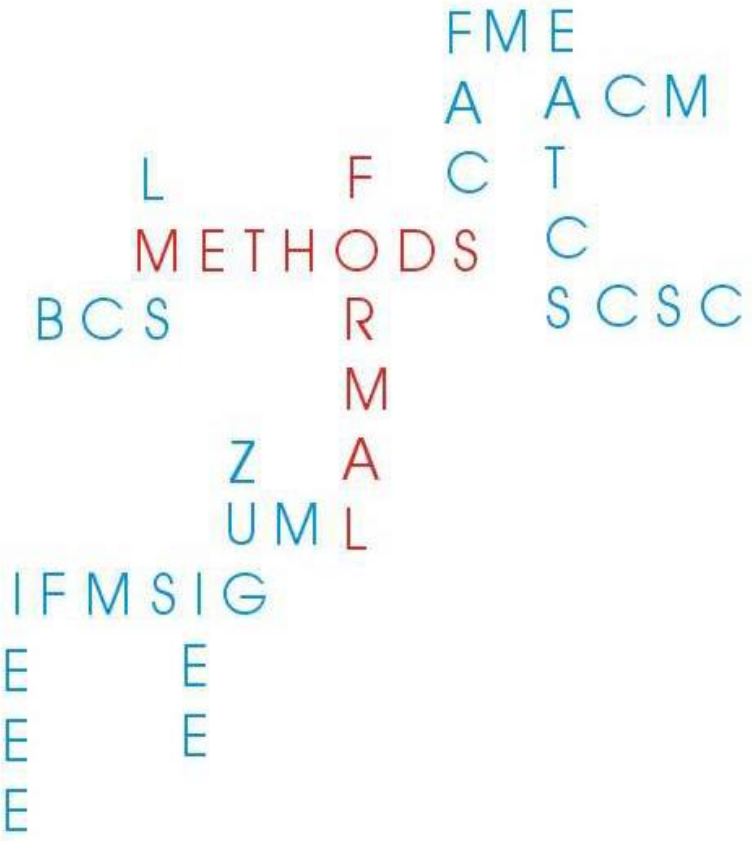
Issue 2014-1

August 2014

FACS

A C T S



bcs The Chartered Institute for IT
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 BCS FACS website for further details (see http://www.bcs.org/category/12461).

Back issues of *FACS FACTS* are available for download from:
http://www.bcs.org/content/conWebDoc/33135

The FACS FACTS Team

Newsletter Editors Tim Denvir timdenvir@bcs.org
Brian Monahan briangmonahan@googlemail.com

Editorial Team Jonathan Bowen, Tim Denvir. Brian Monahan, Margaret West.

Contributors to this Issue

Jonathan Bowen, Tim Denvir, Eerke Boiten, Rob Heirons,

Azalea Raad, Andrew Robinson.

BCS-FACS websites

BCS: http://www.bcs-facs.org

LinkedIn: http://www.linkedin.com/groups?gid=2427579

Facebook: http://www.facebook.com/pages/BCS-FACS/120243984688255

Wikipedia: http://en.wikipedia.org/wiki/BCS-FACS

If you have any questions about BCS-FACS, please send these to Paul Boca <paul.boca@googlemail.com>

Editorial

Welcome to issue 2014-1 of *FACS FACTS*. This is the first issue produced by your new joint editors, **Tim Denvir** and **Brian Monahan**.

One effect of the maturity of formal methods is that researchers in the topic regularly grow old and expire. Rather than fill the issue with Obituaries, we have taken the course of reporting on most of these sad events in brief, with references to fuller obituaries that can be found elsewhere, in particular in the *FAC Journal*. A full Obituary is given in this issue for Professor Wlad Turski and Obituaries in Brief for John C. Reynolds, Kaisa Sere, Lockwood Morris, and Heinz Zemanek.

In the rest of this issue you will find a review by Andrew Robinson and Jonathan Bowen of Mathai Joseph's book: *Digital Republic: India's Rise to IT Power*. Following that there are reports on *FACS* evening seminars that have taken place so far this year. Azalea Raad reports on the joint *LMS-FACS* seminar, *Computational Reasoning for Computer Programs,* given by Philippa Gardner. Rob Heirons reports on his own seminar, *Asynchronous Software Testing*. Eerke Boiten reports on the EPSRC-funded Network of Excellence, *Cryptoforma* and the joint *Cryptoforma* and *FACS* afternoon event which ended with an evening seminar by Cédric Fourner. Finally, Jonathan Bowen gives summaries of two other recent seminars and announcements of a forthcoming joint evening seminars with *LMS* on 22nd October and the annual Peter Landin Semantics Seminar on 8th December, to be given by Peter Mosses.

