flossUK

news@UK

The newsletter of FLOSS UK, the new name for the UK's oldest Open Systems User Group, UKUUG

Published electronically at `http://www.flossuk.org/Newsletter`

Volume 21, Number 2 **ISSN 0965-9412** **June 2012**

Contents

From the Secretariat

Jane Morrison

I am pleased to advise that the Spring Conference held on 20th - 22nd March in Edinburgh was very well attended (in fact we had to close bookings the week before as it was full!). The feedback forms indicate that delegates very much enjoyed the event. See write-up in this issue.

Let's hope we can repeat the success again in 2013 when we are hoping to hold the event in Newcastle. The call for papers should be available shortly.

Since then we have held two Tutorials in London: 'Google App Engine' by Paul Barry and 'Presentation Skills' by Damian Conway. There is a description of the latter event by Ian Norton in this issue.

Despite correspondence etc. with other potential tutors I am afraid, that at the time of writing, we have nothing as yet confirmed tutorial wise for the rest of the year. As soon as we do have any confirmed we shall of course circulate details to members via email.

We continue to work closely with Josette Garcia of O'Reilly to bring you various tutorials.

Other events which are confirmed include the AGM on 20th September and the London Un-conference on 27th October.

SuSE have renewed their Gold Sponsoring membership with us for a further 12 months. The Sponsoring memberships do greatly assist us and allow us to keep event fees as low as possible.

Full details, Agenda etc. for the AGM will be sent to you in August. If you are interested in joining Council please let me know.

Don't forget Inland Revenue allows our members to claim their subscription amounts for tax allowance purposes – see further details on our web site.

The next Newsletter will appear in September and has a copy date of 17th August. Any interesting articles from members will be very welcome – all submissions should be sent to: `newsletter@ukuug.org`.

Chairman's Report

Paul Waring

Spring 2012

Our Spring conference in Edinburgh was a popular event, with a packed programme covering topics across all aspects of system administration, a fully-booked tutorial on Chef and a variety of opportunities to socialise and network. Many thanks to everyone involved - particularly the speakers and local organisers - for making the event a success.

Council is also pleased to announce that a surplus in the conference budget enabled us to make a donation of £1024 to University of Edinburgh Informatics Student Support Fund, which provides financial support to students who would otherwise struggle to afford to study.

Where speakers have provided slides we have made these available online on the conference website: `http://www.flossuk.org/spring2012`

Plans are already underway for the 2013 conference. There are lots of ways to help out, from chairing sessions to preparing the timetable, so please let us know if you would be interested in getting involved.

Open Standards Consultation

The UK Government is currently running a consultation on the subject of open standards across

government departments until Monday 4th June 2012:
`http://consultation.cabinetoffice.gov.uk/openstandards/`

We would like to encourage members to respond as individuals if possible. A greater number and variety of responses will lend weight to the argument in favour of truly open standards across government.

For members who would like further information on how to respond to the consultation, we have placed a document from Open Forum Europe on our website which may be helpful:
`http://www.flossuk.org/OpenStandardsConsultation`

Future of FLOSS UK

Paul Waring

Since the beginning of this year, we have been using the new FLOSS UK brand for all our events and publicity (with the exception of finances, which remain as UKUUG Ltd). Although the ukuug.org website remains as a permanent archive of past events, all new content is being placed on the new wiki at `www.flossuk.org`. This new structure allows a wider group to add and maintain the content, whereas previously the website could only be changed by a handful of individuals.

Externally, the new name has been well-received. Several companies contacted us requesting to sponsor Spring 2012 who had not been aware of the organisation or what it stood for before the name change, whilst existing sponsors have continued to support us. We also continue to be represented on the BSI IST/41 committee, with Richard Melville handing over to Simon Phipps earlier this year.

Looking to the future, we are keen to support events run by small groups or international organisations which do not have an established UK presence, as well as running our usual Spring conference. In particular, the fact that FLOSS UK has server space, mailing lists, a bank account and a staffed office means that we can help with the infrastructure and logistics, whilst the conference organisers concentrate on speakers and publicity. We have already organised one such event this year, by hosting Puppet Camp alongside our Spring conference, and we are in active discussions with other groups who are interested in working with us to put on events.

Spring Conference – Edinburgh

Roger Whittaker

The Spring Conference this year was held in Edinburgh on the 21st and 22nd March. As usual, a tutorial was held the day before the conference (this time it was a tutorial by Brad Knowles on Chef entitled “Just Enough Chef Recursive Bootstrap”). And the day after the conference the one-day “Puppet Camp 2012” was held. I was only present for the two days of the conference proper.

The Informatics Forum at Edinburgh University made an excellent venue for the event. There were two adjacent lecture rooms for the talks and a large, light and airy space outside them for coffee, and exhibitors. The conference was fully booked and there was a good, friendly atmosphere throughout.

There were two streams of talks throughout the conference. As always, it was sometimes difficult to choose between the two talks, but in general the scheduling had been done well and the alternative talks did not clash in terms of content.

On the first day I particularly enjoyed Dag Wieers’ interesting talk about creating a documentation workflow that both satisfies “our” criteria of how such a thing should work, and produces outputs of the type that “they” (wearing suits) want to receive. In particular he discussed the use of asciidoc, unoconv, and command-line workflows built from these and other tools capable of producing "company branded" outputs in various formats including MS Word if necessary as well as PDF and HTML.

Also on Wednesday, I enjoyed Stephen Quinney's talk "How to (hopefully) avoid being r00ted" and "SSH Can Do That?" by Smylers. Richard Moore of IBM gave an interesting talk on the challenges of making live guest relocation work in the world of zVM, across different versions of the hardware and virtualization environment. SUSE's Gerald Pfeifer gave an overview of the Open Build Service, SUSE Studio, SUSE Manager and Cloud initiatives.

On Wednesday evening the Conference Dinner was held at "The Caves", an extraordinary venue that is part of the substructure of the South Bridge, one of the high multistory bridges that are such an unusual feature of Edinburgh. The entertaining after-dinner talk with slides and movies was given by Charles Yarnold, known to many from the TV series "Gadget Geeks".

On Thursday Tariq Rashid of the Home Office spoke very openly but also optimistically about the UK Government Open Source Action Plan. It was very good to see this kind of engagement from a government technical person, and there was considerable discussion as a result.

Simon Phipps spoke about the Open Source Initiative and its activities. As always, his talk was amusing and engaging. A particularly nice analogy was his comparison between a corporation and a crocodile (a crocodile keeper he met in Australia said "you can't have a relationship with a crocodile"), the context being the repudiation by a large corporation of verbal promises that had been made in the past by people speaking on its behalf. See: `http://webmink.com/essays/reptiles/`.

