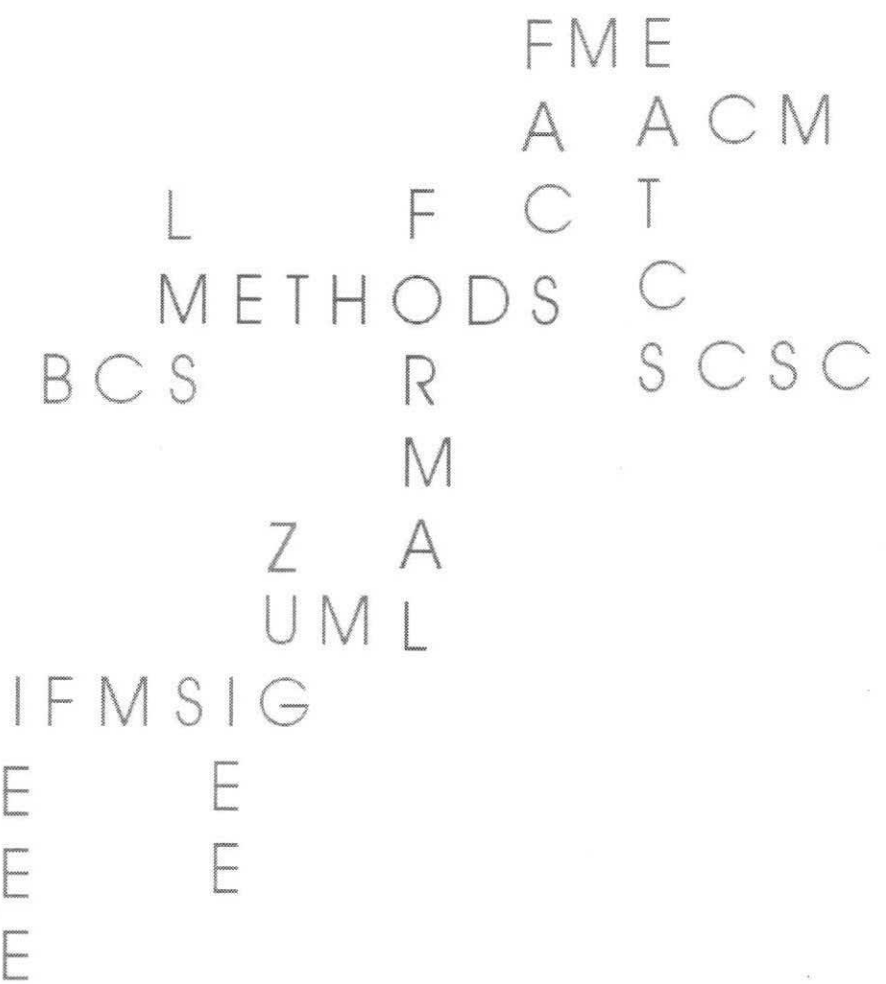
FACS FACTS

Issue 2004-2
July 2004



BCS FACS

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

ISSN 0950-1231

Contents

The *FACS FACTS* Team

Newsletter Editor: Paul Boca

Editorial Team: Jonathan Bowen, Judith Carlton, Kevin Lano, Mike Stannett

RefineNet Reporter: Adrian Hilton

With contributions from: Paul Boca, Eerke Boiten, Raymond Boute, Jonathan Bowen, Roger Carsley, Han-Myung Chang, John Cooke, John Derrick, José Fiadeiro, John Fitzgerald, Adrian Hilton, Stephan Reiff-Marganiec, Carron Shankland, Graeme Smith, Mike Stannett, Mark Utting, Margaret West.

Editorial

Jonathan Bowen, BCS-FACS Chair

Welcome to the new look *FACS FACTS* Newsletter. We have a new and enthusiastic Newsletter Editor, Paul Boca, and we plan to have more regular newsletters for the future. You are very much encouraged to contribute with submissions, both serious and light-hearted. We intend to include conference reports in particular, so if you are attending any relevant conferences, do get in touch with Paul.

This newsletter is timed to be available for the major FACS event of the year, **CSP 25**, a celebration of 25 years of Tony Hoare's *Communicating Sequential Processes* (give or take a year!). The event takes place 7 – 8 July 2004 in the new Keyworth Centre building at London South Bank University and all delegates will receive a copy of this newsletter. The occasion will be reported in the next issue of the newsletter.

Concerning future events, we are always on the lookout for new ideas, but depend on the enthusiasm of members for their organisation. We would like to encourage meetings outside London too. If you have any good ideas and can help with the organisation, please do contact Ali Abdallah, the Events Coordinator, or me to discuss it further.

I look forward to your active involvement with the newsletter. It is an opportunity to report on speculative research, emerging areas, industrial use, etc. Humorous contributions are also very welcome, but the editor's decision is final! If we are overwhelmed with submissions, we can increase the size of the newsletter as necessary. Any reasonable contributions will be accepted, but do discuss your ideas with the editor before submission if you wish to check on their suitability. ■

Other Groups of interest to FACS Members

Z User Group
http://www.zuser.org

London Mathematical Society
http://www.lms.ac.uk

BCS
THE BRITISH COMPUTER SOCIETY

Advanced Programming
Specialist Group
http://www.bcs.org.uk/siggroup/advprog/index.html

Formal Methods Europe
http://www.fmeurope.org

SCSC
Safety Critical Systems Club
http://www.safety-club.org.uk/

The Relaunch of *FACS FACTS*

Paul Boca, Newsletter Editor

At the recent FACS AGM (see report in the next section), I agreed to take over from Kevin Lano as overall Newsletter Editor of *FACS FACTS*. Kevin is still actively involved in the production of the newsletter as an associate editor. Jonathan Bowen, Judith Carlton and Mike Stannett have agreed to act as associate editors too. Jonathan has contributed to both the content and look of the previous two issues, and so it is good to have him on the team officially. With the help of this new team, I aim to increase the frequency of publication of *FACS FACTS* to four issues per year (starting in 2005). This year there will be three issues.

I aim to improve the content too. The ideas I have been exploring to achieve this include introducing regular reporters and commissioning guest columns. I am very pleased to announce that Adrian Hilton (Praxis Critical Systems) has agreed to be the FACS reporter for RefineNet meetings. Adrian's first report can be found on Page 29 of the newsletter, and the next one will appear in Issue 2004-3. There are lots of interesting formal methods projects and initiatives underway, and I would like to include regular reports on these in future issues of *FACS FACTS*. If you are interested in reporting on any of these, please contact me on Paul.Boca@virgin.net .

Mike Stannett is our first guest columnist and has written a clear exposition of X-machines. I hope you find this of interest and it inspires you to look deeper into the subject. If you are interested in being a guest columnist, please contact me at the above address.

I have tried to improve the look of the newsletter as well. Articles are laid out in an informal style, and I have supplemented some of them with "online resources" boxes to give readers easy access to information. Work will continue on improving the newsletter, but feedback is required to let us know whether or not we are producing the kind of newsletter you want to read. Please let me know what you think of this issue: all criticism gratefully received!

Work has already started on the next issue of *FACS FACTS*, which will be out in November 2004. A call for submissions can be found on the next page, which in due course will be sent out to mailing lists and newsgroups to help attract contributions from further afield. I have already received offers of submissions, so the next issue should be as informative as this one. Articles and news items are still required, though, so please do consider submitting something, no matter how small. The success of *FACS FACTS* does lie very much in the hands – more precisely, at the fingertips – of people in the formal methods community. So get typing, and emailing! I look forward to hearing from you. ■

FACS FACTS Issue 2004-3

Call For Submissions

Deadline **15 October 2004**

We would like to receive 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
- 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 organisations
- 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 plain text or Microsoft Word format to Paul Boca [Paul.Boca@virgin.net], the Newsletter Editor, by **15 October**.

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

News from FACS: Annual General Meeting

Roger Carsley, FACS Secretary & Paul Boca, Membership Secretary

The 2004 FACS AGM was held on Friday 23 April in the boardroom of the Technopark, London South Bank University. The meeting consisted of two parts: a discussion of FACS business followed by a talk by Margaret West, University of Huddersfield, on animating Z specifications.

FACS Business

All of the Committee members except Kevin Lano were able to attend. There was a great deal to discuss, as the Committee had not had a face-to-face meeting since the previous AGM. Regular email discussions on the FACS-COMMITTEE JISC mailing list had taken place in the interim, however. Detailed minutes of this meeting will appear on the FACS website in due course. In the meantime, a brief summary appears here:

- Jonathan Bowen, FACS Chair, summarised the year's activities as including an article about BCS-FACS in the *Computer Bulletin*, a workshop at Oxford Brookes University on Teaching Formal Methods, sponsorship of prizes at meetings organised by other groups and an article by Richard Sharpe on Formal Methods in *Computing*. Details of the TFM workshop can be found in Issue 2004-1 of *FACS FACTS*.

- Paul Boca, Membership Secretary, reported that FACS has 47 fully paid-up members, 19 Formal Aspects of Computing (FAC) journal subscriptions and 151 on the JISC mailing list.

- PayPal payment had been arranged by Paul Boca and has proved popular, particularly with overseas members. Online membership is being considered.

- John Fitzgerald, SCSC and FME liaison officer for FACS, reported briefly on recent Formal Methods Europe activities: the upcoming conference FM 2005 and the revamped FME website. A full report on FME appears on Page 24 of the newsletter.

- As Kevin Lano, Newsletter Editor, could not attend, Jonathan Bowen reported back on the newsletter. The committee members were in agreement that the frequency of publication of *FACS FACTS* should be increased to four issues per year. To help achieve this, Paul Boca agreed to take over as Newsletter Editor, managing the overall production and commissioning of articles, assisted by Associate

Editors Jonathan Bowen, Judith Carlton, Kevin Lano and Mike Stannett.

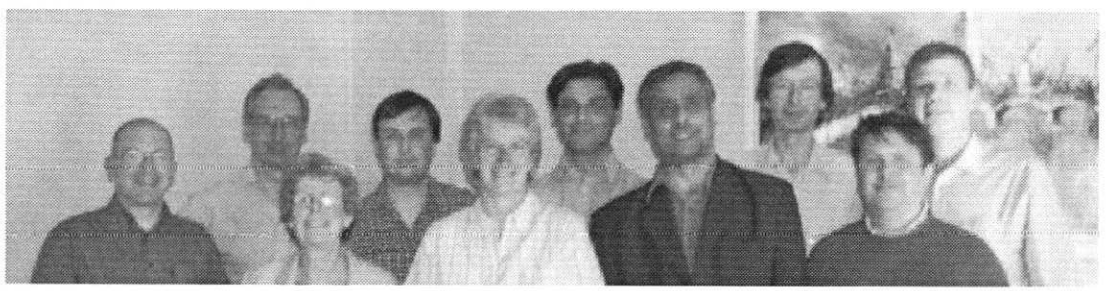
- John Cooke, Journal Liaison, told us about a healthy situation with the FAC Journal having enough papers to fill Volume 16. Further information on the FAC journal can be found on Page 30 of this issue.