In the last issue 2013-1 of *FACS FACTS*, Margaret West wrote about a meeting at the House of Lords, *UK Science Education in the 21st Century: Reporting Turing's Legacy*. She adds a postscript to that report:

> For some time now there has been interest in the US in teaching children to code. This is an article from the NY Times about the subject:
>
> http://www.nytimes.com/2014/05/11/us/reading-writing-arithmetic-and-lately-coding.html?_r=0

The next *FACS* AGM is planned for 4.00 p.m. at the BCS offices on December 8th, immediately before the Peter Landin Semantics Seminar. More announcements of this will follow in due course.

The *FACS* committee and, I am sure, membership, would like to congratulate Professor Sir Tony Hoare on his 80th birthday earlier this year, and Professor Cliff Jones on his 70th birthday, which was also this year. Our best wishes to each of them.

Many of *FACS* seminars take place in the offices of the British Computer Society in the Davidson Building, Southampton Street. These excellent facilities are conveniently situated in Central London close to Covent Garden and we would like to thank the *BCS* for making them available to us.

Recent and Forthcoming Events

Reported by: Jonathan Bowen

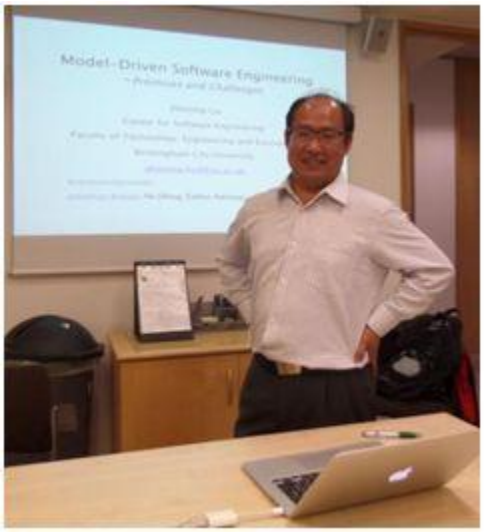
To date during 2014, FACS has held the following evening seminars at the BCS offices in London:

Wednesday 15th January 2014

Joint Event with CryptoForma network.

Reported on below by Eerke Boiten.

Thursday 15th May 2014



Model-Driven Software Engineering – Promises and Challenges

Prof. Zhiming Liu
http://www.bcu.ac.uk/tee/ctn/our-staff/zhiming-liu
Birmingham City University

Abstract: Software engineering was born with the "software crisis", caused by the increasing power of computers and growing complexity of software development. Formal methods have been developed to tackle the grand challenge of correctness and dependability of complex software. Today, software programs are everywhere, but they are deployed on different platforms and embedded in different kinds of devices. These platforms and devices are distributed and connected with different networks. Consider smarter meters networked with industry control systems and mobile phone networks for home automation, and eHealth or mHealth. These applications of the "Internet of Things", "smart cities" and "cyber-physical systems" (CPS) cannot be designed top-down from scratch or synthesised bottom-up from existing pieces. They have to evolve incrementally from existing systems, even non-computerised systems, and be constantly maintained with updated technologies. In this talk, we discuss model-driven software engineering, which shows promises and challenges in dealing with the complexity of these systems and in support of their evolution. In particular, we argue that an architecture-centric modelling method could be the hope to scale up formal method by linking theories to system engineering practice.

Thursday 5th June 2014

Alan Turing: The Founder of Computer Science

Prof. Jonathan P. Bowen
http://www.jpbowen.com
Birmingham City University

Abstract: Alan Mathison Turing (1912–1954), OBE, FRS, has a rightful claim to the title of father of modern computing. He laid the theoretical groundwork for a universal machine that models a computer in its most general form before World War II. During the war, Turing was instrumental in developing and influencing actual computing devices that have been said to have shortened the war by up to two years by decoding encrypted enemy messages that were believed by others to be unbreakable. Unlike

some theoreticians, he was willing to be involved with practical aspects, and was as happy to wield a soldering iron as he was to wrestle with a mathematical problem, normally from a unique angle compared to others.

This talk explored Alan Turing's life, achievements, and legacy, in the month of the 60th anniversary after his death (on 7th June 1954).

Slides for all the presentations above are available on the BCS-FACS website in the past events section: http://www.bcs.org/content/ConWebDoc/51924

17th July 2014

Asynchronous Software Testing

Professor Rob Heirons

Reported on below by Rob Heirons.

