

FACS A C T S

FME
A ACM
C T
L F C
METHODS C
BCS R SCSC
M
Z A
UML
IFMSIG
E E
E E
E



Formal Aspects of
Computing Science
Specialist Group

The Newsletter of the Formal Aspects of
Computing Science (FACS) Specialist Group

ISSN 0950-1231

About *FACS FACTS*

FACS FACTS [ISSN: 0950-1231] is the newsletter of the BCS Specialist Group on Formal Aspects of Computing Science (FACS). *FACS FACTS* is distributed in electronic form to all FACS members.

Submissions to *FACS FACTS* are always welcome. Please visit the newsletter area of the FACS website [<http://www.bcs-facs.org/newsletter>] for further details.

Back issues of *FACS FACTS* are available to download from:

<http://www.bcs-facs.org/newsletter/facsfactsarchive.html>

The *FACS FACTS* Team

Newsletter Editor Margaret West [m.m.west@hud.ac.uk]

Editorial Team Jonathan Bowen, Paul Boca

Contributors to this Issue

Jawed Siddiqi, Margaret West, Jonathan Bowen, Paul Boca, Richard Bornat, Sharon Curtis, Anthony Hall

If you have any questions about FACS, please send these to Paul Boca [Paul.Boca@virgin.net]

Amir Pnueli (1941–2009)

We note the sad death on November 2nd 2009 of Professor Amir Pnueli. Professor Pnueli was an internationally recognized pioneer in the area of verification, and in particular temporal logic. In 1996 he was the recipient of the 1996 Turing Award, the highest distinction that can be bestowed on a computer scientist.

Editorial

Welcome to the second issue of 2009: in October we held our AGM and this issue includes a report from our Chair, Professor Jawed Siddiqi and a list of members of our committee (together with their image). There is also an announcement by Paul Boca of the completion of our book based on the FACS seminars held between 2005 and 2007: We would like to join with our Chair in congratulating and thanking Paul for the huge amount of work he has put into this project. The book is called

Formal Methods: State of the Art and New Directions

and is dedicated to Peter Landin. An obituary of Peter Landin has been written for us by Richard Bornat.

We have held two seminars this autumn – the first was joint with BCS-Women and given by Professor Marta Kwiatkowska of the University of Oxford. The second seminar was joint with the London Mathematical Society and given by Professor Mike Gordon F.R.S. of the University of Cambridge. More details (including a brief report of the first seminar) are presented in this issue.

Since we are approaching Christmas we include an item with a seasonal flavour: *The Cross-Stitch Theorem* by Sharon Curtis. On that note, we wish you

Seasons Greetings and Good Wishes for 2010!

BCS FACS Chair's Annual Report 2008–2009

1. Introduction

What is certain is that the BCS has changed, some would say beyond recognition, others might say significantly. There is a tension between what the BCS and its members want particularly as represented through the Specialist Groups. In this annual report I have set out an overview of the key developments such as the Members Group report, Members Group Convention and the BCS Computing Academy with a commentary to provide a flavour of what has been happening. An earlier draft version of this report was discussed at the AGM and the current version reflects this discussion. In light of the AGM I have also put forward specific proposals that provide a direction and plan for the forthcoming year.

2. Member Groups Report

The BCS in attempting to better understand what the role of specialist groups and branches and its member want produced a consultative document termed as the Member Groups Report which in one sense recognised the tension between the BCS and its members. On the one hand BCS wants its membership to celebrate that the BCS provides considerable support in terms of resources, which include finance, IT, staff, premises, etc., and these are very much appreciated by the volunteers and member groups. On the other hand the report recognised that there are a number of anomalies in the treatment of member groups and these and other issues are noted together with the serious concerns of the volunteers that organise and maintain the member groups.