Also on Thursday, I enjoyed Dag Wieers' second talk, this time on Relax and Recover (Rear), an interesting solution for bare metal recovery. Matthew Richardson of Edinburgh gave an interesting talk on building a highly available virtualization environment.

Exhibitors in the coffee area included O'Reilly, SUSE, Google, Opengear and Bytemark.

The conference ended with the usual lightning talks session.

All in all, a very successful event, in an excellent venue and one of the nicest possible cities for a conference.

Presentation Skills Tutorial – Monday 16th April 2012

Ian Norton

Presentation Aikido

I don't think I've ever attended a training session that began with the tutor pushing me towards a wall after reassuringly telling me that he "wasn't going to hurt me", but this training session is like no other that I've attended.

Damian Conway, minor deity of the Open Source and Perl world is not like other trainers. A professional presenter, Damian tours the world speaking at conferences and providing training far and wide from his native Australia in what can only be described as a unique and elegant style.

Our martial arts introduction to the day was of course all part of a carefully planned and executed methodology designed to teach us how we too could excel at presenting.

Our tutor better described this as "Presentation Aikido" or a way of harmonising with the flow of the Universe, or in our case our audience.

Sticking with the advice of keeping focus on five to seven points, we learned all about the seven key points of presentation.

- Be competent – only talk about things you know about

- Be passionate – only talk about things you either love or hate
- Be entertaining – it's the key to communication
- Be prepared – most of your work is done before you step in the room
- Be stylish – nobody likes a bad slideshow
- Be engaging – connect with your audience
- Be yourself

For me, the biggest eye opener was when Damian told us that he spends around one hundred hours preparing for each hour he presents. Us mere mortals should probably be trying for twenty to thirty hours though!

If you don't know how to do these things then I recommend keeping an eye out for this course running again, though it might be a while before Damian is back in the UK so watch out for him speaking at a conference near you. . . ;)

FLOSS – it's more than the code: much as you might like to, you can't ignore the politics nor the economics

Gerry Gavigan

I recently attended an event hosted by a US organisation that wanted to alert general counsel and other corporate intellectual property types to the dangers of FLOSS.

Their presentations to a set of conventional minded business types was consistent with a hyperbolic marketing approach I've come to associate with American lobbyists. In summary "reds under your beds" Senator McCarthy would have approved of the message "destruction of your businesses" by software "as used by the "Occupy Wall St" movement.

This low level attrition by companies that do not want to go the same way as Kodak is ongoing. We need to deal with it and not by hurling insults.

The audience were opinion shapers and influencers in the corporate environment, the ones that approve contracts and sign-off finances. We need to get them on side, for at least two reasons, self-interest: maximising the opportunities for gainful employment in something we like being part of, and wider interest: enabling the benefits that FLOSS can bring to society.

In a world where most people think a word processor is called "Word" and a spreadsheet "Excel" I suggest that discussions about examining the source code or better security models though important in themselves are probably misdirected.

If you haven't looked already at the KDE digest what you would see is that the BRICS, Germany, Scandinavia, the USA and Canada are all significant contributors to the project. And while correlation should not be confused with causality, these are certainly the countries currently surviving the economic conditions relatively well.

As everyone is aware this country is going through tough times at the moment, looking both to save money and for economic growth. A healthy FLOSS industry fits the bill for both of these.

Unfortunately it seems difficult to persuade those that should be thinking about this to seriously consider FLOSS (and as the public sector spend about £20 billion-a-year on IT that would have a fairly direct effect on the shape of local industry) nor to recognise the changes to the local value chain that would result.

We've all seen the way money was poured down the drain for NPfIT, buying software developed elsewhere, supporting well paid international consultants project "manage" it all and rack up wasted costs that seem difficult to believe possible.

Interestingly, before the days of NPfIT, the National Health Service was a relatively ardent supporter of FLOSS. It published a good guide to its use that disappeared in mysterious circumstances about the same time that NPfIT was announced.

But it's difficult to keep a good idea down. The NHS is a big place, but for me two customer driven projects have emerged that are flagships for the importance of FLOSS. The general use of FLOSS by Leeds PCT on the one hand and "OpenEyes" (GPL v3 code) led by Moorfields but involving other hospitals on the other. You can read more about in the website but I wanted to pull out this: "By sharing experience, pooling ideas, and distributing development effort, it is expected that the range and capability of OpenEyes will evolve rapidly".

I find it difficult to believe that this effort will not be local effort and so create local opportunities. It's not protectionist or nationalistic and, particularly if other, similar, initiatives emerge it will stimulate local networks of expertise, training and customer skills (think plumbing or hairdressing) involving gaining real knowledge not industry maintenance jockey certificates.

Unfortunately this is unlikely to be a headline grabbing quick win, but to me it contains the basis for the messages we all need to pass on to our MPs.

If you're inclined to look at our website or read our seventy page report on the importance on FLOSS for innovation, growth and jobs you'll discover why I don't think we can leave it to Cabinet Office. It all starts from open standards of course, so please respond to the government's consultation. It closes on 4th June – feel free to get in contact if you want any help or information.

I'll finish where I started. I don't think I was the most welcome attendee to the event. On the downside I seemed to get a little low level goading (in 1968 this was described as kick a policeman in the nads then photograph him reacting), on the up side it failed and I had some interesting engagement with several people in the post event networking. For those that don't know they will take a view based on you rather than your argument.

Thanks to Roger Whitaker for inviting me to write this piece.

Whither the Open Source Consortium?

Gerry Gavigan

The Open Source Consortium (OSC) originated in 2005 at the suggestion of the then head of Socitim [1] as a vendor owned company limited by guarantee created with the simple-to-express objective of being better placed to win public sector contracts by being the lead contractor forming business consortia.

Today, older, wiser and different, although we retain the name OSC, V2.0 is now a membership controlled society bound by a constitution. There are two classes of membership, voting and free to join, non-voting associate. Members of either class have to agree to be bound by the aims and objectives.

Through its constitution OSC is an independent voice and volunteer effort driven promoter of open standards and F/OSS.

We take the GNU four freedoms as a given, but look to build awareness of the economic, social and political aspects of open source software (spending most of our effort on not preaching to the converted).