Further events are planned later in the year, including a joint evening seminar with the London Mathematical Society (LMS), organised by John Cooke:

Wednesday 22nd October 2014

London Mathematical Society, De Morgan House, 57–58 Russell Square, London

Prof. Joel Ouaknine
http://www.cs.ox.ac.uk/people/joel.ouaknine/home.html
University of Oxford

Abstract: Linear recurrence sequences (LRS), such as the Fibonacci numbers, permeate vast areas of mathematics and computer science. In this talk, Professor Ouaknine considers three natural decision problems for LRS, namely the Skolem Problem (does a given LRS have a zero?), the Positivity Problem (are all terms of a given LRS positive?), and the Ultimate Positivity Problem (are all but finitely many terms

of a given LRS positive?). Such problems (and assorted variants) have applications in a wide array of scientific areas, such as theoretical biology (analysis of Lsystems, population dynamics), economics (stability of supplyanddemand equilibria in cyclical markets, multiplier-accelerator models), software verification (termination of linear programs), probabilistic model checking (reachability and approximation in Markov chains, stochastic logics), quantum computing (threshold problems for quantum automata), discrete linear dynamical systems (reachability and invariance problems), as well as combinatorics, statistical physics, formal languages, etc. Perhaps surprisingly, the study of decision problems for LRS involves advanced techniques from a variety of mathematical fields, including analytic and algebraic number theory, Diophantine geometry, and real algebraic geometry.

Thursday 25 September

The CakeML verified compiler

Scott Owens (University of Kent)

Abstract: CakeML is a new ML dialect aimed at supporting formally verified programs. The CakeML project has several aspects including formal semantics and metatheory, a verified compiler, a formal connection between its semantics and higher-order logic (in the HOL4 interactive theorem prover), and example verified applications written in CakeML and HOL4. The project is an active collaboration between Ramana Kumar and Magnus Myreen at Cambridge, Michael Norrish at NICTA, and myself.

In this talk, I will explain the architecture of CakeML's verified compiler, and then show how we move verified functions from a theorem prover to CakeML (including bootstrapping the compiler itself). Lastly, I will explain how these tools will allow us to support -- with a very small trusted computing base – a verified version of the HOL Light theorem prover.

CakeML's web site is https://cakeml.org, and development is hosted on GitHub at https://github.com/CakeML/cakeml.

8th December 2014

The annual **Peter Landin Seminar** will be by **Prof. Peter Mosses** of University of Swansea, on ***Correspondences between Programming Languages and Semantic Notations***, organised by Paul Boca.

Abstract: 50 years ago, at the IFIP Working Conference on Formal Language Description Languages, Peter Landin presented a paper on "A formal description of ALGOL 60". In it, he explained "a correspondence between certain features of current programming languages and a modified form of Church's λ-notation", and suggested using that as the basis for formal semantics. He regarded his formal description of ALGOL 60 as a "compiler" from ALGOL abstract syntax to λ-notation.

10 years later, denotational semantics was well established, and two denotational descriptions of ALGOL 60 had been produced as case studies: one in the VDM style developed at IBM-Vienna, the other in the continuations-based style adopted in Christopher Strachey's Programming Research Group at Oxford.

After recalling Landin's approach, I'll illustrate how it differs from denotational semantics, based on the ALGOL 60 descriptions. I'll also present a recently developed component-based semantics for ALGOL 60, involving its translation to an open-ended collection of so-called fundamental constructs. I'll assume familiarity with the main concepts of denotational semantics.

Also on 8th December, immediately preceding the Peter Landin Semantics seminar, *FACS* will hold its AGM. Announcements in due course.

9 – 10th March 2015

A two-day workshop will be held celebrating 25 years since the start of the ESPRIT ProCoS Project [1] on ***Provably Correct Systems*** (followed by the ProCoS II Project [2,3] and ProCoS Working Group [4,5]) is planned, organized by Prof. Jonathan Bowen of Birmingham City University, Prof. Mike Hinchey of LERO, University of Limerick (Ireland), and Prof. Dr. Ernst-Rüdiger Olderog of Universität Oldenburg (Germany).

Information on these and other future events will appear on the BCS-FACS website in due course under: http://www.bcs.org/category/12468

References

1 Dines Bjørner, C.A.R. Hoare, Jonathan P. Bowen, He Jifeng, Hans Langmaack, Ernst-Rüdiger Olderog, Ursula Martin, Victoria Stavridou, Fleming Nielson, Hanne Riis Nielson, Howard Barringer, Doug Edwards, Hans Henrik Løvengreen, Anders Ravn and Hans Rischel, A ProCoS Project Description: ESPRIT BRA 3104. *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, 39, pages 60–73, October 1989.

2 Jonathan P. Bowen et al., A ProCoS II Project Description: ESPRIT Basic Research project 7071. *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, 50, pages 128–137, June 1993.

