype surrounding the commercial use of unmanned aerial vehicles (UAVs), I've been working with a UAV research group here to look at commercial applications of the technology. So far, we've built a vision based automated landing algorithm that runs on commodity hardware and open source software.

The entrepreneurial culture is very well and alive on campus. While Berkeley lags behind Stanford, they're making a very conscious effort to promote and support entrepreneurial endeavours through incubators, mentorship, networking events and hackathons. Since our program sits in between the traditional engineering graduate programs and the MBA program, we often mix with MBA students. I've lost track of how many times an excitable business school student has told me about their healthcare app idea!

The other significant activity that kept me busy last semester was searching for a job. One of the principal reasons that I (and most other international students) have spent a significant amount on US tuition fees is for the considerable employment opportunities available. I had some idea of where I wanted to work and applied to 30 firms. I was interviewed at 14 of these and received offers from 6. Demand for engineers also means that salaries are high and, at my level of experience, better than in London. In June, I will start as an engineer at Mesosphere, a startup that aims to be to the Apache Mesos project what Cloudera was to Apache Hadoop. Life is easy for computer scientists here!

If you're looking to apply to graduate school in the US and want some advice, please do send me an email! Similarly, if you're a Cantab in the Bay Area, get in touch. We're in the process of setting up a Bay Area Ringlet.

*Sunil Shah can be contacted at [sunil.shah@cantab.net](mailto:sunil.shah@cantab.net)*

*Blog: <http://www.geekonabicycle.co.uk>*

# Part IB Group Projects

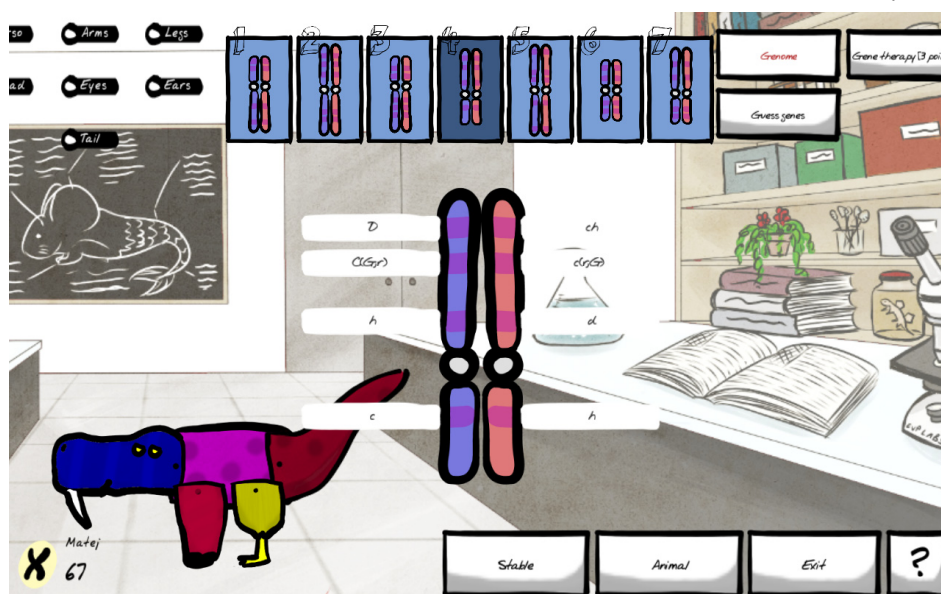
## Nikolaus Blow, project coordinator for Team Bravo, reports

Our project, *Evolve a pet*, is a game designed for students of an A-Level or GCSE level to learn about genetics within the context of breeding animals together. As the project leader, I was immensely grateful to have received this project. I had previously worked on a bio-informatics project, and am incredibly interested in biology, so I was very enthused to be working on this. Luckily for me, my whole team was as enthusiastic and keen as I was about it, and we sketched out a design for the whole game almost as soon as we had been told what our project was.

Initially, we worked on getting the whole behind the scenes breeding mechanism working. We decided on a system in which the animal would have several body parts, and each of those body parts would have certain traits associated with them. These traits would be coded as genes on a particular chromosome, and upon breeding, the offspring would receive chromosomes from both parents which would then be fed into an algorithm to determine what the gene should express itself as.

Our first —and naive — thoughts were that once we could create an animal object that could breed with another animal, and could produce an offspring that contained traits that were a mixture of both parents, we were pretty much done. We did some basic work on the interface, but did not think that we would need to spend too much time on it. This was completely wrong. Yes; we could have made a text based game within a few days that printed out statements such as “your animal is red”, but where would be the fun in that? This was meant to be a game after all!