Those who take view that BCS has changed beyond recognition will probably disagree with this starting point, those pertaining to the significant view change chose to see where the outcome of this report and process will lead to. The report by carefully choosing its terms of reference really avoids the big question at the heart of the change which is:

What is the appropriate balance of power/participation between the BCS centrally (i.e. the paid officers) and the volunteer members?

The report attempted to avoid or "reconcile" the first question with a broad recommendation that a sort of a "Charter of membership groups" could be produced. in which the BCS makes clear to all the membership groups the role and value of the BCS's obligation to them and what the BCS can expect from them. Moreover, to smooth things along Volunteers and Staff also need relevant information/training in this respect.

3. Member Groups Convention: Proposal for new Membership Board

At the September Member Groups Convention at the Royal Society it was announced that Membership Services Board (MSB) will be replaced by a new Membership Board and all BCS member groups will report to this Board.

At that meeting it was indicated that whilst all the details are yet to be worked out, nevertheless, it is expected that two or three functional subcommittees will be part of the Board - one for Finance, one for Policy and one for Best Practice. In order to help develop the details and to answer questions, the volunteer representatives on MSB put together a joint paper about the way we see this Board's function, what its TOR should include and the critical questions that need to be sorted out preferably before the Board is formed. The paper was drafted by the four main member communities, Branches, International Sections, Specialist Groups and YPG, currently in operation within the BCS, in response to the request for input to the new Membership Board Terms of Reference (TOR). The paper raised a series of questions about the new Board, and these questions and their answers have been debated at the last Council meeting. The paper has been circulated to all Council members (including all the trustees).

Recently it was reported that the Trustees agreed the proposed high level structure. It is claimed that by having a Finance Committee and a Policy Committee serving all the member groups, they will receive a better and more consistent service, and that they will have a more effective route for discussing and influencing policy.

However, since this development there have been a number of emails in the Chairs Forum from Chairs of SG's that think these questions and their proposed answers fell short of the issues that need to be address that take a contrary view and have raised matters regarding the balance of power. I have provided some extracts from them the last one being made your Chair:

"It is absolutely vital to have two meetings a year so that some form of check and control can be brought to bear on plans that appear to come from a few at the centre (mainly full-time officials I think)..."

"The BCS and the SGs have in recent times moved from a fairly distributed/autonomous model to a more centralised/controlled model"

"Cutting the means, by which we, the members, can participate, is therefore terribly dangerous, as it allows decision-making to become ever more centralised".

"Whilst it may be that BCS "central" provides services to its members much of the work done and the success BCS experience is also because of the significant effort that is freely given by volunteers . The recent changes that have taken place has severely perturbed that delicate balance, so much so, that many of us feel our voices are not being heard or that we have no impact on decision making"

So we can see the tension still exists and BCS will have to do much more to attempt reconciliation.

4. Computing Academy

There has been a long debate within the computing community regarding the necessity to form a learned society for computing. Key discussions have taken place in three different forums. The first is the BCS itself; the second, the Council of Professors and Heads of Computing (CPHC) - this is the subject body for computing in which all university computing departments via their senior academics participate and finally, UK Computing Research Committee UKCRC which is an expert panel of CPHC and the BCS. Recently as part of its renewed mission the CPHC agreed to work with BCS and UKCRC to form the BCS Computing Academy. The BCS announced its intention:

BCS, The Chartered Institute of IT, announced its intention to create a new Academy of Computing to create an integrated and coherent approach to advancing Computing across education, research and business.

Dr. Bill Mitchell, BCS Director, Academy of Computing, who is spearheading the creation of the new Academy, comments: 'Computing has joined disciplines such as mathematics, physics and medicine as one of the UK's foremost intellectual outputs. The subject is not only vibrant and growing, but more rigorous, more interdisciplinary, more experimental and more user-oriented than ever.'

There is much to discuss here and plenty of work for those have a strong desire to shape the BCS into a learned society. It was agreed at the AGM that this provided an ideal opportunity for FACS to develop a dialogue with Bill Mitchell.