This is a more difficult task than it might seem. There's a lot of vested interest in status quo out there. I attended two events recently (but could list many more) one a legal seminar the other a customer seminar. While it might not come as a surprise that lawyers suck their teeth like some dodgy builder and suggest that while you **could** use that nasty legally risky open source stuff if you **really** want to... nor that some so called open source solutions vendors try to sell in their special sauce "designed to protect poor little you” from that technically dubious open source software, it is still a surprise to me to see companies with otherwise good reputations (nameless because I don’t want to commit the same offence) spend as much time if not more doing down others in the open source arena rather than focusing on the much bigger pie of converting the rest of the world to the benefits.

Our starting point for all of this is that open standards and open source software are a structural measure to deliver growth in the economy. As we say in a report [2] we launched at a think tank event [3] which we funded:

It would be a mistake to confuse correlation with causation, however it is noticeable that Brazil, Russia, India, China and South Africa (the BRICS countries) all have government policy on royalty free (RF) open standards for software.

RF open standards for software and OSS exhibit all the characteristics required for innovation and growth and enable positive effects on wider public policy including:

- education
- local economies, creating sustainable high skilled jobs
- digital inclusion
- the environment (reduce, re-use, recycle)

If OSS were actively adopted in the mix of software in public sector ICT, it is likely that the equivalent value released would exceed the value of the “Catapult” budget (£200m) provided by the Department for Business Innovation and Skills (BIS) for commercialising the outputs of Britain’s academic research base.

We bootstrap our work, with a few ideas on the stocks, e.g., an open source apprenticeship scheme, open source software summer schools but our ambition is constrained by our resources and the current membership does not have a monopoly on ideas nor delivery mechanisms. The bigger we get the more we can do.

Please join.

[1] `http://www.socitm.net/info/155/about`

[2] `http://www.opensourceconsortium.org/content/view/195/91/`

[3] `http://www.opensourceconsortium.org/content/view/193/1/`

Recruiting call!

Martin Houston

The popular movements that are sweeping the world, Occupy, Anonymous, the Indignatos and the revolutionary movements of the Arab Spring are now treading a path that to someone who has been involved in the free software movement for more than 20 years seems rather familiar.

Fighting against the machinations of ‘evil corporations’ is a fight those in the free software world know only too well.

One of the slogans of the Occupy movement is “Nobody Wins unless Everybody Wins” – a realisation that we can never have the peace needed to concentrate on the job of surviving the end of the age of cheap energy while there is such gross injustice in the world.

In our world, free software can be freely copied. If I share my code with you we both still have it, and in fact the act of sharing often makes the thing shared much better for all the parties doing the sharing. However realists would argue you cannot do that with a cup of rice. Sharing my cup of rice with twenty other people is of no help if everybody starves.

I mention the cup of rice because as I write this a recent news story is that the rice harvest in India has been such a bumper one that there is not enough storage space for all of it and many tons are simply being left to rot in the open air. Harvested, in sacks but being wasted!

Why does it not go to feed the poor, better surely than it just feeding rats? The pathetic excuse was to do so would affect 'the balance of payments'.

I am calling on you, the people who already know how to co-operate with others to develop software for mutual benefit. Spare some time in this time of change to see how your skills and experience could be of help to the people trying to solve the much bigger problem of making the physical world – sharing out those cups of rice, work in the same "It is smart to share" way.

The machinations of corporations over the past decades has brought into being much 'property' of the intellectual kind that really has no business being property.

The ACTA treaty that they recently attempted to foist upon is is a very clear example of this. Yes, we have the motivation to get angry about it because of the way it could be used by 'rights' holder to chill any innovation or use of anything similar to their technology or artistic work without the payment of extortion.

This bully boy tactic would also extend to areas such as drugs and with the excuse of genetic modification to foodstuffs, allowing the multi-nationals to have an even tighter grip on the life or death of the ordinary people.

What is perhaps even worse is the general stifling of free speech and free debate that measures such as ACTA, CISPA and our home grown variants like the Digital Economy Act – rail-roaded through in the last week of the Brown government, and the latest attempts to monitor and censor or Internet for our own good.

As the Internet is the means by which the human race is finally waking up to the truth that the control of the world is by a very small number of very powerful people, is it any wonder they are trying everything they can think of to neuter its power?

It is too late for that, dissent has reached critical mass.

Here is an example that happened here in London. Occupy held a march on 11th May ending with a meeting and teach in on the small piece of land in front of the Bank of England and the Royal Exchange. The Police attempted to break up the meeting on the pretence that it was a nuisance to residents (what residents?). A few dozen protesters decided to peacefully resist, linking arms to thwart police attempts to remove them. There was a few arrests then a stand-off for several hours, watched avidly by a large audience across the Internet until the police just gave up and left. As the City of London is pretty comatose on a weekend the allegation that the people holding a meeting were being a 'nuisance' would not have cut any ice with a jury until Monday morning! Occupiers then left as it was time to sleep and the point had been made. No matter how much those in power wish it to be otherwise enough people have now lost their fear. The cost of just doing nothing and hoping that this is all just an 'economic cycle' is simply too great.

Yes working towards a new system carries risks, but they are much less than the risks of just pretending the problems will fix themselves.

This is where YOU come in – I think that people who have had the experience making this culture work in the realm of software will have a very important part to play. Go find where you will be useful and volunteer today.

You can read the full manifesto of the global freedom movement, which has already been printed in several newspapers, here
`http://www.globalmay.org/blog/item/95-globalmay-manifesto-template-v3.html`

You really should read it, a great deal of thought and debate has gone into its production.

And a summary of what is wrong and what is needed to fix it from a specifically UK perspective can

be found in the letter that the Anonymous sent to David Cameron:
`http://anonflag.com/cameronmessage.html`

The audio version takes about 25 minutes.

Hardware notes

Martin Houston

Firstly I saw a Raspberry Pi 'in the flesh' last night at the Chemer LUG meeting in Chelmsford.

There are some 360,000 advance orders for this little computer. By this years 'quad core' mobile phone standards it is slow at a single core, 700MHz Arm but considering this is teamed with a pretty good GPU it results in a machine that is quite fast enough for a huge range of applications.

The LXDE desktop is lightweight enough to be responsive so once the cased versions become available this would be suitable for every family in the land.

Got a TV?

Well for about 50 quid you can turn it into a proper computer that can do anything you want it to by putting software on a memory card and booting it up.

The GPU means that reasonable video games will be possible. The availability of millions of such modest capability computers will be a great counter to code bloat that besets Windows and then sadly affects the 'full fat' Linux distributions that feel that they have to emulate it.

The Pi is economic both in purchase cost and running cost, both of which are important in these cash strapped times.