- Mike Stannett, FACS Webmaster, reported that the website was receiving 4000 hits per month. The website is about to be redesigned and a number of new features are planned. Further details can be found in a separate report on Page 40 of this issue.

- Ali Abdallah, Events Organiser, reported on preparations for the *25 years of CSP* event, which will take place 7–8 July at London South Bank University. The website for the event is:

  http://www.lsbu.ac.uk/menass/csp25.

- Jawed Siddiqi, Treasurer, reported a healthy state of our accounts, including a £500 surplus from the Formal Aspects of Security event in 2002. This profit will be used to help fund future activities, such as prize giving, providing bursaries and organising events.



FACS Committee present at the AGM:
(Back Row) John Cooke, Roger Carsley, Ali Abdallah, Jonathan Bowen, Paul Boca,
(Front Row) John Fitzgerald, Margaret West, Judith Carlton, Jawed Siddiqi, Mike Stannett

The Committee was reappointed, as, regrettably, there are few members putting themselves forward for positions. If you can spare some time to help FACS, please contact Jonathan Bowen on jonathan.bowen@lsbu.ac.uk, especially if you have a particular role or expertise that you can provide.

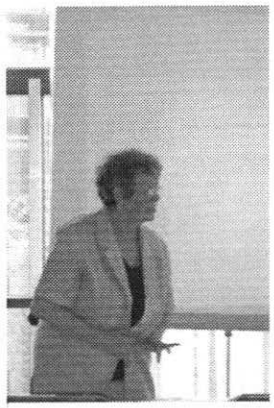
There was a small reorganisation of the Committee. In addition to the changes to the editorial board of *FACS FACTS*, Kevin Lano is now UML Liaison Officer for FACS and Mike Stannett takes on the additional role of LMS-CS Committee Liaison Officer. A report by Mike Stannett on LMS-CS liaison can be found on Page 28. Full details of the FACS Committee can be found, as usual, at the end of this Newsletter.

The Committee will be meeting again either towards the end of this year or early next year to discuss the future of FACS. The aim is to develop a five-year plan and to evaluate the current mission statement. The outcome of this

day-long meeting, which is being organised by Jawed Siddiqi and Paul Boca, will be reported in a future issue of *FACS FACTS*.

Talk by Margaret West

Animating a Z Specification in a Logic Programming Language: Method and Examples



Margaret West

ABSTRACT

This is the first of two seminars describing work covered in my PhD. This talk is concerned with the practical aspects of animating Z specifications using the Goedel logic programming language. The second talk should be given after Easter and will be concerned with theoretical aspects, i.e. the correctness of the approach.

Z (pronounced 'zed') is a formal specification notation based on set theory and first order predicate logic. It is used by industry as part of the software (and hardware) development process in Europe, USA and elsewhere. It has recently undergone international ISO standardization (ISO/IEC 13568:2002). The use of mathematics in modelling means that formal reasoning can then be applied to check the consistency of the specification. Further, a version of the specification can be executed or 'animated' in order to demonstrate its functionality and detect flaws.

The thesis was concerned with the availability of tools for supporting Z and in particular animation. The language used for animation is frequently a logic programming language, and in the first part of this talk the Z notation and the Goedel logic programming language are both briefly described. A simple example is then used to illustrate the translation rules. The second part of the talk is concerned with two case studies: Z specifications of the Unix file system and a simple assembler are translated to Goedel and executed and the practical advantages and disadvantages of the method are demonstrated.

The slides that accompanied the talk can be downloaded from:

http://scom.hud.ac.uk/scommmw/recent_talk/slides4.pdf

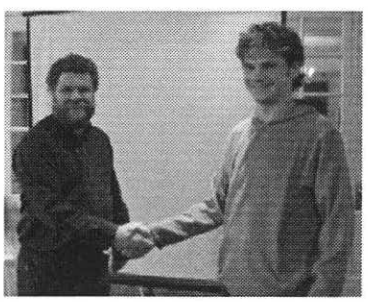
There will be a follow-up talk on **14 July** at the University of Huddersfield. For more information, contact Margaret West on M.M.West@hud.ac.uk. ■

Conference Reports

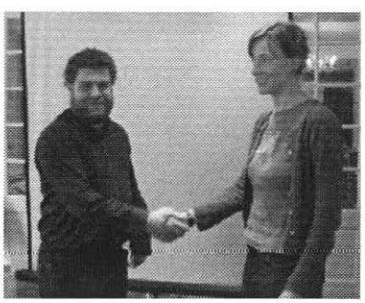
BCTCS 2004: 20th British Colloquium for Theoretical Computer Science, Scotland, 5 – 8 April 2004

Stephan Reiff-Marganiec and Carron Shankland

BCTCS 2004 was held 5 – 8 April 2004 in Pitlochry (Scotland). The event was hosted by the Department of Computing Science and Mathematics, University of Stirling and was supported by LMS, EPSRC, BCS-FACS and Microsoft Research. The colloquium featured an interesting and wide ranging program of invited and contributed talks, as well as some tutorials. Invited talks were presented by Prof Luca Cardelli (Microsoft Research, UK), Dr Sharon Curtis (Oxford Brookes University), Prof José Fiadeiro (University of Leicester), Dr Rob Irving (University of Glasgow) and Prof Kenneth Turner (University of Stirling). Invited tutorials were given by Dr Rachel Norman (University of Stirling) and Prof Rick Thomas (University of Leicester). The invited programme was complemented by 19 contributed talks.



Two prizes, sponsored by BCS-FACS, were given for outstanding talks. These were awarded to Alastair Donaldson (right), University of Glasgow, and Corrina Elsenbroich (below), King's College London. Both prizes were presented by Chris Tofts (HP Labs), the outgoing BCTCS President. The main point of events organised by BCTCS is to provide an opportunity for networking in the UK theoretical computer science community. As usual, participants took great advantage of this. A high point of the colloquium was the formal dinner at which Chris Tofts mused on the past and future of BCTCS. He was presented with a small token of our esteem by the incoming president (Faron Moller, University of Wales), which was purchased earlier in the day when we all visited Edradour distillery for a whisky tasting.



The 21st British Colloquium for Theoretical Computer Science (BCTCS 2005) will be hosted by the University of Nottingham. Anyone wishing to contribute talks concerning aspects of Theoretical Computer Science is warmly invited to do so. Details are available at http://www.cs.nott.ac.uk/~gmh/bctcs2005.html .

Invited Talks

Membrane Interactions
Luca Cardelli, Microsoft Research, Cambridge

Computer simulation of biological systems is computationally intensive, to say the least. Fortunately, living cells are not just random collections of billions of molecules: they are extremely well organised (although stochastic) systems. If we can model their organisation, we should be able to simulate them abstractly and effectively. A large part of the organisation of (eukaryotic) cells depends on hierarchies of nested membranes, their properties, the proteins bound to membrane surfaces, and the way membranes interact dynamically with each other. I discuss some preliminary efforts in modelling dynamic membrane systems. The long-term goal is to represent the structure and function of biological systems via formal languages, for description, simulation, analysis and (eventually) compilation.

Functional Fractal Image Compression
Sharon Curtis, Oxford Brookes University

Fractal image compression is an ingenious technique that compresses an image by representing it as a partitioned iterated function system. As functions are fundamentally involved in this modelling of images, it is a natural step to express the image compression and decompression algorithms in the world of functional programming. This talk will give an introduction to fractal image compression and an account of recent experiences implementing the compression and decompression in Haskell.

Software Architectures in 3D
José Fiadeiro, University of Leicester

The promise of Software Architectures for controlling the complexity of system construction and evolution is based on a conceptually simple separation of concerns: that between the computational aspects associated with the functionality of basic services and the mechanisms through which interactions are coordinated to make required global properties emerge. This approach relies on the fact that individual components can perform the computations that are required to ensure the functionalities specified for their services at the locations in which they are placed, and that the coordination mechanisms put in place through connectors can be made effective across the "wires" that link components in the underlying communication network. Clearly, this model is too "static" for the new generation of systems that will have to operate in dynamic configuration topologies. Whereas the mobility of computations is a problem that we are starting to know how to address, the effects of mobility on coordination are only now being recognised as an additional factor of complexity, one for which current architectural concepts and techniques are not prepared. This lecture will address the challenge of making Software Architectures three-dimensional by proposing a mathematical model in which

Distribution is recognised as a separate concern from Computation and Coordination.

Fifty Years of Stable Marriage
Rob Irving, University of Glasgow

It is over fifty years since the classical stable matching algorithm was first used in a large-scale practical application. During that period, the stable marriage problem and its many variants have been the focus of extensive interest and research on the part of computer scientists, mathematicians, game theorists, economists, and others. This talk presents an overview of stable matching problems, primarily from the standpoint of the algorithmicist. We emphasise a range of recent developments, and explore their practical implications as well as their theoretical significance.

Test Generation for Radiotherapy Accelerators
Ken Turner, University of Stirling

System specification with LOTOS (Language Of Temporal Ordering Specification) will be briefly described. To make test generation practicable, specifications are annotated with event constraints using PCL (Parameter Constraint Language) as a means of stating test purposes. Automated test generation follows the principle of input-output conformance to check whether an implementation agrees with its specification. Test suites are created from a transition tour of the automaton that is generated from the specification. The approach has been applied to a safety-critical systems case study: radiotherapy accelerators as used in cancer treatment. The automatically generated test suite can be executed manually or automatically. The goal is to discover situations in which an accelerator does not behave in conformity with its specification.

Invited Tutorials

Modelling of Biological Systems
Rachel Norman, University of Stirling

The aim of this tutorial will be to convince you that modelling biological systems is important and exciting. Mathematical models have been used successfully to model many different types of biological problems. However, as the models become more biologically realistic and therefore more mathematically complex, theoretical biologists are turning to theoretical computing science techniques to see what they can bring to the party. The main emphasis of the talk will be to give you some background and examples of the types of problem addressed and types of mathematical model used. I will then go on to discuss my ideas about of the types of TCS techniques that could be used and how the two could be linked together. Biological emphasis will be on infectious disease systems but the techniques could be applied to any area of biology.

Formal language theory and word problems of groups
Rick Thomas, University of Leicester

The purpose of this talk is to survey some interesting connections between formal language theory on the one hand and group theory on the other. Groups arise in many ways and are often described by means of "presentations". A presentation for a group G consists of a set of generators for G and a set of relations between words in the generators that is sufficient to define the group operation. If we have a presentation for G with a finite set of generators and a finite set of relations, then we say that G is "finitely presented". One of the classical results in group theory is the unsolvability of the word problem for finitely presented groups; this says that there are finite presentations such that there is no algorithm to decide whether or not a word in the generators represents the identity element of the group defined by that presentation. An alternative way of describing this situation is as follows: there are finitely presented groups G such that, if we consider the set W of all words representing the identity element of G, then there is no algorithm for determining membership of W. There is also an elegant result of Boone and Higman describing which finitely generated groups have a solvable word problem. One natural question that arises from this is the following: if we take some restricted model of computation, which groups have a word problem and which is decidable within that model? The restriction might be, for example, that the word problem is accepted by a particular type of automaton or generated by a particular type of grammar. The purpose of this talk is to survey some of what is known in this field. We will not assume any prior knowledge of group theory and we will review what we need from formal language theory.