5. Continuing Changes at the BCS and their impact on FACS

At the same September BCS Member Groups Convention at which John Cooke and Jonathon Bowen represented FACS, there were two further announcements; they were the complete rebranding of the BCS or "bcs" in green as "The Chartered Institute for IT", and the BCS Computing Academy.

We now have available the new logo and template. In my role as the Chair, I received a display unit that opens up into a 2m high poster with our new logo.

From this massive sea of change one consequence is the adoption of a new constitution that is BCS Compliant. I have produced one that has been approved at the AGM. FACS has chosen to transfer the web site to the BCS server and use the BCS facilities for content management termed as the "managed option".

6. The Way Forward

We have agreed that FACS, which operates on events based on its mission and at present largely meaning the evening seminar series, can still operate our activities within the new brief. However, we have recognised that this work will be distributed amongst the committee and others. Any member wishing to

organise an evening event will seek approval from the Officers/Committee and when it approved will be responsible for all the organisation of the event. Paul Boca will provide guidelines on how to book rooms. The member will liaise with the treasurer regarding potential expenses for the speaker before agreeing them with the speaker.

We have continued with our ever successful BCS FACS evening Seminar series held mainly at the BCS headquarters. We have continued to collaborate with several groups. Jointly with BCS Women we held a seminar at the AGM. Our Christmas event last year on "Formal Aspects of Safety Critical Systems" was held in conjunction with Formal Methods Europe and the Safety Critical Systems Club. Last year in November at De-Morgan House we held a joint seminar with the London Mathematical Society and in December this year we will hold another joint evening event. Margaret West continues to produce our newsletter.

Some of you may remember that Paul Boca, Jonathon Bowen and I agreed to produce an edited collection of papers based on the evening Seminar Series titled:

Formal Methods: State of the Art and New Directions

This has been an enormous effort most of it largely once again falling on Paul Boca.

I will on behalf of the BCS FACS Members take this opportunity to thank a number of people. I am grateful to all committee members who carried out their roles, particularly John Cooke and Margaret West for attending the BCS events on our behalf. Margaret has worked very hard and with great patience to continue to produce our newsletter. My thanks to Jonathon Bowen for taking up the challenge of being Treasurer when our entire surplus was taken from us!. Finally, I think it is difficult to express sufficient appreciation for the tremendous effort put in on our behalf by our Secretary Paul Boca. It is only his effort that has kept BCS FACS alive. We need new members to take over the much of the work which is concentrated in very few hands.

In my opinion our plan for the next year can at best be the continuation of the "*Evening Seminar Series*". We will need to assess the feasibility of that at the AGM.

Finally, Thank You all for the privilege of letting me serve as Chair and a further thanks for electing me for another year.

Jawed Siddiqi
Chair, BCS-FACS

*Peter Landin: a computer scientist
who inspired a generation*

Richard Bornat

***Peter John Landin, computer scientist, academic and gay campaigner.
Born 5th June 1930; died 3rd June 2009.***

Peter Landin, who died in June 2009 of prostate cancer, was an outstanding academic computer scientist who set the direction of the subject for his generation. In an astonishing burst of creativity in the 1950s and 60s he showed that programs could be defined in terms of mathematical functions, how to translate them into functional expressions in the lambda calculus, the creation of Alonso Church in 1936, and how to calculate their meaning with an abstract mathematical machine, prefiguring many of the advances of the next couple of decades. Although later he turned away from computer science towards gay politics, he remained an original and extreme thinker, a complex and intriguing character.

An only child born in Sheffield, son of an accountant father disabled in WW1, he was educated at King Edward's Grammar School. During National Service he amused himself by trying to drill a squad to imitate addition in binary arithmetic. At Clare College Cambridge he completed the mathematics degree in two years, then attempted the very difficult part 3, but came away with only a 3rd class degree. The emerging computing industry was a haven for mavericks, though, and he took a job at English Electric, where he cooperated in the development of EE's Deuce from the National Physical Laboratory's early Pilot Ace.