Luckily, one of our team had the foresight to suggest a game engine, namely Unity, to create the game. This was great for many reasons. Firstly, it made things such as animation and dealing with graphics much simpler, which helped a lot as obviously a static animal does not feel anywhere near as lifelike as an animal that blinks and bounces about. Secondly, it meant we could easily implement networking code, and we had a working demo of breeding an animal with a friend over the local area network within a few days. Thirdly, and possibly most



importantly, when using an engine that has been developed and maintained and updated for several years, we could fairly safely assume that for our purposes any bugs would be on our part, and not to do with Unity.

Around halfway through the project's allocated time, we had pretty much completed and thoroughly tested all the back end coding, but the front end looked remarkably bare. This was when when we started to panic. We realised that putting everything together would take an incredible effort. Even things such as making buttons look good took up a lot of our valuable time. What was worse though was that we realised that we still had design decisions to make at this late stage — things we did not even think about such as button placement or how to navigate between scenes. We had just assumed that things like this would just fall into place, but we soon realised that this was going to be a lot of work.

However, through a superhuman effort on the part of everyone, and several afternoons spent together in the Computer Lab, we managed to get a working interface up and running. Even better, we managed to get some high quality background art, which really transformed what we had from looking like a tech demo into an actual game. In fact, I feel like the art style and the graphics really are the thing which tied everything together, and it almost crept up on us without us noticing it. Of course we realised the importance of creating an aesthetic — if you don't relate to your animal, why would you ever want to breed one — but we never really actively considered an art style and it almost organically evolved into what you see before you.

Working on the project was a great experience, and I could not have wished for a better team.

---

*Team Bravo (Emilian–Dragu Bacila (CTH), Nikolaus Blow (G), Mary Dai (CL), Marianne Gosciniak (CHU), Matej Hamas (R) and Thomas Lefley (G)) won the prize for 'Professional Result'.*

*Further information about the Part IB group projects can be found at <http://www.cl.cam.ac.uk/teaching/group-projects/>*

# Hall of fame news

## ARM

ARM has been named one of the UK's best employers at the Top Employers awards.

The Top Employers awards recognise excellence in people practices, based on independent research conducted by the Corporate Research Foundation.

## Bango

Bango has launched Direct Operator Billing for the BlackBerry World storefront with the mobile network operator STC in the Kingdom of Saudi Arabia. Statista ranked Saudi Arabia the country with the third highest smartphone penetration, behind only the UAE and South Korea. The partnership will allow BlackBerry smartphone customers to easily purchase apps and digital content without the use of premium SMS messages or the limitations of credit cards.

Bango is also well placed to capture the massive Asian smartphone market having opened an office in Singapore, ranked fourth in smartphone penetration.

## Bromium

Bromium has been nominated for a SC Magazine Europe Innovation Award.

The Awards honour professionals working to secure enterprises of all sizes and the vendors that deliver innovative security technologies.

## Cambridge Broadband Networks

CBNL been named as a Red Herring Top 100 Europe award winner — a prestigious list honouring the year's most promising private technology ventures from the European business region.

CBNL, which provides highly efficient mobile backhaul and enterprise access infrastructure for the world's largest service

providers, was selected by Red Herring based on a strong track record and a high potential for growth.

## Fusepump

WPP's wholly-owned operating company Wunderman, the world's largest digital agency network, has acquired FusePump Limited. Fusepump specialises in product data and content for digital marketing.

## Linguamatics

Linguamatics has been named as a winner of the prestigious Queen's Award for Enterprise.

Linguamatics has received the award for International Trade in recognition of outstanding growth in export sales across USA and Europe for its natural language processing (NLP)-based text analytics software platform. Over the past five years Linguamatics has achieved growth in excess of 300 percent in overseas sales and staff growth of almost 200 percent.

The company continues to grow and attract new customers. This year, Lngumatics will increase its work force by over 25 percent, in both Europe and the US.

## Masabi

Thames Clippers has signed a five year contract with Masabi, the leader in mobile ticketing and payments for transit, to bring smartphone ticketing to its River Bus service operating on the River Thames.

Thames Clippers is currently seeing strong growth in passenger numbers. Mobile ticketing will aid this growth further by offering customers a user-friendly way to pay as well as adding a significant sales capacity without the need to invest further in infrastructure such as ticket vending machines.

## RealVNC

deskhop® , RealVNC's computer screen sharing Facebook application, is now available as a free app for iPhone and iPad from the App Store.

Designed to be used by anyone with a Facebook account (about a billion people), the app is an effective tool for the technical wizard who supports friends and family with IT issues.

Powered by VNC, deskhop works by prompting two people wanting to share screens to temporarily run a small software program on their devices. Once established the encrypted connection allows one user to control the other's screen. When one party disconnects, temporary software programs are automatically deleted from computers, leaving no trace. The app ensures that the screen owner always has priority control of the mouse, and can therefore disconnect from the other person at any time.

## Swiftkey

PR Week magazine has named SwiftKey the winner of their Best Places to Work award (in-house team category).

## 1248 Ltd

Start-up 1248 Ltd has received a £250k investment from experienced serial telco entrepreneur Rob Dobson.