3 Jonathan P. Bowen, C.A.R. Hoare, Hans Langmaack, Ernst-Rüdiger Olderog and Anders P. Ravn, A ProCoS II Project Final Report: ESPRIT Basic Research project 7071, *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, 59:76–99, June 1996.

4 Jonathan P. Bowen et al., A ProCoS-WG Working Group Description: ESPRIT Basic Research 8694, Bulletin of the European Association for Theoretical Computer Science (EATCS), 53, pages 136–145, June 1994.

5 Jonathan P. Bowen, C.A.R. Hoare, Hans Langmaack, Ernst-Rüdiger Olderog and Anders P. Ravn, A ProCoS-WG Working Group Final Report: ESPRIT Working Group 8694. *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, 64:63–72, February 1998.

Obituary: Wladyslaw Turski

Professor Wladyslaw Marek Turski came from an academic family. His father, Stanislaw Turski, was rector of the University of Warsaw, his brother Łukasz a physicist and his son Martin a computer scientist. Wlad, as he was known to all his colleagues, was a charismatic and academically significant figure in computer science. Fewer of his colleagues may know that Wlad Turski began his academic life as an astronomer. He graduated in astronomy from the Lomonosov (Moscow State) University in 1960, and continued to an MSc in celestial mechanics. His topic was the mechanics of lunar landing, a problem anticipating any actual lunar landings by some seven years, which required a deal of computation, at that time done by hand.

Immediately after his MSc Wlad Turski took up a one-year Leverhulme Research Fellowship, funded by the British Council, at the Jodrell Bank radio astronomy observatory in Cheshire, England. This was to be the first of several significant British connections. There he investigated the phenomenon of the highly erratic behaviour of the Giacobinids, the meteor shower associated with the comet Giacobini-Zinner, using

the Manchester University Mercury computer to analyse the observed data. By the summer of 1961 he had solved the Giacobinids problem by numerical simulation and applied the same technique to other astronomical observations.

At the end of his fellowship at Jodrell Bank Wlad Turski returned to Warsaw and took up a post in the new Computation Centre of the Polish Academy of Sciences. When the Centre took delivery of a URAL-2 computer in May 1962, Turski took charge of enabling general academic access to the computer. Heading a team of three, Wlad produced a symbolic assembly language, CLIP, and in that same year gave a paper on its implementation to the ACM National Conference in the USA. Also in 1962 he received a doctorate in Mathematical and Physical Sciences from the University of Warsaw and in 1966 a doctorate in Technical Sciences from the Academy of Mining and Metallurgy, University of Krakow.

In 1966 he took part in constructing a system for analysing astronomical data from observatories in the ambit of CMEA (Council for Mutual Economic Assistance), and gave a paper to the Committee of Space Research in Warsaw. There A. Van Wijngaarden invited him to join IFIP WG2.1, of which he later became secretary. Subsequently, through these contacts, Stanley Gill invited him to Imperial College, London where Wlad lectured on compiler construction.

When he returned to Poland a year later he was one of a team of three who designed and implemented the SODA operating system for the ODRA 1204 computer, the last mainframe to be entirely designed and built in Poland. This work was under the auspices of a collaboration between the Polish Academy of Sciences and Elwro, the Polish manufacturer of the URAL-2 and ODRA computers. The SODA operating system could supervise concurrent communicating programs with batch programs running in the background. In the end this operating system was never deployed; Elwro replaced the computer with the ODRA 1300 series, which was compatible with ICL 1900 machines. In 1968 he spent some time in the USA serving on the committee that produced *Curriculum 68: Recommendations for academic programs in computer science: a report of the ACM curriculum committee on computer science*.

After the success of ALGOL 60, IFIP WG2.1 sought to define a new language that was ultimately published as ALGOL 68. Wlad Turski was among those members of WG2.1 who disagreed with aspects of the new language and its description, and was one of the signatories of a minority report, all of whom broke away to form the new WG2.3 on program methodology. Wlad was a founding member of the Polish Information Processing Society and its first president from 1981 to 1987. He co-founded the Institute of Computer Science at the University of Warsaw and was Dean of Faculty 1996-1999. He was a member of the Polish Mathematical Society, the Polish Astronomical Society, a Fellow of the UK Royal Academy of Engineering and a member of the British Computer Society. The BCS made him a Distinguished Fellow, an honour awarded only twenty-seven times over the last forty years. Other holders of the title include Edsger Dijkstra, Maurice Wilkes, Tom Kilburn, Tony Hoare, Donald Knuth, Robin Milner and Tim Berners-Lee.