Contributed Talks

Investigating Structural Symmetry in Models of Concurrent Systems
Alastair Donaldson, University of Glasgow

Over the last decade, symmetry reduction techniques have been shown to be successful in combating the state space explosion problem in model checking. Such reduction techniques usually assume that symmetries of a concurrent system are known in advance, and these symmetries are exploited when model checking the original system description. Two obvious questions are a) "can we automatically detect symmetries of a concurrent system from its textual description?" and b) "would it be possible to apply symmetry reduction directly to the textual description of a system so that a reduced textual description could be model checked in the usual way?" In my talk I will describe some preliminary work in attempting to answer these questions.

Goal Directed Dynamic Abduction

Corinna Elsenbroich, King's College London

Building on an idea by Diderik Batens et al. [Dynamic Dialectical Logics, 1989] we define a goal directed dynamic logic that facilitates reasoning with inconsistencies while giving a larger set of conclusions than a normal paraconsistent logic. The logic will be essentially dynamical as conclusions drawn at certain stages of the proof are conditional and can be withdrawn later on if and when the conditions are not met any longer. This dynamic logic is essentially what is needed for abductive reasoning so that we can define an abduction rule integrated into the above framework.

Dual-context multicategories

Fedor Fomenko, Edinburgh University

Bellantoni and Cook gave a recursion-theoretic description of polynomial time computations that is independent of any externally imposed resource bounds – system B. Their idea was to separate inputs of all functions in B into "normal" inputs or "safe" inputs. Functions could apply any polytime operation to their normal inputs, but on safe inputs no operations that increase the length by more than an additive constant were allowed. Their safe recursion scheme allowed definition of the new functions by recursion on normal inputs and recursively defined values were to be processed only through the safe parameters. Barber proposed DILL – a new sequent style formulation of Intuitionistic Linear Logic which had clear introduction-elimination rules for all connectives including ! (which previously had four rules associated with it). His idea was to split contexts into two parts – one for intuitionistic variables and another one for linear variables. Weakening and contraction was allowed only for intuitionistic variables. We propose to extend the definition of the multicategory [Lambek] to the dual-context multicategory with contexts separated into two parts. We define linear-linear (LL), intuitionistic-linear (IL) and intuitionistic-intuitionistic multicategories (II). IL multicategories can be used as models for DILL. Then we consider the notion of the strong endofunctor of dual-context multicategories, which is an extension of the strong endofunctor on usual categories. We define initial algebras for such endofunctors, corresponding to the Bellantoni-Cook safe recursion scheme and show how system B can be modelled in II dual-context multicategories.

The irreducible word problem of groups

Ana Fonseca. University of Leicester

Given a group G with a finite generating set X, the word problem of G with respect to X is the set of words over the elements of X and their inverses, which are equal to the identity in G. The irreducible word problem of G is then defined to be the subset of the word problem consisting of those nonempty words that have no nonempty proper subword which is equal to the identity. There are many interesting links between word problems and irreducible word problems on the one hand and formal language theory on the other. In

particular, groups whose irreducible word problem is either a finite or regular language have been classified. I will present some results concerning groups with a contextfree irreducible word problem.

Modal Logics of Submaximal Spaces
David Gabelaia, King's College London

The operations of closure and derivation on a topological space can be understood as normal modal operators working on the Boolean algebra of all subsets of the underlying set. This gives rise to modal logics associated with topological spaces. It turns out that the formalism with the closure operator is rather weak in feeling the topological structure, whereas the derivation operator offers more expressivity, and allows for differentiating natural topological classes by logical means. In this talk we will concentrate on the subclasses of submaximal spaces (among others door spaces and maximal spaces), which have been the subject of increasing interest in the topological community for past decades. Such spaces are of interest in digital topology as well (e.g., Khalimsky line is submaximal). It will be demonstrated that the equations involving the closure operator alone fail to distinguish these classes. On the other hand, equations built using the derivation operator capture precisely the individual characteristics of each of them. Finally, for each of the topological classes concerned, a simple class of finite Kripke frames generating the same modal validities will be exhibited.

Representing XML Documents Compactly
Richard Geary, University of Leicester

We consider succinct or space-efficient representations of trees that effectively support a variety of navigation operations. Our representation takes 2n + o(n) bits to represent a tree of n nodes, which is within o(n) bits of the information-theoretic minimum and supports all operations in O(1) time on the RAM model. In addition to the existing motivations for studying such data structures, we are motivated by the problem of representing XML documents compactly so that XPath queries can be supported efficiently.

Categories of Containers
Neil Ghani, University of Leicester

The efficient representation and manipulation of data is one of the fundamental tasks in the construction of large software systems. One of the most successful approaches to date has been Hindley-Milner polymorphism, which provides predefined mechanisms for manipulating data structures providing they are parametric in the data. I will talk about recent work on containers that has led to some intriguing new insights into polymorphism. In particular I will classify all the polymorphic programs between containers and show they have a remarkably simple form. I will then extend this to cover quotient types where a similar beautiful picture emerges.

On the Issues of the Application of Algorithmic Skeletons to Metacomputing

Horacio Gonzalez-Velez, University of Edinburgh

Metacomputing is generally referred to as the execution of parallel codes across multiple supercomputer or clusters located at different locations. In a broader sense, Grid computing is then defined as encompassing not only metacomputing in the form of computational grids, but also scheduling, enhanced security, data access and migration, peer-to-peer processing, and other technologies. With the advent of computational grids as part of the world initiative on the Grid, the use of proven programming paradigms has become an issue. Current trends in scientific application programming for computational grids include the development of frameworks for distributed programming, Internet-based processing or component-based technology, as well as the use of grid-based portals. Having been conceived as higher order functions corresponding to good parallel algorithmic techniques, the algorithmic skeletons have proved to be a reliable method for the design of high-level, structured parallel programming solutions. Our approach is different since we intend not to adapt mainstream, distributed Grid development technology into the main programming arena, but conversely we intend to apply parallel programming techniques to solve major scientific or industrial applicative problems in the Grid. Parallel algorithmic solutions often show certain patterns, such as pipelines, process farms, or branch-and-bound, that are naturally addressed with the use of skeletons. Hence, it may be of great benefit as a novel way of programming a grid-oriented computational environment.

Using Unique Decomposition Bases to Efficiently Compute Bisimilarity on Normed Basic Parallel Processes

Will Harwood, University of Wales

Jancar and Kot show (AVIS 2004) that the PSPACE BPP-bisimilarity algorithm of Jancar (LICS 2003), which characterises states modulo bisimilarity using sets of NORMS, can be made to run in $O(n^3)$ time on the subclass *normed* Basic Parallel Processes. This is considered to be a significant improvement over the previous polynomial-time algorithm for bisimilarity on normed BPP, by Moller, Hirshfeld and Jerrum (MSCS 1996), which relied on *unique decomposition bases* and for which no explicit upper bound was given. However, while the latter's related algorithm for normed Basic Process Algebra (TCS 1996) performs (naively) in $O(n^{19})$ time, we show here that the decomposition base method for normed Basic Parallel Processes is not so pessimistic.

Combining Spatial and Temporal Logics: Expressiveness vs. Complexity

Roman Kontchakov, King's College London

Modal logic S4u is complete with respect to the topological semantics where the diamond modality is interpreted as the closure operator of topological spaces. On the other hand, RCC-8, a fragment of the region connection calculus RCC with eight jointly exhaustive and pairwise disjoint predicates, is

known to be embeddable into S4u. In this talk we first consider several intermediate fragments of S4u extending RCC-8 whose computational complexity varies from NP to PSPACE. Then we construct and investigate a hierarchy of spatio-temporal formalisms that results from various combinations of the propositional temporal logic PTL with these fragments of S4u. We demonstrate how different "blending" principles as well as spatial and temporal components give rise to NP-, PSPACE-, EXPSPACE-, 2EXPSPACE-complete and even undecidable spatio-temporal logics.

Approximability Results for Induced Matchings in Graphs

David Manlove, University of Glasgow

An induced matching in a graph G=(V,E) is a set of edges M, no pair of which are adjacent or joined by an edge in G. Let MIM be the problem of finding a maximum induced matching in a graph G. MIM has applications in channel assignment, VLSI design and network flow problems. However MIM is known to be NP-hard, even if G is d-regular (i.e. each vertex has degree exactly d). In this talk we consider the approximability of MIM in d-regular graphs. We show that, for each $d \geq 3$, MIM admits an approximation algorithm with performance guarantee d-1. On the other hand, we show that, for the same class of graphs, MIM does not admit a polynomial-time approximation scheme unless P=NP. This is joint work with Billy Duckworth and Michele Zito.

Categorical Models of First Order Classical Sequent Proofs

Richard McKinley, University of Bath

It has been thought for a long time that there were no non-trivial models of classical logic that admit full cut elimination – Fuhrmann and Pym have recently shown models that escape the usual collapse to a Boolean algebra, by enriching the proof spaces with a partial order structure – this partial order interprets the structure of cut-elimination. I will briefly recap the notions of sequent calculus and cut-elimination, and the non-collapsing models, and then present the generalisation of this result to the first order case, using an indexed category approach; we exhibit a sound, complete class of models, with examples, and consider what new light the structure of these models might shed on Herbrand's Theorem.

Semantics of a Sequential Language for Exact Real-Number Computation

Jose Raymundo Marcial-Romero, Birmingham University

We present a programming language with a built-in ground type for real numbers. In order for the language to be sufficiently expressive but still sequential, we consider a construction proposed by Boehm and Cartwright. The non-deterministic nature of the construction suggests the use of powerdomains in order to obtain a denotational semantics for the language. We show that the construction cannot be modelled by the Plotkin or Smyth

powerdomains, but that the Hoare powerdomain gives a computationally adequate semantics. As is well known, Hoare semantics can be used in order to establish partial correctness only. Since computations on the reals are infinite, one cannot decompose total correctness into the conjunction of partial correctness and termination as it is traditionally done. We instead introduce a suitable operational notion of strong convergence and show that total correctness can be proved by establishing partial correctness (using denotational methods) and strong convergence (using operational methods). We illustrate the technique with a representative example.

Automatic Presentations and Groups

Graham Oliver, University of Leicester

A structure is said to be computable if there is a coding for its domain and relations that can be read by Turing machines. Restricting the Turing machines in the definition to finite automata then gives us structures that are particularly simple computationally. These structures are said to have an automatic presentation. This talk will briefly review the general area, before presenting a classification in a specific case: a finitely generated group has an automatic presentation if and only if it has an abelian subgroup of finite index.