1248 provides expertise and services to the rapidly emerging Internet of Things (IoT) industry. The company is already playing a leading role in creating an open standard to enable IoT applications and services to work together automatically, as part of a project funded by the UK government's Technology Strategy Board.

# Certifying Algorithms: Designing Convincing Algorithms

Pengming Wang

November 11, 2013

Programming is hard. Translating an algorithm from theory into a bug-free, working program remains a difficult task even for the most experienced programmers. For sufficiently complex algorithms an implementation will almost always contain initial bugs that compromise its correctness. In a time where computer systems control increasingly critical parts of our lives, e.g. security, navigation, or traffic control systems, ensuring the correctness of these systems becomes vital. Most of the research in software engineering or verification focuses on techniques to avoid bugs during the implementation, or to find them afterwards. Here, we argue for an alternative, complementary approach. Instead of tackling the problem of bugs during or after, we propose to deal with them *before* implementation, i.e. when designing the algorithm. The idea is to design algorithms in a way that one can easily verify the correctness of their output.

The concept of checking a result is not new, and is even taught in primary schools. When introducing the division operation to students, teachers often suggest to check the result by performing the inverse operation. When obtaining the result  $y$  for the division  $a/b = y$ , one can verify the result by checking whether  $b \cdot y = a$  holds or not. Since students are more familiar with multiplication, the check is easy to perform, and a successful check ensures the correctness of the result. The same principle can be generalised and applied for algorithms. We say an algorithm is *certifying* if, in addition to its normal output  $y$  for an input  $x$ , it also produces a *certificate*  $c$ , such that it is easily checkable given  $x$ ,  $y$ , and  $c$ , that  $y$  is a correct output for  $x$ . A simple algorithmic example is testing a graph for bipartiteness. Given an undirected graph  $G$ , a certifying algorithm would not only output yes, bipartite or no, not bipartite, but also provide a two-colouring of  $G$  along with a yes-answer, or an odd-length cycle in  $G$  along with a no-answer. From basic graph theory we know that any graph is bipartite if and only if it admits a two-colouring, and any graph with an odd-length cycle is necessarily non-bipartite. Checking that the certificate is actually a two-colouring or odd-length cycle then suffices to convince us that the output was correct. Thus outputs that pass the check can be trusted even if the actual implementation contains bugs. In [1], McConnell et al. show that many classical algorithms can be modified to a certifying version.

While computer scientists traditionally only care about correctness and efficiency when designing algorithms, we want to encourage the reader to also think about how algorithms can be made certifying. Not only does this lead to more reliable programs, but it might also inspire different (and more efficient) ways to solve the actual problem, as for instance in [2] for the case of algebraic root isolation.

## References

- [1] R. M. McConnell, K. Mehlhorn, S. Näher, and P. Schweitzer. Certifying Algorithms. *Computer Science Review*, 5(2):119–161, 2011.
- [2] K. Mehlhorn, M. Sagraloff, and P. Wang. From approximate factorization to root isolation. In *Proceedings of the 38th international symposium on International symposium on symbolic and algebraic computation*, ISSAC '13, pages 283–290, New York, NY, USA, 2013. ACM.

# Computer Laboratory news

## The Research Skills course

In 2009, the Computer Laboratory set up its MPhil in Advanced Computer Science. The MPhil has just one compulsory module: Research Skills. This module aims to teach the range of skills required for a successful research career: critical reading, summarisation, and review of research papers; writing of technical documents and research papers; presenting research findings in seminars and at conferences; and the design and analysis of experiments. In addition to MPhil students, the module is also taken by all first-year PhD students who have not already had such training and is an optional module for fourth-year undergraduates on the MEng course.

One of the 2013-14 exercises was to write a 4–500 word essay on a research topic, explaining it to an audience of computer science graduates such as read ‘The Ring’. The best three essays are being published in this (see page 11) and the next two editions of The Ring.

## Computer Laboratory hosts 1st Oxbridge Women in Computer Science Conference

The 1st Oxbridge Women in Computer Science Conference was held at the Computer Laboratory on the 27th February 2014. The Oxbridge Women in Computer Science Conference, an annual conference co-organised by Dr Mateja Jamnik, Director of `women@CL`, and OxWoCS, brings together junior and senior female computer scientists at both universities, with the aim of encouraging collaboration through formal and informal discussion.

The conference consisted of student presentations and poster sessions with feedback given by a panel of senior female academics from Oxford and Cambridge Universities as well as technologists. The keynote was given by Professor Ursula Martin from Oxford University.

The event was a great success with over 50 attendees. Said Professor Martin ‘This was a terrific meeting and a great opportunity to meet the smart, energetic and enthusiastic women in computer science at Oxford and Cambridge.’

The 2nd Oxbridge Women in Computer Science Conference will be held at Oxford in February 2015.