Turski's writings ranged widely, from columns in popular IT magazines to tutorial works and academic papers. He even dipped his creative toes into the topic of Software Science. He is the author of over half a dozen books on mathematics, computer science and programming methodology.

A more comprehensive obituary will be found in *Formal Aspects of Computing*, Vol. 26, no. 5.

Wladislaw Marek Turski, born 17th October 1938, died 18th July 2013.

Selected references

Safety, Security and Dependability in Crowd Computing, Wladislaw M. Turski, in Dependable and Historic Computing, 2011.

It was Fun, Wladislaw M. Turski, in Information Processing Letters, 88 (2003) 7-12.

The Reference Model for Smooth Growth of Software Systems Revisited, Wladislaw M. Turski, in IEEE Transactions on software Engineering, August 2002 (vol. 28 no. 8) pp. 14-15.

Essay on Software Engineering at the Turn of Century, in *Fundamental Approaches to Software Engineering*, LNCS 1783 (2000), pp 1-20.

Wladislaw M. Turski and Thomas S. E. Maibaum, *The Specification of Computer Programs*. Addison-Wesley Publishing Company, 1987.

Wladislaw M. Turski. *A Model for Data Structures and Its Applications*. I. Acta Inf. 1, pp. 26-34, 1971.

William F. Atchison, Samuel D. Conte, John W. Hamblen, Thomas E. Hull, Thomas A. Keenan, William B. Kehl, Edward J. Mccluskey, Silvio O. Navarro, Werner C. Rheinboldt, Earl J. Schweppe, William Viavant, David M. Young. *Curriculum 68: Recommendations for academic programs in computer science: a report of the ACM curriculum committee on computer science* . "Communications of the Association for Computing Machinery." 11, pp. 151-197, 3 1968.

Tim Denvir
timdenvir@bcs.org

Obituaries in Brief

John C. Reynolds, 1935-2013, graduated from Purdue University and gained a PhD in Theoretical Physics from Harvard University in 1961. While studying for his PhD he developed an intense interest in computer science and subsequently held positions at Argonne National Laboratory, Stamford University and the University of Chicago. He was a Professor of Computer and Information Science at Syracuse University from 1970-1986 and finally Professor of Computer Science at CMU. He held visiting positions at Aarhus University (Denmark), University of Edinburgh, Imperial College London and Queen Mary, University of London. John Reynolds' principal research contributions were in the area of program language semantics and specification languages. He devised the Polymorphic Lambda Calculus (independently discovered by Jean-Yves Girard). He applied Category Theory to Program Language Semantics and worked on Parametric Polymorphism, Intersection Types, Separation Logic, Definitional Interpreters, which used the concept of Continuations, and much else. John Reynolds was a former editor of Communications of the ACM and Journal of the ACM. He was given the ACM SIGPLAN Program Language Achievement Award in 2003, CMU's Dana Scott Distinguished Research Career Award in 2006, and awarded an honorary Doctor of Science from Queen Mary, University of London in 2007 and the British Computer Society Lovelace Medal in 2010. He gave the *Peter Landin Annual Semantics Seminar* to BCS FACS in London on 6th December 2010, his title being *Toward a Grainless Semantics for Shared-Variable Concurrency*. A memorial paper, The Essence of Reynolds, can be found in *Formal Aspects of Computing*, Vol. 26, pp 435-439. John Reynolds passed away on 28th April 2013.

Kaisa Sere, 1954-2012, gained an MSc in Mathematics and a PhD in Computer Science from Åbo Akademi University, Finland. She held academic positions at Ohio State University (USA), the University of Utrecht (Netherlands), the University of Kuopio (Finland), the Academy of Finland, finally being a vice-rector at Åbo Akademi. She made major research contributions in the field of *Refinement* of parallel and distributed systems, in particular in the framework

of Action Systems, working closely with R-J Back. Kaisa Sere also adapted the tool support for J-R Abrial's B method, resulting in the B Action Systems framework. A comprehensive obituary can be found in Formal Aspects of Computing, Vol. 26, pp 197-201, from which this short note is derived. She passed away after a long illness on December 5th 2012.

F. Lockwood Morris, 1943-2014, graduated in Mathematics from Harvard University and gained a PhD in Computer Science from Stanford University. He was a professor of Computer Science at Syracuse University for 39 years. He worked for some time on a project to turn the early sections of Mac Lane's *Categories for the Working Mathematician* into HOL. His research interests included Functional Programming, Types and Polymorphism, Combinatorial Games and Computational Geometry, but most recently focussed on Category Theory, especially applied to data structures, and machine assisted mathematical proofs and theorem processing, specifically with HOL. In early days Lockwood Morris was one of the inventors of Continuations, and in his paper *The Discoveries of Continuations* in *LISP AND SYMBOLIC COMPUTATION*, 1993, John Reynolds (see above) wrote:

> [I]...had the good fortune to attend Morris's talk. It was my first exposure to the use of continuations, and to the fact that there are many styles of definitional interpreters, varying in abstractness and degree of circularity. These ideas were the genesis of my own work on definitional interpreters, which eventually did much to popularize continuations.