Demand is so high there is a waiting list and you are at the moment limited to one per customer. If you want to join that list go to
`http://uk.rs-online.com/web/generalDisplay.html?id=raspberrypi`

The Pi is a vital tool at equipping everyone with both internet access and a general purpose computing facility that can not just be taken away from them again at a whim. It is time everybody understood about Linux and the Free Software philosophy and the Pi 'Revolution' could very well be the way to do that!

Secondly I am writing this email sitting on the train. The reason I can do this, direct into gmail is that I have 3G Internet access. I had a USB dongle from Three for about a year. It worked OK but was a little slow at times for the computer to recognise. Also there was always the worry that a careless commuter would knock it and maybe damage my computer too.

I thus decided to take an upgrade to a 'Mi-Fi' device. This is a package the same size as a mobile phone the sole purpose of which is to act as a 'tether' to WiFi enabled devices offering them Internet in the same way you would get it in a cafe or railway station.

Being a dedicated device if no screen to speak of or voice calls to handle the battery life is pretty good. It lasts a good few hours. Another advantage is that you can get your phone to use wifi instead of needing its own data plan. If you are the sort of person who does not send many texts or make many voice calls then having Internet access taken care of means you can go to Pay as You go for the phone. A substantial saving.

My old USB dongle cost me £15.99 a month for 5GB of data. I upped this to a whopping 15GB a month with the new MiFi for just an extra £3. A great deal considering how mean the data plan parts are in most mobile phone deals – the ones with say they are unlimited tend to have the caveat that you are not allowed to 'tether' – that is to attach a laptop or other device over WiFi or cable to the phone. The MiFi is 100% tethering.

As well as the phone and my netbook the MiFi also lets me use the web browser capability of my Kindle when on the move. Not as ideal but if it is standing room only it does mean I can at least get to read some online news.

An added benefit on my phone is turning it to only use WiFi for the Internet has lead to a HUGE improvement in battery life. I suppose it is because to does not have to do the radio equivalent of shouting to reach the base station. Having separate phone and mifi means you are carrying twice the battery capacity with you.

If you mostly use email/twitter rather than calling/texting then consider the two gadget approach, with just a basic PAYG phone, it makes a lot of sense.

Think Complexity

Allen Downey

O'Reilly Media
ISBN: 978-1-449-31463-7
160pp.
£ 22.99
Published: March 2012

reviewed by Roger Whittaker

This is a small book about some very interesting topics under the general heading of "Complexity Science", with hints and examples of how to write Python programs implementing some of them.

If you have come across and been intrigued by such things as fractals, Conway's Game of Life, Cellular Automata, the Prisoner's Dilemma, and artificial societies, you will enjoy this book.

"Complexity Science" is something that came about largely as the result of the existence of computer programs capable of modelling such things: many surprising results including particularly the emergent large scale properties of systems obeying very simple small scale rules have been discovered through such models. And those emergent properties give rise to deep philosophical questions.

The book does not go into huge detail about each of its topics, but gives you enough material to start your explorations, and in many cases provides some sample code or programs that can be downloaded from the book's web site at `http://thinkcomplex.com/`.

The book has been developed from materials that the author used for a course at Olin College – a course that sounds like a lot of fun.

The preface begins with a section headed "Why I wrote this book" which begins:

> *This book is inspired by boredom and fascination: boredom with the usual presentation of data structures and algorithms, and fascination with complex systems. The problem with data structures is that they are often taught without a motivating context; the problem with complexity science is that it is usually not taught at all.*

The book encourages the reader to think about how to implement some of the complex systems covered, but also (and very interestingly) motivates philosophical thinking about their meaning. About, for example, the implications of the fact that some of Wolfram's one dimensional cellular automata and Conway's Life are Turing complete. This introduces a discussion about realism versus instrumentalism, and about free will.

I suppose this is the only O'Reilly book that has all (or probably any) of the following as section headings: "Traffic Jams", "Free Will", "Sand Piles", "Pink Noise" and "Determinism". Great stuff.

The entire book is also available online under a Creative Commons licence, and the author encourages readers to submit further related case studies. There is an expensive reading list and many interesting links and references.

The author, Allen Downey, is also the author of a number of other free books that are available from `http://greenteapress.com/`, including Python, Java and C++ versions of "How To Think Like a Computer Scientist" and "Think Stats: Probability and Statistics for Programmers".

If like me, you find the topics covered in this book interestingly mind-boggling, you will enjoy it. And if you so wish, it can be an excellent starting point for an extensive and open ended exploration.

HTML 5 Cookbook

Christopher Schmitt and Kyle Simpson

O'Reilly Media
ISBN: 978-1-4493-9679-4
284pp.
£ 26.99
Published: November 2011

reviewed by Gavin Inglis

O'Reilly's Cookbook line feels familiar by now: technical information packaged as recipes for particular tasks. This book is not an introduction to HTML 5; but neither is it particularly specialised. It covers a wide range of HTML 5 features with enough code detail to see them actually working. As with some other books in the Hacks and Cookbook range, the solutions are not always technical; sometimes they simply provide an explanation of how to approach a particular task.

The first chapter concerns additions and changes to the markup itself, and is largely conceptual. Here you'll find discussion on how the `<em>` and `<strong>` tags have changed their meaning in this iteration, your new freedom to wrap links around block-level content, or how to choose between the `<article>` and `<section>` semantic tags. The problems and solutions in this chapter can seem contrived, but reading through them is actually a reasonable way to pick up the subtleties of the new markup.

Chapter two offers a healthy dose of reality. The HTML 5 specification is still evolving, and this section offers techniques to deal with the uncertainties of browser version and partial implementation, such as graceful degradation of functionality using Modernizr. Anybody who has been around HTML/XML for long enough will have to suppress a smile at the line "no browsers currently support this feature". Nevertheless this is all good practical stuff, with occasional tricks such as creating new elements explicitly in the DOM to trick Internet Explorer into recognising them and applying CSS.

The remainder of the chapters go into more depth on specific topics. First up is nitty-gritty about forms, including such delights as integrated input checking and autofocusing on a particular field when a page loads.

For audio, the text shows how to reference files in multiple codecs and offers an impressively small chunk of code to visualise an audio waveform using the canvas. This chapter wraps up with the design and implementation of a custom audio player in a mere five pages. Examples like this really show the power of HTML 5. A similar chapter follows on video.

Then follows a change of scene, discussing microdata and how to hook into the formats collected at `http://schema.org/`. Also here are instructions on attaching and accessing custom data in the markup without necessarily displaying it.

Among the closing chapters is one on geolocation; it includes an interesting example on determining a street address from lat/long – and vice versa – using the Google Maps API. Perhaps less interesting is the example which provides directions from one branch of Starbucks to the next closest branch of Starbucks.