From Discrete Games to Counter Automata and Matrix Reachability Problem

Igor Potapov, The University of Liverpool

In the last few years, discrete games have attracted more and more attention in fundamental computations theory since the dynamics complexity is directly connected to computational power of the game. In models of discrete adaptive games, a finite number of players or agents pick from a finite set of actions, at discrete times, and receive rewards that depend on their own and the other agents' actions. Each agent chooses according to a rule called its strategy; if agents have more than one strategy available, they use the second layer strategy to choose among them. For computation power analysis we suggest representing the whole process as a counter automaton with guards. In that case, many questions about the predictability of such games reduce to the reachability problems in automata-like systems. In this presentation we show new decidability and undecidability results for counter automata with guards which are connected to standard computational models, the reachability problems for piecewise and non-deterministic iterative maps, matrix semigroups etc. This is joint work with Alexei Lisitsa.

Graph Problems in a Data Streaming context

Paul Sant, King's College London

The Data Streaming model is a new model in which the input is presented as a stream of data, for example, a stream of edges of a graph. This model has

been studied, by among others, S. Muthukrishnan and G. Cormode. An added constraint of the model is that only polylogarithmic (in the size of the input) space is available for storage. Much of the recent work in the data streaming model has concentrated on finding most frequent items in a data set, approximating the L_p norm, etc. However, graph problems have not been extensively studied in this model and only recently have any results been published on what is possible (from a graph theoretic perspective) within this model. In this talk I will describe some of the graph problems and results that are known and also explain the main problems that are encountered when looking at graph problems in the data-streaming context.

An Algebra of Security Policy

Francois Siewe, De Montfort University

Despite a considerable amount of existing work on access control policies, enforcing multiple policies is still a challenge in order to achieve the level of security required in many real-world systems. Moreover current approaches address security settings independently, and their incorporation into a system development lifecycle is not well understood. This paper presents an algebra for the specification of access control policies. The approach can handle the enforcement of multiple policies through policies composition. Interval Temporal Logic (ITL) is our underlying formal framework and policies are modelled as safety properties expressing how authorisations are granted over time. The properties express how authorisations are granted over time. The approach is compositional, and can be used to specify other system properties such as functional and temporal requirements. The use of a common formalism eases the integration of security requirements into system requirements so that they can be reasoned about uniformly throughout the development lifecycle. Furthermore policy specifications are executable in Tempura, a simulation tool for ITL. This provides a way for validating a formal specification against the (informal) security requirement early before proceeding to the design and implementation.

Tree Matching: Metrics and Algorithms

Gabriel Valiente, Technical University of Catalonia

Tree matching is a natural generalisation of string matching to trees, with practical applications in combinatorial pattern matching, pattern recognition, chemical structure search, computational molecular biology, and other areas of engineering and life sciences. In this talk, a gentle introduction to tree matching will be given, recent algorithmic results will be reviewed, and similarity measures based on tree matching

Online Resources

BCTCS website:
http://www.csc.liv.ac.uk/~ped/bctcs/summary.html

BCTCS 2004 Website:
http://www.cs.stir.ac.uk/events/bctcs/

BCTCS 2005 Website:
www.cs.nott.ac.uk/~gmh/**bctcs**05.html

will be discussed.

A Logic for Describing Operating Systems in Terms to their Resources and Processes

Daniel Wiley, University of Bath

Instances of the Logic of Bunched Implications have been successfully used to describe resource semantics. Judgements of the form 'a resource model satisfies a bunched logical assertion' have been used effectively to describe systems such as heap storage. We hope to enrich this judgment to make assertions about resources relative to processes acting on these resources satisfying bunched assertions. We demonstrate that, through this enriched judgement, we are able to describe complex dynamic systems, an operating system and, in doing so, we are able to gain a significant insight into the properties and resource nature of such systems. ■

IFM'04: Fourth International Conference on Integrated Formal Methods, Canterbury, Kent, 4 – 7 April 2004

Eerke Boiten, John Derrick, Graeme Smith

IFM 2004

The meeting of IFM'04 at the University of Kent was considered a great success, and in total nearly 100 delegates attended it and the associated events.

Activities started on Sunday 4 April with a tutorial on the Unifying Theories of Programming given by Jim Woodcock and Ana Cavalcanti. Attendance exceeded expectations, with around 50 delegates listening to a well-received exposition of the subject.

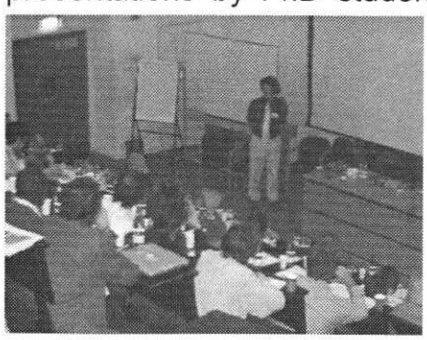
Online Resources

IFM2004 Website:
http://www.cs.kent.ac.uk/events/conf/2004/ifm/

FORTEST Website:
http://www.fortest.org.uk

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

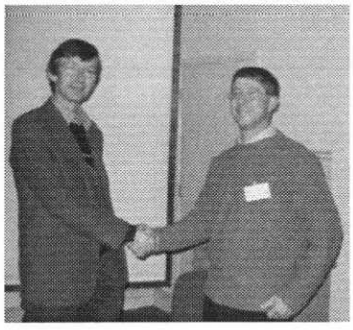
The FME AGM and a Doctoral Symposium also took place on the Sunday. John Fitzgerald hosted the FME AGM, which was attended by about 15 members. The Doctoral Symposium was sponsored by the EPSRC, and around 30 delegates listened to short presentations by PhD students of their work. The sponsorship allowed an additional 15 PhD students to attend the conference, and was considered an excellent use of the resource.



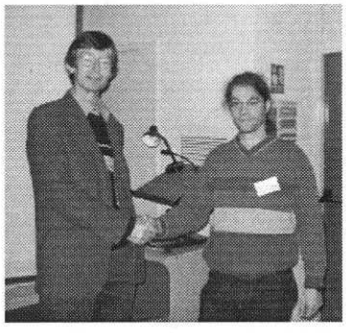
The main event began on Monday, with the FME sponsored talk by Tom Ball (left) of Microsoft. Additional invited talks were given by Ursula Martin and Tom Melham. All talks

were excellent, and additionally, the invited speakers took time to attend all of the conference and engage with the community. For this we are grateful.

The technical program was spread over two and a half days, with a high standard of talks and attendance. The programme was also varied without being too diverse, and fell firmly within the IFM remit.



BCS-FACS sponsored two prizes, for which the organisers were grateful. Both were keenly contested. The best paper award was won by Steve Schneider (left) and Helen Treharne, while the best student presentation was won by Steffen Schlager (right) of the University of Karlsruhe. Both prizes consist of a year's membership to FACS and a year's subscription to the Formal Aspects of Computing journal. They were presented by Professor Jonathan Bowen, the BCS-FACS Chair.



A meeting of the EPSRC FORTEST network followed the main IFM events, which as usual consisted of a number of informal presentations. This dovetailed with the final IFM session, which had a testing theme.

A number of social activities were enjoyed. The highlight of these was the conference dinner at nearby Leeds Castle (left). We were given an individual tour round the castle followed by a reception and dinner in the Terrace and Fairfax rooms, which overlooked the floodlight castle and moat. Jim Davies provided the after dinner entertainment.

The next IFM is scheduled for the autumn of 2005; details and location will be announced soon. ■

4TH INTERNATIONAL CONFERENCE OF B AND Z USERS

University of Surrey, Guildford, UK, 13-15 April 2005
Organised by APCB and the Z User Group

http://www.zb2005.org

Important dates:

Submission of draft paper: **1 November 2004**
Submission of tutorial proposals: **22 November 2004**

ICSE 2004: 26th International Conference on Software Engineering, Edinburgh, 23 – 28 May 2004

Han-Myung Chang

Background

ICSE (http://conferences.iee.org/icse2004) is one of the largest events on software engineering. This year the main conference, which was held in Edinburgh, consisted of the following:

- 3 keynote presentations
- 3 panel sessions
- 8 formal research demonstrations
- a track of presentations on Linkages to Other Disciplines
- 58 technical papers

In addition to the main part, there were 19 workshops, 15 tutorials, a doctoral symposium, and 5 co-located and adjunct events.

The keynote speakers and the titles of their talks were:

- Richard Stallman, "Against Software Patents"
- Karl Lieberherr, "Controlling the Complexity of Software Designs"
- Janet Thornton, "Software Engineering Challenges in Bioinformatics"

The proceedings are published by IEEE: Proceedings of 26th International Conference on Software Engineering (ICSE 2004), IEEE, May 2004.

A Selection of Papers on Formal Methods

The program committee reviewed 436 technical papers and 59 were accepted, but one of the accepted papers was subsequently rejected. So the acceptance rate was around 13%. From these, I have selected seven papers that are relevant to formal methods and which particularly interested me. These are summarised briefly below:

(1) Visual Timed Event Scenarios (pp. 168–177), A. Alfonso, V. Braberman, N. Kicillof, and A. Olivero

This paper proposes VTS, which is a scenario-based notation to express real-time requirements in a visual way. Visually specified scenarios are translated into observer timed automata in order to model check system models. A tool has been developed to support editing scenarios, automatic translation from visual forms into timed automata, and model checking.

(2) Verifying DAML+OIL and Beyond in Z/EVES (pp. 201–210), J. Dong, C. Lee, Y. Li, and H. Wang

This paper presents verification methods for the Semantic Web using the Z/EVES proof tool. The Z semantics for the ontology language DAML+OIL are shown. Ontologies are transformed into Z specifications and then Z/EVES can be used to checked properties of the ontologies. By applying these methods to a real application, undiscovered errors in the original ontologies were found.

(3) Assume-Guarantee Verification of Source Code with Design-Level, Assumptions (pp. 211–220), D. Giannakopoulou, C. Păsăreanu, and J. Cobleigh

This paper demonstrates a methodology to integrate model checking at different phases of system development. The approach is modular analysis at the design level to improve the performance of verification at the code level. One of the key techniques is to automatically decompose the verification task by generating component assumptions for the properties to hold. A case study to apply the methodology to a NASA system is shown.

(4) Compositional Verification of Middleware-Based Software Architecture, Descriptions (pp. 221–230), M. Caporuscio, P. Inverardi, and P. Pelliccione

This paper presents a methodology to model check middleware-based software architecture specifications. The approach is to exploit the structure imposed on the system by the software architecture in order to develop an assume-guarantee methodology. A subset of LTL (Linear Temporal Logic) formulae, which can be automatically generated by MSC (Message Sequence Charts), is proposed and then a theorem on Architectural Decomposability is shown.

(5) Revisiting Statechart Synthesis with an Algebraic Approach (pp. 242–251), . Ziadi, L. Hélouët, and J.-M. Jézéquel

This paper proposes an algebraic framework for synthesizing statecharts from UML 2.0 sequence diagrams. Algebraic compositions of sequence diagrams and a framework for statecharts compositions are presented. Then an algorithm generating flat statecharts from sequence diagrams is shown. The modification or replacement of a given scenario has a limited impact on the synthesis process.