Lockwood Morris spent several sabbatical periods in the U.K., at the Universities of Essex, Oxford and Manchester. It is clear from many traces from meetings, conferences and on-line discussions that Lockwood Morris had an influence among many of his contemporaries. He died on 12 March, 2014.

Heinz Zemanek, 1920-2014, graduated from the Vienna University of Technology and gained a doctorate there in 1951 on the topic of time-sharing methods in multiplex telegraphy. He developed the first completely transistorised computer in Europe in 1955, the "Mailüfterl" ("May Breeze", a reference to "Whirlwind", an earlier computer developed at MIT), which is now in the Technical Museum of Vienna. He founded the IBM Vienna Laboratory in 1961, where he was its first Director until 1976. He initiated the renowned series of IFIP "Working Conferences" and the one in 1964 at Baden-bei-Wien ("Formal Language Description Languages") brought together the key people thinking about this topic. This was a springboard for his colleagues at the IBM Vienna Lab to produce the formal definition of the *PL/I* programming language. It used *VDL*, the *Vienna Definition Language*, a means of describing an abstract interpreter. The *PL/I* definition work led on in due course to *VDM*, the *Vienna Development Method*. From 1947 onwards Heinz Zemanek held academic appointments at the Vienna University of Technology, where a lecture hall was named after him. He was a long-term member of IFIP, and its President from 1971 to 1974. Zemanek was awarded the Austrian Cross of Honour for Science and Art, the Joseph Johann Ritter von Prechtl Medal and the Rudolf Kompfner Medal by the Vienna University of Technology, the Leonardo da Vinci Medal of the European Society for Education of Engineers, and many other honours. Other biographies can be found courtesy of the IEEE, and Wikipedia. Heinz Zemanek died on 16th July 2014.

Tim Denvir
timdenvir@bcs.org

Book Review

Digital Republic: India's Rise to IT Power

By Mathai Joseph

ISBN 978 93 82792 57 4
Power Publishers, 2013

http://www.amazon.co.uk/Digital-Republic-Indias-Rise-Power-ebook/dp/B00CGR5JLU

Review by Andrew Robinson and
Jonathan P. Bowen

Despite the unparalleled success since the 1990s of India's IT industry—today valued at $100 billion—there are surprisingly few good books about it, and even fewer about the history of computing in India: perhaps only journalist Dinesh Sharma's *The Long Revolution: The Birth and Growth of India's IT Industry* (2009). *Digital Republic: India's Rise to IT Power*, a briefer book than Sharma's, which is both a history and a personal memoir by a significant Indian computer scientist and formal methods researcher, Mathai Joseph, is therefore required reading for anyone trying to understand the development of digital computing in India from its shaky beginnings in 1955. Not only has its author worked at several key institutions—including Cambridge University in the UK during the 1960s, the Tara Institute of Fundamental Research in Mumbai and Tata Consultancy Services in Pune (where he retired in 2007 as head of research)—he also writes with directness, candour and variety, telling pointed and often amusing anecdotes about research and life in India, the UK and the USA from the 1950s until now.

Neither the Indian government nor Indian academia has served computing well. 'Of all the industries expected to create a technological transformation in India, this was the one least likely to succeed', Joseph writes. Soon after Independence in 1947, the government began to lavish attention on space and nuclear power but it controlled computers as if they were a 'danger' to employment, without the least vision of their ultimate importance. The situation became so discouraging for Joseph by 1985 that he

abandoned his secure position in Mumbai and moved to the University of Warwick until 1997. The reason for the software industry's 1990s success was precisely that software did not fall under any established government categorization. For a long time, it 'was not even recognized as an industry.'

But as he readily concedes, this commercial expansion has yet to invigorate either India R&D in IT or academic computer science, where the research community has remained virtually the same size for past two decades and 'has produced only a few notable successes', such as the internationally award-winning AKS primality test by Manindra Agrawal et al. to determine prime numbers. *Digital Republic* adds constructively to this debate and should help, we hope, to develop and improve computer science in India.