His ideas had already turned to the problems of relating the mathematics of logic to the ad-hoc languages invented by programmers. At the time languages were tied to machines, and programs written for one machine would usually not run on another, not even another machine made by the same manufacturer. He was involved in the discussions which led to the development of Algol 60, a seminal programming language, though he failed to convince the committee that it should be defined in terms of substitution as in the lambda calculus. His interests drew him into Mervyn Pragnell's discussion group. Mervyn used to prowl bookshops spotting people who were buying books on mathematical logic and invite them to his group which read and discussed the books in a kind of perpetual cycle. It even met underground, in the basement of Birkbeck College, where Mervyn knew a helpful technician who would let them in. Despite the depth of his understanding compared to the rest of the group, Peter was always kind and helpful to novices and stragglers, and even programmers could join and learn. Lots of famous computer scientists cut their logical teeth there.

In 1960 he married Hanne and he was taken on as a research assistant by Christopher Strachey, then an independent consultant and later Professor at Oxford, to write a compiler to translate Mercury Autocode, an early programming language from Manchester University, into the machine language

of Ferranti's new Orion machine. He envisaged the compiler as an automatic product of the semantics of the autocode, matching its forms to semantic representations of the instructions of the machine, and generating LISP expressions that could be executed, but his attempt was never finished, and it was a long time before such a radical approach became feasible. He was held in awe by the elite team who were writing the 'supervisor' (operating system) of the machine: he would appear before them, dressed in a long overcoat, and scribble expressions in the lambda calculus, rather small, in the middle of a blank piece of paper, trying to influence them to make their design fit the needs of his compiler.

Even though he never finished his compiler, his time with Strachey wasn't wasted. He clarified his ideas about program semantics and published "The mechanical evaluation of expressions" which showed how to translate programs into the lambda calculus and defined the SECD machine, an operational definition of the meaning of a small language which he hoped might form the basis of the design of future computers. In particular he showed how functions and procedures in languages like Algol, which introduced the now universal notation of nested scopes, could be given a concrete meaning in an implementation. Every compiler writer since has had cause to be grateful to him, and modern programming languages such as JavaScript, which underpins the World Wide Web, all use functional values heavily and have implementations based on his definition of a 'closure'.

Even at that stage of his life computing wasn't his entire life. All of his life he was a radical in politics, a regular on demonstrations, and he was arrested at one of the Committee of 100 demonstrations in Trafalgar Square (he was briefly gaoled, refusing at first to pay the fine, but he found prison boring and eventually paid his way out). He was an accomplished musician, and he and Strachey would often end technical seminars by playing piano duets. But characteristically he was interested in understanding rather than performance: each new piece was a challenge to be surmounted then surpassed.

In 1964 he was 'brain drained' to the USA and moved with his wife and two small children to New York to work for Univac, then a major US computer manufacturer. No expense was spared: though they'd been put in a hotel he asked for a house with a garden, which is what he'd had in London, and they were given an approximation -- half a house in Greenwich Village. He published ***A correspondence between Algol 60 and Church's Lambda notation*** which drew out in greater detail and with greater fidelity the correspondences he'd shown already, which he later extended to the problematic notions of labels and jumps. In ***The next 700 programming languages*** he gave a witty account of how *all* programming languages of the time were just sugared versions of the lambda calculus, and he defined ISWIM (If You See What I Mean), an invented but unimplemented programming language which could be used to explain his ideas to those deterred by the austerity of mathematical notation. Like many of his advances, ISWIM was presented casually, but it was much more than a presentational device: it was the first programming language to be given a meaning independent of a particular computing machine, being defined by simple rules to translate it into the lambda calculus.