(6) Feature-Based Decomposition of Inductive Proofs Applied to Real-Time Avionics Software: An Experience Report (pp. 304–313), V. Ha, M. Rangarajan, D. Cofer, H. Rueβ, and B. Dutertre

This paper reports the use of the PVS theorem prover to analyse a real-time

operating system for embedded avionics. This technique demonstrates an incremental approach to theorem proving that is indeed scalable. When two new features are incrementally added to a base model, most of the proofs were reused without modifications.

(7) A Tool for Writing and Debugging Algebraic Specifications (pp. 449–458), J. Henkel and A. Diwan

This paper proposes a tool that helps programmers write and debug algebraic specifications. The approach is to provide a seamless integration between an interpreter for algebraic specifications and Java applications. The tool creates a prototype implementation of a class from its algebraic specification; the execution of the specification is then compared to a hand-coded implementation.

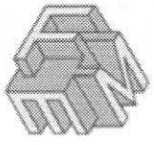
Final Remarks

ICSE is a large event on software engineering, covering a wide range of areas. Although ICSE 2004 had no technical sessions specifically on formal methods, there were many papers, other than the seven I have selected above, focussing on formal approaches in areas such as verification, testing, requirements, UML and Object-Oriented Programming. I felt that formal methods were one of the fundamental key technologies in software engineering. ■

News from FME

John Fitzgerald, FME Chair

Background on FME



http://www.fmeurope.org

I am often asked what FME actually is. The three-letter abbreviation (TLA!) is commonly associated with a successful series of formal methods conferences but now people are seeing "FME lectures" at other symposia, FME event sponsorship and may have heard about new events like the International Symposium on the teaching of formal methods. So what *is* FME?

FME stands for Formal Methods Europe. It started life as the European Commission's advisory panel on formal methods but flew the nest in 1998 and is now incorporated as an Association in Netherlands Law. Unlike FACS, it has no parent organisation. Our aim is to stimulate research and industrial application in formal methods by encouraging research scientists and practitioners to cooperate closely in developing and applying formal methods.

The important thing to remember is that FME is just the name of an organisation. There is nothing limiting our activities, or our members, to Europe. Anyone can join FME, and there are no membership charges. We do not have big recruitment drives, preferring to have a smaller membership of "activists". Today there are about 100 members coming from the Americas, Asia, Australasia, Europe and Africa, about half from industry and half from academia. If you would like to join, simply complete a membership form, downloaded from http://www.fmeurope.org/fme/member.htm, and send it to the address shown on the form.

Special interests are served in FME through subgroups. The currently active groups are in Education & Training (led by José Oliveira at Universidade do Minho in Portugal) and the Formal Techniques Industry Association (ForTIA) [http://www.fortia.org], initiated last September and chaired by Anthony Hall of Praxis Critical Systems in the UK.

Like FACS, FME is run by an elected Board. There are currently five board members:

- John Fitzgerald (Chair);
- Nico Plat (Secretary);
- Kees Pronk (Treasurer);
- Stefania Gnesi (Symposia Officer);
- Bernhard Aichernig (Publications Officer).

We are considering expanding it with officers responsible for activities in Education and in Industry in the next couple of years.

FME's most public activities are the "FM" Symposia (note – no "E"!), held about once every 18 months. There have been 12 to date, if you go back to FME's earliest incarnation as VDM Europe. The 13th will be FM'05 in Newcastle upon Tyne, UK (see call for papers on Page 38 of the newsletter) and the 14th will be in Autumn 2006 at McMaster University in Canada.

FME has also been involved in several EU dissemination actions and other initiatives in the past. We have been involved in CoLogNet [http://www.cologne.net] and several sector-specific activities, especially in the Rail sector, through Dines Bjørner.

Online Resources

FME Website:
http://www.fmeurope.org

FME membership form:
http://www.fmeurope.org/fme/member.htm

ForTia Website:
http://www.fortia.org

Teaching Formal Methods workshop:
http://www.intec.rug.ac.be/groupsites/formal/Sympos2004/Symposs2004.htm

FM2005 Website:
http://www.csr.ncl.ac.uk/fm05/

FACS-FME Liaison

In 2003, Jonathan Bowen asked me to undertake the role of FACS-FME liaison, reporting to the FACS Committee. My role in FME is similar to Jonathan's in FACS. Below I give some information on the state of FME.

Events and Event Support

We are happy to sponsor events from a small budget, as does FACS. Our normal policy is to give about €500 towards the cost of a given speaker from the same continent as the event or, more exceptionally, up to €1000 for a speaker travelling further. FME and FACS both supported IFM this year: FME sponsored Tom Ball's invited talk (a report on IFM can be found on Page 19). FME will also be supporting ICFEM 2004 [http://research.microsoft.com/conferences/icfem2004] in Seattle and other events in Europe and the Americas. We welcome sponsorship proposals (please make an informal enquiry to John.Fitzgerald@ncl.ac.uk in the first instance).

Several members of the subgroup on Education and Training were involved with the FACS-supported conference in this area, held at Oxford Brookes University in December 2003 (see report in Issue 2004-1 of *FACS FACTS*). This led on to FME and CoLogNet establishing an International Seminar in the same topic. Further details can be found on Page 31.

Relaunch of Web site

Like FACS, FME has revamped its website. We used a design from Ghent University backed by a commercial CMS from West Consulting in Delft, who

also sponsor the site hosting. The site is maintained and developed by Bernhard Aichernig at UNU-IIST. Please have a look at:

http://www.fmeurope.org.

Formal Aspects of Computing Journal

FME members can now get the FAC journal on similar terms to FACS members. FME will be a close collaborator on the FAC journal in future, with an occasional column, starting with Volume 16, Issue 1. More details on the FAC journal can be found in John Cooke's report on Page 30. ■

AMAST 2004

10th International Conference on
Algebraic Methodology And Software Technology

12 – 16 July 2004, University of Stirling, Scotland

www.cs.stir.ac.uk/events/amast2004

Invited Speakers:

R. Backhouse (UK)
D. Batory (USA)
M. Bidoit (FR)
M. Calder (UK)
B. Jacobs (NL)
JJ. Meyer (NL)

AMAST 2004 provides an ideal opportunity to refresh your knowledge, and to meet leaders in the field of software technology and its mathematical foundations. AMAST 2004 is being held in Stirling, a historic city located in the heart of Scotland. The conference venue and accommodation are all based on the University campus, convenient for visiting the ancient castle and Wallace monument, or for going further afield and exploring the beautiful countryside. Social events include a civic reception, an afternoon excursion to Loch Lomond, and dinner in a quaint Scottish village.

5 days combining Algebra with Software, in a beautiful setting - what more could you want? You do want more? AMAST 2004 is co-located with:

- **MPC 2004** (Mathematics of Program Construction, 12-14 July)
- **ARTS 2004** (AMAST Real Time Workshop, 12 July)
- **CMPP 2004** (Constructive Methods in Parallel Programming, 14 July)

Participants for AMAST will be able to attend talks of all these events for *free* (and vice versa).

Late registration places are still available. Please contact the conference organisers on amast@cs.stir.ac.uk as soon as possible. EPSRC are supporting special reduced rate places for PhD students – **STOP PRESS: 10 places currently available!!**

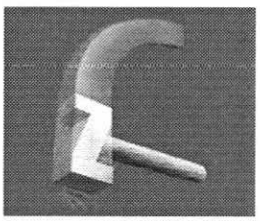
AMAST are please to accept the support of: the Edinburgh Mathematical Society, and the following organisations:

EPSRC BCS FACS

The AMAST proceedings will appear as Springer LNCS volume 3116.

Community Z Tools (CZT) Project

Mark Utting



The CZT project was proposed by Andrew Martin (Oxford University) in 2001 and is now actively developing Z tools on http://czt.sourceforge.net. Mark Utting (The University of Waikato, New Zealand) is managing the Java tool development and 5-10 programmers are busy working on various aspects of the tools.

The Z specification language was adopted as an ISO standard in 2002 (ISO/IEC 13568:2002). It can be used to precisely specify the requirements or behaviour of systems, and analyse that behaviour via proof, animation, test generation etc. However, one of the biggest barriers to the widespread use of the Z specification language seems to be the issue of tool support, especially for an integrated set of tools that support the ISO Z standard.

Later this year we expect to produce the first end-user release of the CZT tools, containing mark-up converters, parsers, type-checkers, pretty-printers and animators for the ISO Z standard. Many of these tools will also support Object-Z and TCOZ (Timed Communicating Object Z).

Right now, you can download a developer release that contains an elegant Java framework for building Z tools. This provides simple facilities for parsing Z and Object-Z specifications into annotated syntax trees, transforming those trees using the visitor design pattern, reading and writing specifications in XML format etc. As this Z tool development framework becomes more robust and offers more functionality, it will become easier and easier for student projects and research projects to produce new Z tools that are integrated into the CZT environment in a robust and reusable way. ■

Online Resources

Z user Group:
http://www.zuser.org

Z Specification Language:
http://vl.zuser.org

CZT Website:
http://czt.sourceforge.net

Object-Z website:
http://www.itee.uq.edu.au/~smith/objectz.html

The Third Irish Conference on the Mathematical Foundations of Computer Science and Information Technology, **MFCSIT2004**
University of Dublin, Trinity College.
22nd and 23rd July 2004

See http://www.cs.tcd.ie/MFCSIT2004 for further details

Formal Methods and Mathematics: a new collaboration

Mike Stannett, LMS-CS Committee Liaison Officer

Recent initiatives have seen links re-established and strengthened between BCS-FACS and the Computer Science Committee of the London Mathematical Society (LMS), with each group appointing a representative to the other's Committee. This increasing spirit of collaboration comes at an interesting time in the history of UK research, for as the recent *International review of UK Research in Mathematics* (March 2004, p. 27) explains:

> *The software parts of computer science were largely initiated by mathematicians, and mathematics continues to play a fundamental role in the development of algorithms. This has led to incredible growth in research on discrete mathematics and computer algorithms in other countries. This research area is underrepresented in the UK, both in mathematics and computer science departments. We recommend that the mathematics community takes a proactive role in addressing this issue, both by increasing research in the area and by establishing closer relationships with computer science departments. This also means that clear training paths at this interface have to be created, if possible with some visibility.*

It is hoped that our two organisations, working together, can help to close this gap in UK research, and to this end a number of possible collaborations are under discussion, such as shared support for conferences and research networks. Further details will be announced on the FACS mailing list and in *FACS FACTS* in due course. ■

Online Resources

International Review of UK Research in Mathematics:
http://www.cms.ac.uk/irm/irm.pdf

LMS Website:
http://www.lms.ac.uk

LMS CS-Committee:
http://www.lms.ac.uk/activities/comp_sci_com/index.html

Symposium on Logic-Based Agent Verification

9 July 2004

Liverpool, United Kingdom

http://www.csc.liv.ac.uk/~michael/symposium04.CALL

RefineNet Workshop, Praxis Critical Systems, 13–14 May 2004

Adrian Hilton, RefineNet Reporter

RefineNet is an EPSRC-funded collaboration of UK university research departments and firms from industry, similar to the FORTEST Network. The second network meeting was held at Praxis Critical Systems in Bath on May 13 and 14, and aimed to establish the problems with using refinement techniques and tools in an industrial context.