Views: Compositional Reasoning for Computer Programs

Professor Philippa Gardner

Reported by: Azalea Raad

The 2013 LMS/BCS-FACS Seminar was held on Tuesday 8th of October 2013 at the London Mathematical Society where Professor Philippa Gardner of Imperial College London gave a talk titled “Views: Compositional Reasoning for Computer Programs”. Amongst members of the audience were Professor Gardner’s research group, colleagues and collaborators, as well as prominent program verification researchers such as Sir Tony Hoare FRS FREng, Peter O'Hearn and Matthew Parkinson.

Professor Gardner opened her talk by introducing her research group and collaborators who contributed towards the material presented at the seminar. She gave a brief history of concurrent program verification and praised the efforts of those most influential in the development of the field.

She motivated her talk by stressing the importance of abstraction in understanding and reasoning about large computer systems. She considered the various levels of abstraction through an example of a client program used for manipulating mathematical sets. She contrasted the intuitive mathematical notion of a set against a possible heap-based implementation and its representation at machine level (captured by zeros and ones in memory).

She then turned to compositional abstraction, which underlies many reasoning principles for concurrent (multi-threaded) programs. She described how a concurrent environment is abstracted in order to reason about a thread in isolation without having to know the precise concurrent context in which it will be placed. In particular, she extended the set example to reason about a program where two separate threads compete to remove the same element from a set. She then pointed out how this approach can be generalised and how these abstractions are composed to reason about a program consisting of many threads. She emphasised the significance of such

modularity for reasoning about incomplete code or libraries where the context is not known.

She subsequently focused on the "Views" Framework and explained how it brings together the core principles underlying compositional reasoning systems for concurrent programs. She gave an intuitive account of various ingredients and requirements of a concurrent system:

> A thread's view consists of abstract knowledge about the current state of the machine and the thread's rights to change the state of the machine. The knowledge of a thread must be stable under the operations of concurrent threads. That is, no other thread may have rights to invalidate the thread's knowledge. Conversely, no thread can have knowledge that another thread has the right to invalidate. Views are compositional; in other words, knowledge and rights may be distributed between threads and recombined.

Finally, she gave a formal definition of the generic concurrent program logic provided by the Views Framework and how it embodies the essential ingredients for sound compositional reasoning:

- A set of views (representing the *abstract* state of a thread) ranged over by *p*; *q*; ... with a composition operator (denoted *)
- A set of machine states (representing the concrete state of a thread)
- A set of atomic commands together with their semantics (behaviour at machine level) and axiomatisation (behaviour at abstract level)
- A reification function relating abstract views to concrete machine states
- An axiom soundness property ensuring that the axioms for atomic commands are sound with respect to the reification, in the context of an arbitrary environment view. As such, compositionality is embedded in the meaning of 'program C

updates the view from p to q': for all environment views r, **C** must update $p * r$ to $q * r$.

She clarified the various ingredients of a compositional reasoning system through a simplified example of a concurrent tree library used for representing and manipulating HTML objects. She concluded her talk by highlighting the strength of the Views Framework in providing a general framework in which a wide variety of compositional reasoning approaches can be constructed and proved sound by appealing to its general soundness result.

Asynchronous Software Testing

Professor Rob Heirons

17 July, 2014

The talk concerned model-based testing (MBT), where testing is automated on the basis of a formal model of the system under test (SUT) or some aspect of this. It assumed that the model was an input output transition system (IOTS): a labelled transition system in which we distinguish between input and output. It was noted that the usual approach to MBT represents testing as the tester and the SUT synchronising on common actions. It is then possible to apply standard automata theory methods to solve test automation problems such as finding a test to reach a state, finding a test to distinguish two states, and checking an observation against the model. However, communications may be asynchronous due to there being a network between the tester and the SUT or the test tool buffering output.

One approach to adapting MBT to asynchronous communications is simply to model the communications channels and this is a very general solution. However, this involves composing the original model M will an infinite state model (or an exponentially large model if channels are bounded). The talk thus looked at approaches that do not require one to explicitly model the channels. A key issue is that the trace (sequence of inputs and outputs) observed by the tester will be that produced by the SUT but potentially with some reordering of observations. For example, if the tester observed trace ?i.?i.!o,

where ?i is an input and !o is an output, the SUT might actually have produced ?i.!o.?i with the communications latency having led to !o being observed after the second input. The ways in which inputs and outputs can be reordered can be represented in terms of a semi-commutation, in which there is an independence relation that says which elements can be swapped in a reordering. For example, if the SUT has input set {?i} and output set {!o} then the independence relation is {(!o,?i)}. This says that an output can be delayed past an input but an input cannot be delayed past an output (since if the SUT observes an input before producing an output then the tester will also observe the input and output in this order). FIFO communications leads to the independence relation in which all outputs can be delayed past inputs and non-FIFO leads to the relation in which an input cannot be delayed past an output but all other swaps are allowed.