Although already discussed, the HTML 5 canvas gets a chapter of its own. The recipes here range from basic resizing, drawing, and pasting of images, to gradients, patterns, text, graphs and animation. It closes with some ideas about saving a generated graph to a file.

Although the Cookbook series are intermediate volumes, and often seen as an aid to a language already mastered, the HTML 5 Cookbook can be read from start to finish as a solid learning experience by anybody who has previously taken fifteen minutes to read a conceptual introduction to HTML 5. The specific examples, presented with code, are just what's needed to get going in practice, grounding the somewhat elusive changes to the spec in usable details. In short, not a bad way to update your HTML skills for the next big step.

Programming Perl 4th edition

Tom Christiansen, brian d foy and Larry Wall and Jon Orwant

O'Reilly Media
ISBN: 978-0-596-00492-7
1184pp.
£ 42.50
Published: March 2012

reviewed by Martin Houston

Programming Perl – otherwise knows as the Camel Book – was one of the original books about Perl, a language that is now over 20 years old but is still very much alive and evolving. This is the only book where I have bought each edition as it came out. There has been a long wait for the new one.

Much of the popularity of Perl in the early days was the easy to read documentation in the form of earlier versions of this book which first appeared back in 1991. Larry Wall is a linguist and that has showed by the approachable style and use of humour of this book. Although Larry is now only one of several co-authors of the latest version the very readable style is still very much in evidence.

Version 4 of this book has been a long time coming. There was only a 4 year gap between edition 2 (1996) and edition 3 (2000) but we have had to wait three times as long for an update. That is about half of Perl's life without an update to the major reference book!

Has it been worth the wait? The new version of the book covers Perl 5.14 which is for a while yet considered the main supported production release. The printed version of the book is inevitably now quite a bit heavier and is now of a roughly equivalent shelf filling capacity to the other O'Reilly title 'Programming Python'.

Luckily for the health of itinerant Perl programmers backs an ebook and pdf version is also available, at a very keen price (less even for FLOSSUK members) and with the huge added benefit of NO DRM. Yes you read that right, O'Reilly appreciate that DRM destroys the value and convenience of the electronic form. Please repay that trust by betting your co-workers to buy their own copies of this very useful book.

The electronic copy of the book was in fact available for sale first, initially just as a pdf, but then followed by more ereader friendly formats for Kindle and iPad. I bought my own ebook to update my now 12 year old printed copy of Programming Perl.

The printed copies are now available and I was sent one of these for review.

There is still a place for both. Trying to read really long sections in comfort from a computer screen is never going to be as easy, ereaders like the Kindle make this a lot less arduous but for something you are going to use A LOT having both a paper and an electronic copy to hand is a best of both worlds. The only slight quibble I have with books of this size is the weight. I know it would add to the cost but splitting the book into two volumes would make each one a lot more comfortable to read. If the 5th edition grows by as much again that will have to be a serious consideration.

I have not covered what is in the book, as you can find that out for yourself in an instant using the Internet it would be dull for me to repeat it here.

Why should I buy this book? Well if you use Perl and are working on the 3rd edition and other works of the same age you are missing out on more than a decade of improvements to the Perl language. Just think of how much more productive you will be if you stop having to type messy constructs like `$x = 3 unless defined $x;` and instead say `$x //= 3;`.

Buy it! Even if you do not yet use Perl, discover why this now mature language is still very much alive and kicking and doing important work.

Perl was one of the original bastions of Free Software, and has built up a huge base of very solid contributed code in the form of the huge CPAN archive network. You may find that what you are after has already been done, is well tested and in wide use.

Even if you are happy with your circa year 2000 level of Perl skills you may just come into contact with code that other people have written in the past decade so the clear treatment that this book gives on all of Perl, both the new and the mature, you will find very useful and well worth the cost, after all, as Perl is free software, there isn't that much to spend that money it helps you earn is there ;)

Confessions of a Public Speaker

Scott Berkun

O'Reilly Media
ISBN: 978-1-4493-0195-8
240pp.
£ 12.99
Published: January 2011

reviewed by Paul Waring

Plenty of technical books are available for every programming language, database server and application you can think of. However, there is no escaping the fact that sooner or later you will find yourself in front of an audience delivering some form of presentation, so a book which promises to help you along the way has got to be worth a look.

The first chapter is all about the author, who unfortunately comes across as a bit egotistical without imparting a great deal of useful advice. This is followed by a chapter on the fear of speaking, although the first half of this concentrates a bit too much on the psychology of fear, and only later does the author offer some suggestions on how to beat it. Some common sense advice is given, such as walking around the room before your talk to get the feel of the layout, and talking to potential audience members beforehand so that you are not speaking to a room of complete strangers.

Chapter three discusses payment for speakers, and it is interesting to see how much you can earn from speaking if you are sufficiently talented or (more often) famous. Chapter four, on the other hand, contains some useful advice on how to deal with a 'tough' room, such as persuading people to move to the front when you find yourself in the unfortunate position of being in the opposite stream to a famous speaker whom everyone wants to listen to.

Chapter five covers how to avoid 'eating the microphone', which is apparently what happens when a speaker wanders off-topic and the audience lose interest. Despite the slightly confusing name, there is some helpful advice on choosing a title (more important than you might think) and a five-point list which you can build a talk around.

The final chapters cover how not to bore people (given the importance of this skill, I'm surprised it doesn't appear earlier) and some of the lessons the author has learned from his experiences speaking to various audiences. A useful selection of appendices with various tips and advice on everything from how to make a point to what to do in the event that things go horribly wrong (which should be read by organisers as well as speakers), rounds off the book.

Overall, this is a valiant attempt to explore a subject which is difficult to write about, since many of the lessons about public speaking can only be learned by practising and trial and error. There is perhaps a bit too much time spent on issues such as speaker fees, and I'm not convinced by the claims that readers will 'greatly benefit from this book' and that your talks will be 'ten times better' as a result, but nevertheless it makes useful reading.

Mapping With Drupal

Alan Palazzolo and Thomas Turnbull

O'Reilly Media
ISBN: 978-1-449-30894-0
150pp.
£ 15.50
Published: December 2011

reviewed by Mike Fowler

These days Drupal needs little introduction – if you haven't build a site yourself with Drupal, you'll almost certainly have used one that was. Web mapping is little different it seems these days too. From websites with mundane store finders to the more interesting route planning to that relative you've not visited in too long. It's the interactivity of the route map that makes it much more interesting than a simple image and this book aims to show you how to use the mapping modules of Drupal to add dynamic and informative maps to your site.