On Thursday, Wilson Ifill (AWE) discussed the use of code generation from specifications, using the B Toolkit and feeding code into DeCCo (the Demonstrably Correct Compiler). Michael Leuschel (University of Southampton) demonstrated the latest version of the Pro-B toolkit and discussed implemented and planned functional improvements.

Friday opened with a report from Martin Henson (University of Essex) from the Grand Challenge (GC6) Workshop. One key suggestion coming out of the resulting discussion was that RefineNet should be thinking about a Grand Challenge for refinement. Adrian Hilton (Praxis Critical Systems) spoke on the difficulties of refining specifications to architectures with diverse components. David Crocker (Escher Technologies) presented the current state of the Perfect Developer refinement tool and identified the main challenges for future functionality.

The primary result of the meeting was a list of the main problems identified over the two days. These will inform future research in the area of refinement, and in particular the next meeting which will be in early September at the University of York. The theme for this meeting will be case studies of refinement in practice.

Online Resources

RefineNet website:
http://www.refinenet.org.uk

RefineNet Case For Support:
www.cs.kent.ac.uk/people/staff/jd1/refinenet.ps

RefineNet inaugural meeting agenda:
http://cswww.essex.ac.uk/staff/henson/public/RefineNet

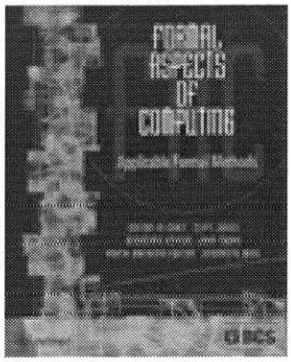
Photos taken at the inaugural meeting:
http://cswww.essex.ac.uk/staff/henson/public/RN/default.htm

The RefineNet website is at http://www.refinenet.org.uk and includes a list of current members, a schedule of future meetings and contact details. Enquiries are welcomed: enquiries@refinenet.org.uk. ■

If you are interested in becoming a member of FACS, please complete and return the membership form at the end of this issue of *FACS FACTS*.

FAC Journal: news from the editor's office

John Cooke, Associate Editor



So here we are on volume 16 of our journal. It certainly does not seem like 18 or 19 years since we first started negotiations with the BCS and Springer to start "The International Journal of Formal Methods" (that is our sub-title). We started with four issues per year and small format pages; increasing in frequency to six (and even seven) issues per year, and then to larger format pages, now four issues per year but with an increasing page count. And now, of course, we also have online access – but more about that later.

Recently we have had an increasing number of special issues but, as our declared area of interest is relatively narrow (in concepts and theory if not in application) this seems not to cause the kind of problem – reader alienation – experienced by some other journals whose content might be substantially at odds with their readers' interests and expectations. If you feel that we are 'drifting', please tell us; I guess that goes equally well for FACS as for the journal.

In step with many of our special issues we have a growing, yet informal, relationship with various FACS, and FACS sponsored, events such as conferences and workshops (most notably the derivatives of the FACS Refinement Workshops). We hope that these arrangements will preserve our international reputation and that, despite having been founded in the UK, the journal is in no sense parochial; quality and relevance are our only two watchwords.

Online Resources

FAC journal page:
http://www.bcs-facs.org/journal.html

Online access to FAC journal (for subscribers)
http://www.springerlink.com

Subscribe to the FAC journal:
http://www.bcs-facs.org/forms.html

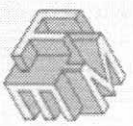
Of course the rate at which we publish papers depends on the rate at with they are submitted and upon their quality. This fluctuates greatly; and sometimes there are delays, sometimes there are bottlenecks. To help smooth out potential mismatches of journal input and output, Springer have a system called Online First. Basically this means that once a paper has been accepted, barring technical problems with the text, it can be accessed electronically within a matter of weeks, even if the printed issue to which it is assigned is not due for many months. Form the time it is accessible on the web, it has a DOI (digital object identifier) and can be cited just like a traditionally printed paper.

And so authors are happier, and subscribers to the journal have access to these papers much more quickly.

As of today, a glimpse at the FAC Online First page will show eight papers in the queue; four (together with two more papers currently in production) are destined for Volume 16, Issue 3; the others are likely to be in the subsequent issue.

So the 'view from the bridge' looks good. Please continue to support us. We shall strive to bring you high quality, accessible, relevant papers on formal methods as speedily as possible. ■

CoLogNet / Formal Methods Europe



Date: **18-19 November 2004**
Location: Ghent University, "Het Pand"
Local organisation: Ghent University, department: INTEC, group: Formal Methods

Motivation

Is education failing?

- Failing to convince students, academics and practitioners alike that Formal Methods are truly pragmatic;
- failing to overcome a phobia of formality and mathematics;
- failing to provide students with the basic skills and understanding required to adopt a more mathematical and logical approach to software development.

Until education takes these failings seriously, Formal Methods will be an obscure byway in software engineering, which in turn will remain severely impoverished as a result.

This symposium will serve as a forum to explore the failures and successes of Formal Methods education, to consider how the failings might be resolved, to learn from the successes, and to promote cooperative projects to further the teaching and learning of Formal Methods (FMs).

The symposium aims to bring together:

- Formal Methods educators, both actual and potential;
- other computer science and software engineering educators;
- industrial practitioners and project managers;
- technical and scientific publishers.

Invited talks will be given by

David Gries, Cornell University
Leslie Lamport, Microsoft Corporation
Peter Pepper, Technische Universität Berlin

Organisation

Symposium chair: Raymond Boute, Ghent University (B)
Organising chair: Jean-François Raskin, Brussels University (B)
Program chair: C. Neville Dean, Anglia Polytechnic University (UK)

For more information
http://www.intec.rug.ac.be/groupsites/formal/Sympos2004/Sympos2004.htm

X-machines: correctness via testing

Mike Stannett

Background

X-machines were introduced some thirty years ago, as a minor topic in Eilenberg's comprehensive introduction to *Automata, Languages and Machines*. He used his new model, which is similar to the EFSM (*Extended Finite State Machine*), to give an essentially algebraic description of computation.

Imagine for a moment that we want to model the behaviour of a simple calculator. We can think of this as a machine for manipulating numbers, or as Eilenberg would say, a *number-machine*. In the same way, a word processor is a device for manipulating documents, or a *document-machine*. More generally, an *X-machine* is a machine that manipulates entities of type *X*, where we choose the *fundamental data type*, *X*, to be whatever seems most convenient for the task in hand. To keep things simple, the control structure of an *X*-machine is taken to be a finite state machine (FSM), *i.e.*, we consider the various categories of configuration the device can enter, and use an FSM to model the transitions between them. As we will see below, it is precisely this decision to keep the control structure simple that makes *X*-machines so useful. Unlike an FSM, however, transition labels are not simply atomic symbols. Instead, we require them to be elements of $R(X)$, the set of (binary) relations on *X*. In most practical situations, these labels will be functions of type $X \to X$, rather than proper relations on *X*, *i.e.* subsets of $X \times X$.

Informally speaking, the behaviour of an *X*-machine is fairly straightforward. Suppose we have an object *x*, and decide to operate on it with the machine. Each manipulation the machine carries out is indicated by the label of a corresponding transition, and crossing a transition labelled by a function f: $X \to X$ is equivalent to performing the computation $x \to f(x)$. If we now cross another transition, g, the overall effect will be to compute $g(f(x))$, and so on. Every successful path through the machine's underlying FSM is deemed to correspond to a successful computation on *X*, and we define the behaviour, $|M|$, of a machine M to be the conjunction of all such computations. So an *X*-machine M is considered capable of successfully computing a transformation $x \to s(x)$ precisely when there is some successful path through the control structure that computes s.

Formally, we can model this by defining an *X*-machine, M, to be a finite state machine, F, together with a *behavioural labelling* function, Λ: $A \to B$, where A is the alphabet of F and B is a subset of $R(X)$. We sometimes write $M = F^{\Lambda}$ to stress the relationship between an *X*-machine M and its underlying FSM, F, and similarly, if a is one of the transition labels in A, we can write a^{Λ} instead of

$\Lambda(a)$. The labelling Λ extends naturally to finite strings over A, provided we define $(ab\ldots c)^\Lambda$ to be the relational composition $a^\Lambda \circ b^\Lambda \circ \ldots \circ c^\Lambda$. This in turn extends to give a natural function on languages, defined by

$$L^\Lambda = \cup\{ w^\Lambda \mid w \in L \}$$

In particular, if L is taken to be $|F|$, the language recognised by F, then L^Λ is the behaviour, $|M|$, of M. That is, X-machines embody the commutative relationship $|F^\Lambda| = |F|^\Lambda$.

The structure of X-machines lends itself well to top-down design and refinement. For suppose M contains a transition, f. If we can establish that a second machine, N, also has behaviour f (and we show how to do this below), then we can legitimately replace all of M's f-transitions with copies of N, without affecting the overall behaviour $|M|$.

Example – A Simple Calculator

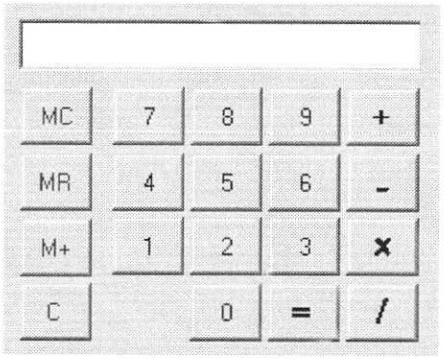


Figure 1. A simple calculator

Consider the simple calculator in *Figure 1*. In general, this calculator needs to store three different values at the same time, which we will call *total*, *memory* and *display*. We can think of each of these as belonging to *Int*, the type of integers. In addition, the calculator needs to remember which operator key the user last pressed. We will model this using the type $Op = \{doPlus, doMinus, doMultiply, doDivide\}$, whose members apply the binary functions in question; *e.g.*, $doPlus(m,n) = m+n$, and $doDivide(m,n) = m/n$.