But he hated corporate work in a skyscraper and in 1966 he moved to MIT. It wasn't much better there. He was used to sharing his ideas with his London colleagues, in Mervyn's group and in loud pub discussions, but in

Cambridge Mass. people kept their ideas to themselves. They wouldn't listen to him very much: in particular they didn't take to his 'closure' idea, and as a result LISP, the first language to deal with list processing and functional values, got functions wrong in all its implementations for a decade and a half. Peter remarked that "the LISP they understood and the LISP I understood had little to do with one another".

In 1967 he was tempted back to London to take a chair at Queen Mary College, where he remained for the rest of his academic career. He involved himself into teaching and research but he didn't publish substantially again though there were a couple of papers on universal algebra and programming semantics, one with Rod Burstall, who he'd met originally in Mervyn's group, and who became a professor at Edinburgh. He worked with John Reynolds, who was his research assistant for a year and later became professor at CMU and a world-leading programming language theorist, and with Hans Bekic of the VDL/VDM group which was working on mathematical definitions of programming languages. He was always trying to explain his ideas about the connections between computing and fundamental mathematical logic to his colleagues and to students. His work continued to have influence: ISWIM profoundly affected the design and definition of ML, and through that even languages such as Java which underpin the World Wide Web.

But increasingly he seemed to be losing interest in computer science. Always bisexual, he threw himself into the politics of the Gay Liberation Front in the early 1970s. He drifted apart from his wife, and they separated amicably in 1973. For the rest of his life he pursued gay politics, and his house in Rona Road became a famous gay commune. He was always more of a facilitator than an activist, though he was once arrested on a GLF demonstration. His talent for inviting people to his dinner parties that he thought should meet hatched many plots. He was always a radical: AIDS the musical, with its chorus of WPCs in rubber gloves, was conceived as a protest against those he felt weren't pressing the gay cause as hard as he thought they should and his phrase "an iron fist in a rubber glove" a statement against police elements who played up the 'threat' of interactions with gay people. It was at one of his dinner parties that those who reinvigorated Gay Pride marches in the mid 80s met, just in time for the battle over clause 28. And he always kept up his music, playing and singing and attending Wagner operas.

Never a joiner, always a critic, always original and extreme, "annoying but lovable" as one of his colleagues once put it, he kept his academic work to an acceptable minimum but never stopped thinking and writing. Towards the end he became convinced that computing had perhaps been a bad idea, giving support to profit-taking corporate interests and a surveillance state, and that perhaps he'd wasted his energies in promoting it. But he never completely abandoned those of us who were still pursuing the subject, remaining a frequent visitor to BCS FACS meetings in London, even though he had little sympathy for formal reasoning and was opposed even to variable typing.

He is survived by his wife Hanne, his children Daniel and Louise and by his ideas which remain central to the study of the subject and which, although computing has outgrown the simply sequential languages of his youth, are constantly revisited by new researchers wanting to understand how to build on the foundations he laid.

(He needs a biography. I hope somebody will attempt one before all his contemporaries are dead. Olivier Danvy and Rod Burstall are looking through his papers, but they aren't biographers either.)

Richard Bornat

(Versions of this obituary appeared in the Guardian and in Formal Aspects of Computing Journal. Most of the information comes from Peter's friends and from being his colleague for a couple of decades. On a personal note, I tried to sack him once in the 1980s but, luckily, I was talked out of it: George Coulouris pointed out that Peter's contribution to computer science would outweigh mine no matter how long I worked, though he put it more politely than that. Even more luckily, Peter forgave me my aberration.)

**Book
Announcement**

We are pleased to announce the completion of our book based on the FACS seminars held between 2005 and 2007:

Formal Methods: State of the Art and New Directions

The online version is available at:

<http://www.springerlink.com/content/978-1-84882-735-6>

The print version can be pre-ordered from Springer:

<http://www.springer.com/computer/programming/book/978-1-84882-735-6>

Royalties will be ploughed back into the seminar series.

The book is dedicated to Peter Landin who sadly passed away in June.