The first two chapters highlight the basics of web mapping (especially terminology) and the features of a good map. From here we move into a sustained and progressive tutorial that will quickly leave Drupal beginners behind. The authors thankfully point the beginners in the direction of some basic materials for Drupal as configuring the mapping modules is not straight forward. There are two main mapping modules available in Drupal are OpenLayers and GMap, and both receive thorough treatment. Conveniently each chapter shows you how to achieve the task at hand in both modules rather than dividing the book into an OpenLayers half and a GMap half. It certainly makes evaluating the modules easier – especially when you see the JavaScipt involved in fine tuning OpenLayers.

To me the strength of a sustained tutorial is also it's weakness. To get the most out of the book you really do need to work your way through the examples on a private Drupal installation. Simply thumbing through to find the feature you need most in isolation is tricky as it will certainly depend on all that preceded it.

I recommend that anyone who is considering adding maps to their Drupal site read this book and to work through the tutorial. Once you've gone through the process once, working with whichever module you prefer is much simpler. With web maps the complexity really is with the developer – once you've got your map embedded in your website the users will know exactly what to do.

Programming Your Home: Automate with Arduino, Android, and Your Computer

Mike Riley
O'Reilly Media
ISBN: 978-1-93435-690-6
200pp.
£ 25.50
Published: March 2012

reviewed by Paul Thompson

Despite the steady evolution of X10 and other automation standards over the last few decades, it's surprising that there is still only a very small mass-market adoption of home-automation products (besides the thermostat and the Goblin Teasmade). The recent availability of low-cost microprocessors may see more appliance manufacturers providing communication stacks and processing capabilities to washing machines and refrigerators.

Since hearing IBM "Master Inventor" Andy Stanford-Clark's memorable "The House that Tweets" talk at Oggcamp a couple of years ago I've been meaning to implement something similar – but never quite got around to it. Mike Riley's book promises to lead me through that process (and much further), with the help of Arduino and friends.

I was sceptical about the claim that it was written for "smartphone programmers, web developers, technology tinkerers, and electronics project DIY'ers", and was curious to see how it could appeal to all; really it doesn't. In writing to such a broad audience, he inevitably has to explain the fundamentals of every step in sometimes tedious detail, with the risk of boring those who don't need such hand-holding.

Part I is all about preparations, and save for those who've not explored more than the workings of a pencil, Chapter 1 should be skipped entirely. It has a rather condescending description of what is required to begin working on home-automation projects such as how to set up the workbench, then some suggestions on documenting ideas followed by some wordy disclaimers to absolve the publisher of liability for physical harm or damage that may result from the tinkering that the remainder of the book promotes.

Interesting home-automation technologies are briefly mentioned in Chapter 2, focusing particularly on the Arduino and XBee, and concludes with yet another half-page of warnings and disclaimers.

Part II is the main body of the book, comprised of step-by-step instructions on how to build increasingly sophisticated projects. Chapter 3 describes how to create a simple water-level notifier that works by sensing changes in impedance of a flex resistor attached to a float. It is a neat design to introduce the newcomer to the basic techniques for programming the Arduino and some electronics required in subsequent chapters, as it has a minimal set of components and just 32 lines of code at its core. Ready-built libraries are then used for coding an Ethernet shield to send status messages to a PHP server and finally to email alerts to the house-holder. I'm surprised to be already 20% through the book and feel that what I've read so far would have been better suited to two or three pages of a magazine had the narrative been more succinct.

The "electric guard dog" in Chapter 4 uses a PIR sensor to detect visitors at the door, a servo motor to shake the curtain and an Arduino wave shield to deliver an audible "bark" (.mp3) to keep the postman on his toes. As with many of the worked examples in the book, I can't imagine this nor indeed the "tweeting bird feeder" in Chapter 5 to be particularly high on the householder's list of must-have gadgets, but that's not the point: these are meant to inspire the reader to dream up his own designs and to provide the confidence to build them – and the author does a good job of that.

The bird feeder uses the ideally-suited XBee radios and gently introduces the reader to Python and SQLite in a fun design to monitor the feeding behaviours of pet- or garden birds using a sensor on the perch. A "Next Steps" section at the end of each chapter in Part II provides interesting suggestions for extending and enhancing each design. For example the ambitious may consider extending the bird feeder service to plot (using a Python graphing library like CairoPlot) the average duration of bird visits, variances due to seed type, and even collate and share data online with other bird enthusiasts. I'm making rapid progress through the book (already at page 83), and realise that not only is the text too verbose but the inclusion of many of the diagrams and images is completely unnecessary, for example the black-and-white photograph (in hardcopy format) of resistors and the half-page cartoon-style drawings are superfluous. Other photographs could have been better if substituted for uncluttered sketches or schematic diagrams.

The "package delivery detector" in Chapter 6 is really a variation of the tweeting bird feeder. It reuses the components and techniques of the previous projects to detect when a parcel arrives in a mailbox, then with the help of some Python libraries, cross-checks XML tracking data from UPS and FedEx couriers' web sites to notify of an eagerly-awaited delivery. The benefit of this project is in demonstrating how with sufficient imagination the components, libraries and techniques from previous designs can be combined to build powerful services without the need for advanced skills in electronics or programming.

In Chapter 7's "web-enabled light switch" the book moves from being a series of introductory exercises to something that most of us will consider as being real home-automation. With the aid of some Ruby on Rails, an Android phone and some venerable X10 components, it demonstrates how to control a home's lighting and other electrical appliances from a custom smartphone application. Actually, this first step is just a graphical toggle switch on an Android phone controlling a single lamp, but the possibilities for extending the system are endless, including interfacing it with other projects in the book, such as Chapter 8's "curtain automation". That employs the Arduino motor shield controlling a stepper motor to control blinds or shutters depending on light and temperature. The programming (use of libraries) and electronics involved are trivial, but the thought of bolting a motor and pulley to the living-room wall might put some off.

The "Android door lock" in Chapter 8 can open doors wirelessly from a smartphone by energising a 5v relay from a PIC-based Sparkfun IOIO ("yo-yo") board – a hardware bridge allowing Android devices to communicate with I/O ports. The relay can subsequently control any connected electrical appliance and for convenience the project employs a an off-the-shelf PowerSwitch Tail (a mains socket with built-in relay) – so only minimal electrical skills are required. This is the most sophisticated project of the book: on triggering the relay via the IOIO, a photograph of whoever is entering is taken via the phone's built-in camera and emailed (using JavaMail for Android) as an attachment to the designated recipient. Advantages in power-saving and flexibility of location are gained from running the required web server (to listen for inbound requests for a specific URL) on an old Android smartphone rather than a PC. A second Android phone or tablet is required to run the client application (plus Eclipse IDE, Android SDK and ADK).