Armed with this information, we choose the fundamental datatype of this machine to be $X = Int \times Int \times Int \times Op$, and assume an underlying finite state machine similar to that shown in *Figure 2*. To complete the model, we need to define the various transition labels as relations on X. Some of these relations might be defined as follows:

```
// Clear the contents of memory (MC)
MC^Λ (total, mem, display, op) = (total, 0, display, op)

// Display the contents of memory
MR^Λ (total, mem, display, op) = (total, mem, mem, op)

// Press the + key (and likewise for -, *, /)
+^Λ (total, mem, display, op) = (newtotal, mem,
newdisplay, doPlus)
      where newtotal = op(total, display)
            newdisplay = newtotal
```

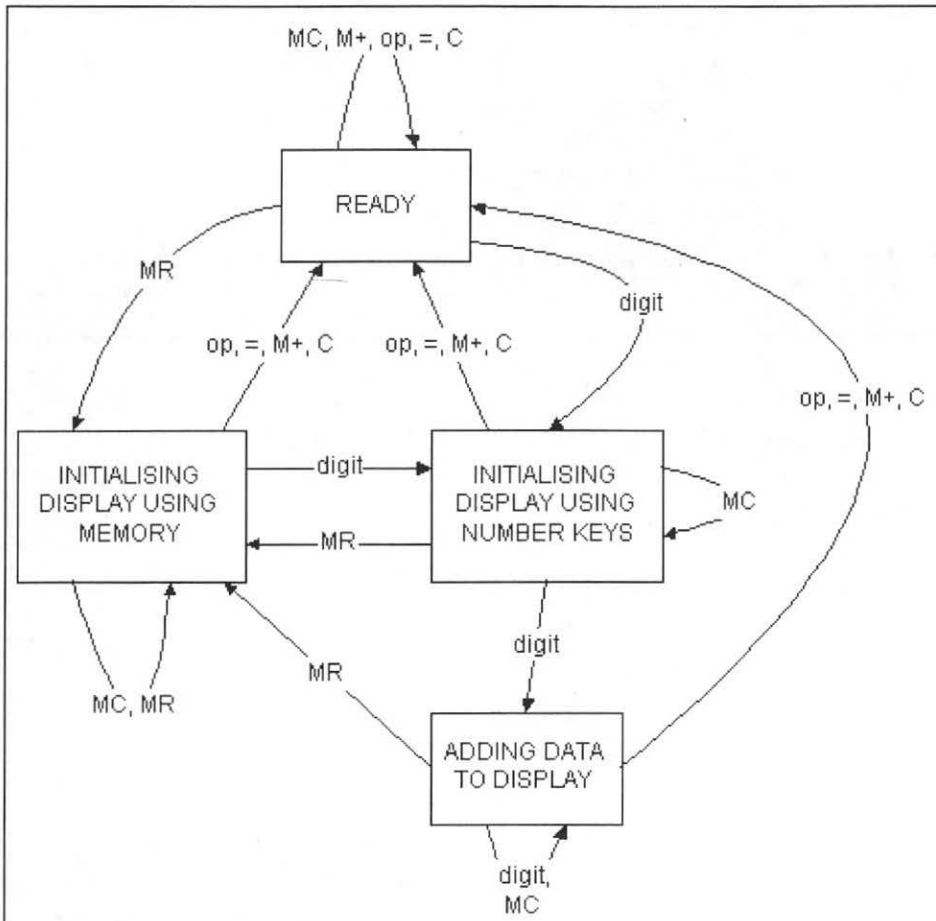


Figure 2. A possible XM representation of the simple calculator in *Figure 1*. The state *READY* is both the initial, and the only terminal, state.

```
// Press the equals key (=)
=^ (total, mem, display, op) = (newtotal, mem,
newdisplay, none)
      where newtotal = op(total, display)
            newdisplay = newtotal

// Press the 7 key (and likewise for other digits)
7^ (total, mem, display, op) =
    {(total, mem, 7, op), (total, mem, display⊕7, op)}
```

Stream X-machines and Testing

Notice that our *X*-machine model of a calculator is non-deterministic – it uses proper relations on *X* because there are two possible outcomes whenever we press a digit key (the result depends on whether or not the last key pressed was also a digit). This ambiguity indicates that the machine in *Figure 2* is an under-specification of the calculator. One way to solve this problem would be to introduce new transition labels – the transition emerging from the *ADDING*

DATA state might be renamed '*append7*', thereby allowing us to disambiguate the two behaviours. Alternatively, we can give the machine access to its past behaviours. Doing so leads us to introduce the *stream X-machine* (SXM).

Originally introduced by Gilbert Laycock, a member of Mike Holcombe's Sheffield group, the SXM is a variant of the *X*-machine in which we require *X* to have the form

$$X = In^* \times Mem \times Out^*.$$

Here, *In* is a data type of *inputs*; *Out* is a data type of *outputs*; and *Mem* is the machine's *memory* type, which is used to store all other relevant information. Each SXM transition label can be thought of as a function f: $In \rightarrow (R(Mem) \times Out)$, for suppose we are in a state s for which $f(in)$ is defined and equals (r, out). If the input *in* is received, this transition can be triggered, in which case r is applied to the current memory, *mem*, and *out* is appended to the output stream. (One can also envisage versions of the SXM in which the value *out* depends upon *mem*.)

Just as *X*-machines are based on finite state machines, so the SXM is closely related to the finite state transducer (FSM with output). Laycock suggested that the extensive literature on transducer testing might, therefore, be applied to stream *X*-machines, an idea that was subsequently developed to great effect by Holcombe, Ipate and others in the late 1990s. In particular, various researchers with the Sheffield group noted that Chow's well-known *W-Method* could be used to determine whether or not two finite state transducers – one a minimal machine (the *specification*), the other a potentially non-minimal *implementation* – have identical behaviours. All that is required is that (*i*) they are defined over the same alphabets; (*ii*) we have a rough estimate as to the number of additional states in the implementation; and (*iii*) we can supply inputs at will, and observe the resulting outputs.

The disadvantage of transducers is that they can only be used to represent regular behaviours, whereas many commercial systems are of far greater computational power. In contrast, both the *X*-machine and SXM models can be used to capture arbitrary Turing machine behaviours. But it is also the transducer's limited computational power that makes testing feasible – establishing the behavioural equivalence of arbitrary Turing machines is well known to be an undecidable problem. An obvious question is, therefore,

> *Is it possible to use the simplicity of transducers to ensure the decidability of SXM equivalence, without at the same time restricting the computational power of the SXM?*

Online Resources

X-machines:
http://x-machines.com

Object-oriented testing:
http://oo-testing.com

Methodology for Object Testing, Integration and Verification (Motive):
http://www.dcs.shef.ac.uk/~ajhs/motive/

The answer, as demonstrated by Ipate and Holcombe, is a guarded *yes*. What matters is how you *build* your software in the first place.

Design-for-Testability

Hardware designers are well aware of the need to design their products in such a way as to ensure their easy testability. SXM Theory shows how to apply the same idea to computational systems in general.

The first thing we require is an SXM-style specification of the system we want to build, which we will call *Spec*. Because we have full access to *Spec*, we will assume that the underlying transducer is minimal (if not, we can minimise it). We also require that any (alleged) implementation be describable as a stream *X*-machine, which we will call *Imp* – we do *not* assume any knowledge as to the internal structure of *Imp*, beyond the following basic conditions:

- We need to know an upper bound on how many extra states are present in *Imp* as opposed to *Spec*.
- We need to know what components *Spec* and *Imp* are constructed from, and it must be possible to distinguish one type of component from another. In other words, we need to identify which relations occur as labels in the SXM.
- We need to know that these labels are themselves correctly implemented. If they are too complex for this to be determined, we need the designer to refine the implementation until all of the required components are so simple, or so trusted, that their correctness can be established by other means.

Provided these conditions are satisfied, an adapted form of Chow's method can be use to generate a finite set, *Tests*, of finite tests with the property that

> If *Spec* and *Imp* behave identically when presented with each of the tests $t \in Tests$, then $|Spec| = |Imp|$.

In other words, the correctness of an implementation can be determined simply by testing it.

Although researchers are still assessing the applicability of this approach (commonly called the *Stream X-Machine Testing Methodology*, or *SXMT*), work by Bogdanov, Simons, Vanak and others suggests that SXMT can be applied successfully to a number of practical problems, in both software and hardware.

Continuing developments

Impressive though the SXMT methodology is, it is not obvious how to apply it to certain key classes of system. In particular, *X*-machine structures tend to emphasise functional behaviours (machines *operate upon* values), which makes them ill-suited to modelling concurrent or object-oriented systems. Current research in the UK and elsewhere is pushing back these boundaries. Various *concurrent X-machine* models have been developed, which allow

independent devices to interact with one another. And a recent EPSRC project, *Motive* (*Methodology for Object Testing, Integration and Verification*), has introduced a number of techniques for extending SXMT to object-oriented verification; negotiations to develop these techniques further are currently underway with a major player in the UK IT industry. My own work in Sheffield seeks to extend the technique to *all* systems for which the term 'behaviour' can be applied, including systems specified using concurrent process algebras like CSP, CCS and the π-calculus. More generally, Rob Hierons and his colleagues at Brunel have extended SXM techniques to test for behavioural conformance of non-deterministic systems rather than simple equivalence, and researchers as far afield as Greece and Romania continue to play an important role in the development of the underlying theory.

In summary, *X*-machines have come a long way since their introduction thirty years ago. They have been used to develop 'super-Turing' models of computation; to model social insect societies; and as the foundation for a commercially significant, novel, testing methodology. It is a great time to be an *X*-machine theorist!

Suggestions for Further Reading

Online Resources

The website for *X*-machine research at http://x-machines.com includes a short history of the topic, an extensive bibliography, and links to other relevant sites. These include http://oo-testing.com, a repository for MOTIVE and other object-oriented testing materials. Both these sites are under development, and include a growing collection of significant resources.

General Theory of X-machines

- Eilenberg, S. (1974). *Automata, Languages and Machines, Vol. A.* Academic Press.
- Stannett, M. (1990). 'X-machines and the Halting Problem: Building a super-Turing machine.' *Formal Aspects of Computing* **2**, pp. 331–341.
- Stannett, M. (2001). *Computation over arbitrary models of time.* Technical Report CS-01-08, Dept of Computer Science, Sheffield University, UK. Available online from:

  http://www.dcs.shef.ac.uk/research/resmes

SXM Testing Techniques

In addition to the articles cited below, readers may find it helpful to consult Volume 12, Issue 6, of the *Formal Aspects of Computing* journal. This special issue is devoted entirely to *X*-machines and the SXM Testing Methodology.

- Bogdanov, K. & Holcombe, M. (2001). 'Statechart testing method for aircraft control systems.' *Software Testing, Verification and Reliability* **11**, pp. 39-54.

- Hierons, R. & Harman, M. (2000). 'Testing conformance to a quasi-nondeterministic stream X-machine.' *Formal Aspects of Computing (Special Issue on X-Machines)* **12**, pp. 423-442.
- Holcombe, M. & Ipate, F. (1998) *Correct Systems: Building a Business Process Solution.* Springer.
- Laycock, G. (1993). *The Theory and Practice of Specification-Based Software Testing*. PhD Thesis, Dept of Computer Science, Sheffield University, UK.

Call For Papers

Photo by Graeme Peacock, used with permission

http://www.csr.ncl.ac.uk/fm05

24 January 2005	Paper submission deadline
07 March 2005	Workshop & Tutorial proposals
09 April 2005	Decisions on papers
02 May 2005	Final versions of papers due
09 May 2005	Tools Exhibition & Demonstration proposals

News from the BCS

Margaret West, BCS Liaison Officer for FACS

Changes to the BCS membership structure

www.bcs.org

The BCS have modernised their grading structure and entry procedures. The idea is to make BCS membership more accessible to IT practitioners and the key feature is the separation of Chartered membership into two elements, professional membership and chartered status. Speed of processing is an essential feature of the new system and a target of the BCS is to admit new members within **three** days of receiving their applications. For more information see the BCS new members web page:

http://www.bcs.org/BCS/Join/welcome.htm

Report on the Challenges of Complex IT

A report has been published by the BCS in conjunction with the Royal Academy of Engineering: *The Challenges of Complex IT*. It tells a familiar story – however this is well worth repeating! Also there are some new perspectives including some case studies highlighting both good and bad practice. The authors of the report provide some advice to Managers of IT projects and recommend that Management schools ensure IT is a core module of future MBA courses.