I hope you enjoy the collection of papers

Paul Boca

Cross-Stitch Theorem
Sharon Curtis

Puzzle: Prove the Cross-Stitch Theorem
Sharon Curtis, Oxford Brookes University



Cross-stitch uses different coloured threads to form many 'X'-shaped stitches, arranged in an aesthetically pleasing pattern. The stitches are typically sewn on a woven fabric containing holes arranged in a convenient square grid pattern.

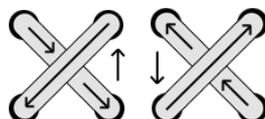
Define a *contiguous patch of colour* in a cross-stitch pattern to be a set of grid squares all stitched in the same colour that is *path-connected*. That is, any grid square in the set can be reached from any other square in the set by means of a path through other squares in the set, in which any two adjacent

squares along the path share a common boundary. For example, in the illustration, the snowman's nose forms a contiguous patch of orange. His mouth, on the other hand, is formed from three contiguous patches of black.

To ensure neatness of the finished work, cross-stitchers follow a rule for making the individual stitches: the two halves of the stitch must always be completed in the same order. Let us, say, set the rule that the \ half of a X stitch is always made before the / half,

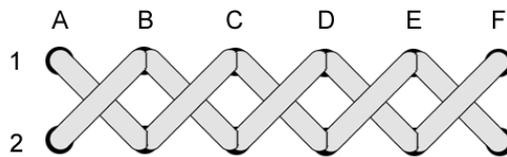
so stitches like this  are permissible, but not ones like this: 

Note that when making half of a stitch, the thread can go either way along the diagonal, so for example, two ways of sewing a whole stitch are:



Other ways of forming this stitch could also involve sewing the \ part of the stitch first, then sewing elsewhere on the fabric, returning to form the / part of the stitch later.

The amount of thread required to complete several stitches can vary. Consider this row of cross-stitching:



Taking the grid squares to be one unit wide, if the stitches are fashioned from left-to-right one X at a time (by taking the thread up \uparrow through the fabric at grid position A1, then down \downarrow through the fabric at B2, then \uparrow B1, \downarrow A2, \uparrow B1, \downarrow C2, \uparrow C1, \downarrow B2, \uparrow C1, \downarrow D2, \uparrow D1, \downarrow C2, \uparrow D1, \downarrow E2, \uparrow E1, \downarrow D2, \uparrow E1, \downarrow F2, \uparrow F1, \downarrow E2), then the row requires thread of length $5 + 14\sqrt{2}$. In practice there will be some extra thread to start with before \uparrow A1, and some extra thread to finish with after \downarrow E2, but we will ignore this extra thread for starting and finishing a section of cross-stitch.

If, however, the \ halves of stitches are made first, then the / halves, in the order \uparrow A1, \downarrow B2, \uparrow B1, \downarrow C2, \uparrow C1, \downarrow D2, \uparrow D1, \downarrow E2, \uparrow E1, \downarrow F2, \uparrow F1, \downarrow E2, \uparrow E1, \downarrow D2, \uparrow D1, \downarrow C2, \uparrow C1, \downarrow B2, \uparrow B1, \downarrow A2, then the row only requires thread of length $9 + 10\sqrt{2}$, a saving of $4(\sqrt{2} - 1)$ units of thread.

Your challenge is to prove the following cross stitch theorem:

A contiguous patch of colour containing n squares of unit width can be cross-stitched using thread of length $2n - 1 + 2n\sqrt{2}$.

Another way of thinking of this theorem is this: it is possible to cross-stitch a contiguous patch of colour in such a way that when the thread is passed down through the fabric, it always comes up again at an adjacent hole exactly 1 unit away. Note that the thread can't come back up through the same hole it just went down, otherwise the thread will fall back through the hole and won't form the stitch properly.

Hence or otherwise,

Devise an algorithm to produce a suitable sequence of stitches for cross-stitching the contiguous patch using thread of length $2n - 1 + 2n\sqrt{2}$.