The talk then explored classic testing problems, showing that for asynchronous testing they can be formalised in a manner that is almost identical to the usual formalisation for synchronous testing, the difference being the inclusion of the independence relation as a parameter. However, the actual problems do change significantly. For example, the problem of deciding whether one (finite) model conforms to another becomes undecidable for FIFO communications. There are some more positive result concerning test case generation problems (such as deciding whether there is a test case that is guaranteed to force IOTS M into a given state s). For FIFO communications these problems have the same (polynomial time) computational complexity as for the synchronous case (under certain well-defined conditions). In contrast, the test generation problems are EXPTIME-hard for non-FIFO, even for deterministic models. Finally, if communications are FIFO and the tester observes a trace t then it is possible to build a finite automaton M(t) whose language is the set of traces that the SUT might have produced: the possible explanations for the observation of t. The problem of checking whether t is an acceptable observation then becomes one of deciding whether the languages of M and M(t) contain a common word. Since the size of M(t) is quadratic in the size of t, this can be solved in polynomial time. For non-FIFO the problem is NP-complete.

Overall, the results are mixed. One of the benefits of using semi-commutations to model asynchronous communications is that it leads to simple statements of testing problems that are parametrised by the independence relation. While some test automation problems are undecidable, there are also some more positive results for FIFO communications. In contrast, the problems are typically more difficult for non-FIFO communications and there also appear to be fewer results. Finally, it is important to note that the talk was based on results in two papers [1,2] but did not discuss the many excellent papers in this area that have been produced by other authors.

[1] R. M. Hierons: The Complexity of Asynchronous Model Based Testing, *Theoretical Computer Science*, 45 1, pp. 70-82, 2012.

[2] R. M. Hierons: Implementation relations for testing through asynchronous channels, *The Computer Journal*, 56 11, pp. 1305-1319, 2013.

Rob Heirons

CryptoForma: Cryptography and Formal Methods, the Next Generation of Abstractions

Report by: Eerke Boiten

15 January 2014

CryptoForma is a UK EPSRC-funded network of excellence (running until November 2015) for supporting the development of formal notations, methods and techniques for modelling and analysing modern cryptographic protocols. The network brings together cryptographers, developers of security protocols, and the formal methods community which looks to formalise, analyse, and verify such protocols. This work increases security and confidence in such protocols and their applications, to the benefit of protocol designers, businesses, governments, and application users.

On 15 January 2014, CryptoForma held a joint event with BCS-FACS at the BCS London headquarters. During the afternoon, regular CryptoForma attendees and several other BCS-FACS members attended short talks by CryptoForma members on topics such as electronic voting, privacy, and multi-party computation. In the evening, Cédric Fournet of Microsoft Research gave a well attended seminar on a fully verified implementation of TLS, the underlying protocol for most secure web communication.

See http://www.cryptoforma.org.uk/event/cryptoforma-and-bcs-facs-joint-event-15-january-2014/ for more information on the talk including slides.

Regular meetings of the network continue; the most recent (11th) one was at the University of York on 28 May (http://www.cryptoforma.org.uk/event/meeting-york-28-may-2014/), with 9 talks from 8 different UK sites, across a wide range of topics in security and cryptography, and with some 30 attendees.

The next CryptoForma meeting is on 25th September at UCL, University College London. Meetings are open to all those interested, and have regular attendance from industry and nearby European countries as well as from UK academics and PhD

students - with over 100 people having attended one or more meetings so far. Please contact Eerke Boiten (CryptoForma coordinator and liaison on BCS-FACS Board; e.a.boiten@kent.ac.uk) to register for the mailing list to be informed of future events.

FACS Committee

Formal Aspects of Computing Science Specialist Group

Jonathan Bowen
Chairman
ZUG Liaison

Jawed Siddiqi
FACS Treasurer

Paul Boca
FACS Secretary

Roger Carsley
Minutes Secretary

John Cooke
BCS Liaison
Publications

John Fitzgerald
FME Liaison

Margaret West
BCS Women Liaison

Rob Hierons
Chair, Formal Methods and Testing Subgroup

John Derrick
Chair, Refinement Subgroup

Eerke Boiten
CryptoForma Liaison

Tim Denvir
Co-Editor, FACS FACTS

Brian Monahan
Co-Editor, FACS FACTS

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 Jonathan Bowen at the contact points below:

BCS-FACS

c/o Professor Jonathan Bowen (Chair)
Birmingham City University

Email info@bcs-facs.org.uk

Web 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.