Chapter 10's project, "Giving Your Home a Voice" is a demonstration of the use of an AppleScript program using Mac's Text-to-Speech renditions to notify the householder of activation of specific events, and to instruct it to execute events such as "play music" or "unlock door" at the householder's spoken command. Examples in the "next steps" section suggest how this could be extended or combined with previously-constructed projects to have a highly automated home.

Part III is a discussion about some of the emerging technologies in home automation, and some predictions on what smart homes might entail in the future. It will undoubtedly seed many more innovative ideas for the reader. We are seeing a proliferation in the number of sensors and notification features in the typical home, but a means of managing and monitoring them is still missing. Maybe it will take Google, Microsoft or Apple to plug that gap.

I enjoyed the refreshing and novel projects presented in the book and found the inclusion of "next steps" sections (with suggestions of where to go with each project to further enhance them) to be useful. There is surprisingly scant description of the security considerations necessary in sending home-automation data across the internet though. Mike Riley's infectious enthusiasm for the technologies has provided inspiration for me to explore further in this field; I too now have the urge to constantly be on the lookout for the automation possibilities in hitherto uninspiring objects around the home, such as the letterbox and the bird table.

I found the informal American narrative style both irritating and tedious (e.g. “we’re almost done... we just have to write the body... all together now... we’re nearing the home stretch...”). The large number of typographical errors was disappointing, as was the decision to use a sans-serif proportional font for code and filenames inline (i.e. within paragraphs) and to let them get split/hyphenated across long lines – as if they were regular English words.

In summary, the book contains some good project ideas, but I believe it would have been better if shortened and published as a series of magazine articles instead.

Programming Your Home: Automate with Arduino, Android, and Your Computer

Mike Riley
O’Reilly Media
ISBN: 978-1-93435-690-6
200pp.
£ 25.50
Published: March 2012

reviewed by Kimball Johnson

‘Automate with Arduino, Android, and Your Computer’; A pretty impressive subtitle, for what is a pretty impressive book. I have read many of the Pragmatic Programmers series of books, and this is another that is once again up to their usual standard. The author describes his primary objective as ‘get you excited about... home automation and... build upon these and your own ideas’. Well I think it will succeed in this. I am already excited about home automation, I lived as lodger in a house that greeted you when you came in the door and set the heating and lighting accordingly, so I am well aware of the possibilities, however have never had an opportunity to build any of my own system. I wanted this book to give me ideas, in preparation for owning my own home (or at least inflicting my home owning friends with fun toys). I believe it easily achieves this.

The book is separated into three parts, Preparations, Projects, and Predictions, it also has an appendix that contains information on installing Arduino libraries. As you would expect, the preparations section gives an overview on home automation, and gets you setup with a working environment to proceed with the projects. What I especially like about this, it is not limited to installing software and telling you the hardware to buy, but it give advice on the physical workspace required for electronics and good practices for documentation. These extra attentions to detail make the book stand out, and are present throughout.

Moving onwards, there is a nice selection of projects which as well as being useful in their own right, cover a lot of ground for hardware and software solutions required in home automation, for example the first project is a water level notifier, this covers interfacing with an external analogue sensor, specifically a flex sensor but technique is valid for any analogue sensor where you have a valid and invalid ranges to act upon, and then for notification it combines a web-based mailer system, and an ethernet shield for the arduino that purely activates the web form. This again gives flexibility for other projects as you could alter the web page side to log into a database, send multiple alerts perhaps with SMS messaging, and also use the arduino side for other interaction with web forms.

Finally the Predictions section of the book the author takes a look forward to 2025 and how prevalence of embedded devices can enable not only gimmicks that save time and effort, but can make a real difference to improving your life and protecting the environment. This gives inspiration for the reader to look beyond the detailed project plans provided and create their own, plus there is a chapter with further ideas to try. Although these have very little detail, they give ideas based on the projects in the book on how to get started, enabling the reader to build on the excellent ground given.

In summary this is an excellent book, well written so even without a huge amount of knowledge of embedded systems programming, or even programming at all the reader should be able to build the projects detailed and be inspired to learn more. There is a bibliography in the book to help with the programming and hardware aspects covered, but I would also combine it with the Arduino Cookbook (O'Reilly Media, 978-0-596-80247-9) that I reviewed for the previous newsletter as that also gives wonderful grounding on the possibilities the Arduino gives you.

Node for Front-End Developers

Garann Means

O'Reilly Media
ISBN: 978-1-4493-1883-3
60pp.
£ 11.50
Published: February 2012

reviewed by Kimball Johnson

Since I first heard about Node.js I have been intrigued by the concept, not because I think that Javascript is a particularly amazing language, but because it takes a new approach to an emerging problem. Websites no longer merely take some data, think for a while and send some data back, they act more like interactive applications, with constant data flow to and from a server. This is not something Apache et al were intended for, and although they do a pretty good job, an opportunity to explore new approaches is always welcome and interesting.

The book takes a very specific approach, as the title says, it is aimed at front end developers, those that are already well versed in bending Javascript to their will, and probably working against a set of APIs for data transfer. This book aims to concisely build on that knowledge directing it towards creating a server with Node, starting with a basic server and building up to more advanced topics such as sharing libraries with client and server.

Unfortunately it takes concise a little too far, the book has 45 numbered pages plus prefix and appendix, but even if I am generous in counting useful pages of content from the whole book I can only make it 39. The author claims that she is going to start off with showing you how to perform a task and then move on to using libraries to do it for you, and admirable aim as it is always useful to understand the libraries you use, even in concept, but sadly there is no depth and even the examples never grow beyond snippets. There are cases where you are left hanging, desperate for more information just to fill the void, even if only to help with searching online for more, for example there is a section where the author comments that restarting the server everything you make a code change is tedious, and to use one of the methods to allow automatic server reloads. That's it, no hint as to what these methods might be, or even really a good point to search from.

Now, it is not all bad, I came in knowing nothing about Node, and with the book I suspect I could write a basic application, and find more about each topic by searching for the libraries mentioned, reading the documentation etc. However I could get all that from a simple search of 'Node.js Tutorial' which as the first link gives free book, all be it with a nag to buy a bundle of it and another book, plus forums, howtos and even as the last link on the first page, a short review of 7 free books. Interestingly to note as well, if you were to follow the nag link on the first link, they want less for two books than the RRP on this book (in US Dollars).