You get bonus points if your algorithm is realistically implementable by a cross-stitcher in their head (i.e. it doesn't require a great feat of memory to figure out which bit to stitch next).

Disclaimers for the upcoming holiday season:

If you do cross-stitch yourself, the author refuses to take responsibility if you get all frustrated because you have managed to deviate from your algorithm and used a whole millimetre of thread more than was strictly necessary.

Alternatively, if you have friends or family who like doing cross-stitch and you take it upon yourself to tell them they are doing it all wrong and using far too much thread, then the author takes no responsibility for any family rows or injuries you may suffer from as a result.

Author's Note:

I am not the only one to notice this cross-stitching result. The topic of cross-stitching a region with a minimal amount of thread was also presented by Barbara Ashton and Kevin Dove in a talk *Cross Stitching, Graph Theory and a Least Path Problem* that they gave at the joint meetings of the AMS and MAA in New Orleans in January 2007.

:

Tools Announcement:
Z through Pictures for Word users and others
Anthony Hall

Anthony Hall has sent us the following information about a new tool:

I have just released the latest version (1.3) of the Z Word Tools. You can now draw diagrams of your specification structure showing schema inclusion and use of global variables. This feature can be used stand-alone, so you can draw diagrams from LaTeX files even if you are not a Word user.

The Z Word Tools are available on SourceForge at <http://sourceforge.net/projects/zwordtools/> where you can find more details

FACS Evening Seminar Series 2009–2010

1. *Software Verification for Ubiquitous Computing* by Marta Kwiatkowska

Report by Margaret M. West



The first seminar of the year was joint with BCS-Women and took place on 18th October 2009 at the London offices of the BCS. It was given by Marta Kwiatkowska who is Professor of Computing Systems and Fellow of Trinity College, University of Oxford; she has recently been awarded a highly competitive European Research Council (ERC) Advanced Investigators Grant for a new 5-year project *Veriware*.

Ubiquitous computing systems are now widespread in for example intelligent buildings, environmental monitoring, healthcare monitoring and automotive software. They have been termed 'everyware' by Adam Greenfield: they continuously interact with the environment through sensors and actuators. The failure of one of these systems can result in the cost of a large sum of money and in the worst case, loss of life. A real-life example of system failure given by the speaker was that of the Toyota Prius resulting in sudden stalling or stopping of vehicles at high speed, due to a software glitch in a control system. This failure resulted in the company recalling about 75,000 cars in May 2005.

The talk focussed on PRISM, a probabilistic model checker aimed at systems which exhibit stochastic behaviour. PRISM was first developed at the University of Birmingham and latterly at the University of Oxford. A PRISM model typically includes sensors and actuators and comprises a set of modules, each of which captures the behaviour of different components of the system. The transitional behaviour of a module is modelled by a set of guarded commands which give the change in state of module variables at a specified rate. Synchronisation is possible using labelled commands and PRISM has been used to analyse diverse systems such as those in communication and security.

For more information see the PRISM website:

<http://www.prismmodelchecker.org>

BCS-women is a group for female members of the British Computer Society (BCS) – follow this link to see what they do: <http://www.bcs.org/bcswomen>

2. Forward with Hoare **by Mike Gordon**

Our second evening seminar was given by Professor Mike Gordon, FRS of the University of Cambridge on December 1st 2009 at De Morgan House, Russell Square, Bloomsbury. The abstract is presented here – in the next issue we hope to include a report.

Abstract:

Hoare's celebrated paper entitled "*An Axiomatic Basis for Computer Programming*" appeared in 1969, so the Hoare formula $P\{S\}Q$ is now forty years old! That paper introduced Hoare Logic, which is still the basis for program verification today, but is now mechanised inside sophisticated verification systems. My talk aims to give an accessible introduction to methods for proving Hoare formulae based both on the forward computation of postconditions and on the backwards computation of preconditions. Although precondition methods are better known, computing postconditions provides a verification framework that encompasses methods ranging from symbolic execution to full deductive proof of correctness.

FACS FACTS Issues in 2010

Call for Submissions

We welcome contributions for the next issue of *FACS FACTS*, in particular:

- Letters to the Editor
- Conference reports
- Reports on funded projects and initiatives
- Calls for papers
- Workshop announcements
- Seminar announcements
- Formal methods websites of interest
- Abstracts of PhD theses in the formal methods area
- Formal methods anecdotes
- Formal methods activities around the world
- Formal methods success stories
- News from formal methods-related organizations
- Experiences of using formal methods tools
- Novel applications of formal methods
- Technical articles
- Tutorials
- Book announcements
- Book reviews
- Adverts for upcoming conferences
- Job adverts
- Puzzles and light-hearted items

Please send your submissions (in Microsoft Word, LaTeX or plain text) to Margaret West [editor@facsfacts.info], the Newsletter Editor.

If you would like to be an official *FACS FACTS* reporter or a guest columnist, please contact the Editor.

Recent and Forthcoming Events

BCS FACS Seminars: Unless stated otherwise, these take place at: BCS London Offices, First Floor, The Davidson Building, 5 Southampton Street, London WC2E 7HA.

October 2009

Marta Kwiatkowska
Software verification for ubiquitous computing
 October 19th 2009
 Joint meeting with BCS-Women

November 2009

FM2009
 November 2nd – 6th 2009
 Eindhoven, The Netherlands

December 2009

Mike Gordon
Forward with Hoare
 December 1st 2009
 Joint Meeting with LMS held at De Morgan House

January 2010

Mike Hinchey
Forward with Hoare
 January 28th 2010

February 2010

ABZ 2010
 February 22nd – 25th 2010
 Orford, Quebec, Canada
<http://abzconference.org>

March 2010

Mike Holcombe
Formal methods, supercomputers and simulation – understanding complex biological, economic and social systems
 March 10th 2010

For further conference announcements, please visit the **Formal Methods Europe** (FME) website [<http://www.fmeurope.org>], the **EATCS** website [<http://www.eatcs.org>] and the **Formal Methods Wiki** Virtual Library [<http://formalmethods.wikia.com/wiki/Meetings>].

FACS Committee



Formal Aspects of Computing
Science Specialist Group



Jawed Siddiqi
FACS Chair



Jonathan Bowen
FACS Treasurer
and ZUG Liaison



Paul Boca
Secretary



Roger Carsley
Minutes Secretary



John Cooke
FAC Journal Liaison
and BCS Liaison



John Fitzgerald
FME Liaison and
SCSC Liaison



Margaret West
Newsletter Editor



Rob Hierons
Chair, Formal Methods
and Testing Subgroup



John Derrick
Chair, Refinement
Subgroup

External Liaison



Tom Melham
LMS Liaison Officer

FACS is always interested to hear from its members and keen to recruit additional helpers. Presently we have vacancies for officers to help with fund raising, to liaise with other specialist groups such as the Requirements Engineering group and the European Association for Theoretical Computer Science (EATCS), and to maintain the FACS website. If you are able to help, please contact the FACS Chair, Professor Jawed Siddiqi at the contact points below:

BCS FACS
c/o Professor Jawed Siddiqi (Chair)
Sheffield Hallam University
E info@bcs-facs.org.uk
W www.bcs-facs.org

You can also contact the other Committee members via this email address.

Please feel free to discuss any ideas you have for FACS or voice any opinions openly on the FACS mailing list [FACS@jiscmail.ac.uk]. You can also use this list to pose questions and to make contact with other members working in your area. Note: only FACS members can post to the list; archives are accessible to everyone at <http://www.jiscmail.ac.uk/lists/facs.html>.

Coming Soon in FACS FACTS...

Conference reports

Details of upcoming FACS Evening Seminars

And more...