In summary, it's not a terrible book, and definitely worth getting if you like physical books and can pick up a copy cheap, maybe with a discount or similar, but I don't think it is worth the $14.99 on the back cover, or the £8.05 that Amazon ask for it. Maybe a second edition will go into more depth, with fuller examples, even if it was still short, that would make the world of difference.

Liars and Outliers: Enabling the Trust That Society Needs to Thrive

Bruce Schneier

John Wiley & Sons
ISBN: 978-1-118-14330-8
384pp.
£ 16.99
Published: February 2012

reviewed by Roger Whittaker

The view one takes of human nature is often taken to be a function of one's political or perhaps theological outlook. People are imagined to be born either sympathising with Rousseau or Hobbes: those on the left thinking that we are essentially noble and cooperative creatures who would build paradise on earth if only the oppressors could finally be defeated, and those on the right considering any talk of human beings being naturally cooperative to be painfully and dangerously naive, and believing selfish motivations regulated by law to be the only basis of society. Meanwhile the Christian conservatives are muttering in the background about "original sin".

But whatever people try to tell you, one's view of human nature is no longer simply a matter of temperament and political outlook.

There is a **science** of human nature, and much of the work that has been done in the fields of game theory, evolutionary psychology, agent-based models and related areas has been done using the sort of methods described in sections of the book "Thinking Complexity" reviewed above. See for instance the section on Sugarscape in that book, or the classic discussion of the iterated prisoner's dilemma in Robert Axelrod's "The Evolution of Cooperation". These show, respectively, that there are reasons why (in the absence of progressive taxation) the distribution of wealth can be expected to be somewhat as it is in the western world, and (more hopefully) that if the conditions are right, cooperation emerges spontaneously and everyone benefits. But this latter conclusion is well known to readers of this newsletter, because we know that the open source software development model works.

Bruce Schneier is well known for his frequent blog posts about security issues. He is a specialist on cryptography and computer security, but has widened his remit to the wider security issues seen during the so-called "War on Terror", and has been scathing about "security theatre". He has written:

> *Secrecy and security aren't the same, even though it may seem that way. Only bad security relies on secrecy; good security works even if all the details of it are public.*

and

> *If researchers don't go public, things don't get fixed. Companies don't see it as a security problem; they see it as a PR problem.*

Open source advocates will find these sentiments both familiar and sympathetic.

This book may disappoint those who are looking for a collection of Schneier broadsides against foolish or ineffective security policies. But that's not the aim. What he does here is first to look at the basic principles of trust in society, go over the work that has been done about the conditions for cooperation to emerge, examine the effects of a growth in the size of societies, look at the effects of reputation and shared moral values, and then discuss a wide variety of "societal dilemmas" where individual interests are in conflict with the group interest. These are analysed cleverly and in some detail.

The book is not directly about computer security, but computer security is a subset of security as such, and everything that Schneier writes here has relevance to security in the computer world. But it has a wider significance also, because this is a book about how to make a better society. And the way to make a better society is not always to bang political drums of one kind or another: it is to create conditions in which cooperation is favoured and exploitation is discouraged; in which honesty is rewarded and lying does not pay; in which the rules and the institutions are tweaked in such a way as to optimise them for trust and reduce the opportunities and benefits of defection.

These are vitally important aims, and this is an important book.

Contributors

Mike Fowler is a Senior Software Engineer working in the Midlands. He is a firm open-source advocate and has helped companies move from Microsoft based solutions to complete Linux/PostgreSQL stacks. He is a major contributor to YAWL, an open source BPM/Workflow system. When not melded with a keyboard, he spends his time playing as many different stringed instruments as he can.

Gerry Gavigan is the chair of the Open Source Consortium (OSC). He's a former professional engineer with multidisciplinary experience in optoelectronics and control engineering. In 1983, one of his insights was to repurpose a third party Epson FX80 printer buffer card as an Arduino-like instrument controller except it had IEEE 488 as its network interface. Subsequently he developed a career in economics and policy consultancy. He's being using Linux on the desktop since 1999.

Martin Houston has been using Unix since 1980 starting with a PDP11/70 and Linux since 1994. He was the founder of the original UKUUG Linux SIG and is currently a freelance Linux consultant. He lives in Essex with wife Teresa and 2 cats, 2 dogs and too many computers.

Gavin Inglis works in Technical Infrastructure at the EDINA National Data Centre in Edinburgh. He is a teacher, photographer and musician who has recently discovered the joys of spreadsheets as a recreational tool.

Kimball Johnson is a Systems Developer for Lancashire County Council. He has been programming since a very early age, starting with BBC Micros, then MS DOS and Windows Systems: however was enlightened with a copy of Debian GNU/Linux Woody at university. He is always wanting to learn and has recently started to take on embedded programming on a variety of devices, on everything from a Arduino to a Nintendo DS.

Jane Morrison is Company Secretary and Administrator for UKUUG, and manages the UKUUG office at the Manor House in Buntingford. She has been involved with UKUUG administration since 1987. In addition to UKUUG, Jane is Company Secretary for a trade association (Fibreoptic Industry Association) that she also runs from the Manor House office.

Ian Norton is a keen Perl hacker co-founding and co-chairing Northwest England Perl mongers for the last three years. Lately his focus has gone back to an old love in the form of network monitoring and management since working on an OpenNMS project for his employer. When not wrangling exim, herding linux system and working on access control technologies, he can be found roleplaying or singing tenor.

Paul Thompson is based in London and works for SUSE as a Linux consultant specialising in the financial sector. He can be reached via @raganello on Twitter.

Paul Waring is chairman of UKUUG and a director of a wholesale insurance broker. Outside of work he can usually be found filing documentation bugs against various open source and free software projects.

Roger Whittaker works for SUSE supporting SUSE Linux Enterprise Server for major customers in the UK. He is also the UKUUG Newsletter Editor, and co-author of three successive versions of a SUSE book published by Wiley.

Contacts

Paul Waring
UKUUG Chairman
Manchester

Phil Hands
Council member
London

Martin Houston
Council member
Chelmsford

Kimball Johnson
Council member
Lancaster

Holger Kraus
Council member
Leicester

Ian Norton
Council member
Rochdale

John Pinner
Council member
Sutton Coldfield

Howard Thomson
Treasurer; Council member
Ashford, Middlesex

Jane Morrison
UKUUG Secretariat
PO Box 37
Buntingford
Herts
SG9 9UQ
Tel: 01763 273475
Fax: 01763 273255
`office@ukuug.org`

Roger Whittaker
Newsletter Editor
London

Alain Williams
UKUUG System Administrator
Watford

Sam Smith
Events and Website
Cambridge