It will not surprise our members to hear the following from the report by Martyn Thomas:

"Few projects capture the users' requirements using notations and methods that make it possible to analyse them rigorously to find contradictions and omissions, even though these are among the most common causes of project delays and failures."

The final recommendation of the authors is that the DTI and EPSRC establish a UK research program on complex IT systems.

The 'Complexity' report can be downloaded from the BCS website:

http://www.bcs.org/BCS/News/PositionsAndResponses/Positions/complexity.htm

The FACS Website: update and improvements

Mike Stannett, FACS Webmaster

Originally run from Loughborough, the FACS web site was redesigned in 2000 and moved to its current server at London South Bank University in September 2002, since when it has received just over 107,000 hits. The site is accessible under http://www.bcs-facs.org (and http://www.bcs-facs.org.uk). The current layout includes a news ticker, event listings, and links to related organisations; it offers easy access to the online version of *FACS FACTS*, as well as membership renewal forms, details of the Constitution, Committee member contact details, and so forth.

We see the web site as a resource to be exploited by and for the benefit of FACS members, and all members are warmly invited to suggest new applications for inclusion. If you have any comments or suggestions, please send these to our general contact email address info@bcs-facs.org.uk. We regularly feature details of forthcoming conferences and workshops, including the relevant links for Calls for Papers, and intend making the site increasingly useful over the coming months. If there are any conferences you think we should feature on our site, please let us know.

The next stage in the site's development will see a wholesale migration to PHP, so that detailed access controls can be introduced; the detailed design is still under discussion, but we hope to introduce such utilities as discussion boards, online membership renewal, a feedback facility, comprehensive search facilities, and user-profiling (so that individual members can add and edit details of their research or industrial interests, and match these against other members' interests). Please check the FACS website from time to time and let us know what you think of it. ■

Newsletters from other organisations

CoLogNET

http://www2.cs.ucy.ac.cy/projects/colognet

EA-SST

http://www.easst.org/newsletter

http://www.liacs.nl/~beatcs

http://www.lms.ac.uk/newsletter/newsletter.html

Workshop and Conference Announcements

The following is a selection of the large number of calls for papers and conference announcements that have been received. More listings can be found on FME's events page linked from http://www.fmeurope.org.

Conference	25 Years of Communicating Sequential Processes
Dates	**7 – 8 July 2004**
Location	London South Bank University
Deadline	N/A
URL	http://www.lsbu.ac.uk/menass/csp25
Conference	CONCUR 2004: 15th International Conference on Concurrency Theory
Dates	31 August – 3 September 2004
Location	Royal Society, London
Deadline	PASSED
URL	http://www.doc.ic.ac.uk/concur2004
Conference	CPA 2004: Communicating Process Architectures
Dates	5 – 8 September 2004
Location	Oxford Brookes University, UK
Deadline	13 August (Registration)
URL	http://cms.brookes.ac.uk/computing/cpa2004/
Conference	FMICS04: 9th International Workshop on Formal Methods for Industrial Critical Systems
Dates	20 – 21 September 2004
Location	Linz, Austria
Deadline	PASSED
URL	http://www.fmics04.cclrc.ac.uk
Conference	ICTAC 2004: 1st International Colloquium on Theoretical Aspects of Computing
Dates	20 – 24 September 2004
Location	Guiyang, China
Deadline	**5 July 2004 (Extended)**
URL	http://www.iist.unu.edu/ICTAC2004/
Conference	FATES 2004: Formal Approaches to Testing of Software
Dates	21 September 2004
Location	Linz, Austria
Deadline	PASSED
URL	http://fates.cs.auc.dk
Conference	SEFM 2004: 2nd IEEE International Conference on Software Engineering and Formal Methods
Dates	26 – 30 September 2004
Location	Beijing, China
Deadline	PASSED
URL	http://www.iist.unu.edu/SEFM2004/

Conference	CSD-UML 04: 3rd International Workshop on Critical Systems Development with UML
Dates	12 October 2004
Location	Lisbon, Portugal
Deadline	**15 July 2004**
URL	http://www4.in.tum.de/~csduml04/
Conference	FMCO 2004: 3rd International Symposium on Formal Methods for Components and Objects
Dates	2 – 5 November 2004
Location	Leiden, The Netherlands
Deadline	20 September 2004 (Registration)
URL	http://fmco.liacs.nl/fmco04.html
Conference	ICFEM: 6th International Conference on Formal Engineering Methods
Dates	8 – 12 November 2004
Location	Seattle, USA
Deadline	PASSED
URL	http://research.microsoft.com/conferences/icfem2004
Conference	FMCAD 2004: 5th International Conference on Formal Methods in Computer Aided Design
Dates	14 – 17 November 2004
Location	Texas, USA
Deadline	PASSED
URL	http://www.cs.utexas.edu/users/hunt/FMCAD/2004/index.html
Conference	BCTCS 2005: 21st British Colloquium for Theoretical Computer Science
Dates	22 – 24 March 2005
Location	University of Nottingham
Deadline	7 February 2005
URL	http://www.cs.nott.ac.uk/~gmh/bctcs05.html
Conference	ASWEC 2005: Australian Software Engineering Conference
Dates	29 March – 1 April 2005
Location	Brisbane, Australia
Deadline	1 October 2004
URL	http://aswec2005.itee.uq.edu.au
Conference	ETAPS 2005: European Joint Conference on Theory and Practice of Software
Dates	2 – 10 April 2005
Location	Edinburgh, Scotland
Deadline	8 October 2004
URL	http://www.etaps05.inf.ed.ac.uk
Conference	ZB2005: 4th International Conference of B and Z Users
Dates	13 – 15 April 2005
Location	University of Surrey
Deadline	1 November 2004
URL	http://www.zb2005.org
Conference	FM2005: 13th International Symposium of Formal Methods
Dates	18 – 22 July 2005
Location	University of Newcastle Upon Tyne, UK
Deadline	24 January 2005
URL	http://www.csr.ncl.ac.uk/fm05/

Book Announcement

Categories for Software Engineering, J L Fiadeiro, Springer-Verlag 2004, ISBN 3-540-20909-3, due in the Autumn

This book provides a gentle introduction to category theory oriented to software engineering. Assuming only a minimum of mathematical background, this book explores the use of categorical constructions from the point of view of the methods and techniques that have been proposed for the engineering of complex software systems: object-oriented development, software architectures, logical and algebraic specification techniques, models of concurrency, *inter alia*. After two parts in which basic and more advanced categorical concepts and techniques are introduced, the book illustrates their application to the semantics of CommUnity – a language for the architectural design of interactive systems.

Written for: Advanced students, professionals, lecturers

Keywords: Agent-Oriented Software Engineering, Categories, Category Theory, CommUnity, Complex Systems, Component-Based Systems, Coordination Languages, Formal Methods, Object-Oriented Software, Service-Oriented Software Development, Software Architecture, Software Engineering, Software Specification, Systems Design, Systems Modelling, Systems Theory.

Paid-up BCS-FACS members receive a generous **25%** discount on books published by Springer-Verlag and **20%** discount on the Requirements Engineering Journal. If you would like to take advantage of these discounts, please contact Springer-Verlag London directly on:

journals@svl.co.uk

FACS Committee

Jonathan Bowen
FACS Chair
ZUG Liaison

Jawed Siddiqi
Treasurer

Roger Carsley
Secretary

Paul Boca
Membership Sec.
Newsletter Editor

John Cooke
FAC Journal
Liaison

Ali Abdallah
Events Coordinator

John Fitzgerald
FME Liaison
SCSC Liaison

Margaret West
BCS Liaison

Mike Stannett
Webmaster
LMS-CS Liaison

Judith Carlton
Industrial Liaison

Kevin Lano
UML Liaison

FACS is always interested to hear from its members and keen to recruit additional Committee members. At the moment, we have vacancies for a Publicity Officer and EATCS Liaison Officer. If you are interested in helping the Committee, please contact the Chairman, Professor Jonathan Bowen, at the contact points below:

BCS FACS

c/o Prof. Jonathan Bowen (Chair)
London South Bank University
Faculty of BCIM
Borough Road
London SE1 0AA
United Kingdom

Tel. +44 (0)20 7815 7462
Fax. +44 (0)20 7815 7793
Email. info@bcs-facs.org.uk
Web: www.bcs-facs.org

You can also contact 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).

Coming Soon in FACS FACTS....

Reports on:

Mathematics of Program Construction Conference

25 Years of CSP conference

Next RefineNet meeting

LMS Computer Science Committee

CoLogNet / FME Symposium on Teaching Formal Methods

AMAST conference

Formal Methods in New Zealand

PhD Summer School on Logics of Formal Software Specification Languages

Jobs

CLOSING DATE JULY 16

Reader in Software Engineering
Department of Computer Science
University of Leicester, UK

Available from 1 September 2004
£37,558 to £42,573 pa (Aug 2004 rates)

Ref: A1028

The University seeks to appoint a Reader in Software Engineering. The successful applicant will be ambitious, able to develop and lead research within a multi-faceted environment. We are looking for candidates with a proven strong research record that would normally include direct participation in PhD supervision and successful grant applications. There will be ample scope for persons of energy, drive and ambition to assume a rewarding role in a young and dynamic department. Preference will be given to applicants who can build upon, and extend, the existing research in the Department in the general area of Software Specification and Design; candidates with research strengths in fields such as collaborative systems, domain modelling, embedded systems, re-engineering, requirements engineering, service-oriented software development, software architectures, software evolution and software reliability are particularly encouraged to apply. However, outstanding applicants with expertise in any related area of Software Engineering are most welcome. The post is available from 1st September 2004, or as soon as possible thereafter.

Downloadable application forms and further particulars are available from www.le.ac.uk/personnel/jobs/a1028.html or from the Personnel Office, tel: 0116 252 5114, fax: 0116 252 5140, email: jobs@le.ac.uk. **Please note that CV's will only be accepted in support of a fully completed application form.**

Closing date: July 16, 2004

Candidates are invited, if they so wish, to contact Professor J. Fiadeiro, Head of the Software Specification and Design Group (email: jose@fiadeiro.org, telephone: +44 (0)116 252 3907), Professor Rajeev Raman, Head of Computer Science (email: r.raman@mcs.le.ac.uk, telephone: +44 (0)116 252 3894), who will be pleased to discuss the Readership further.

BCS FACS

FACS membership application/renewal (2004)

Title (Prof/Dr/Mr/Ms) _____ **First name** _____ **Last name**_____

Email address (required for options * below)_____

BCS membership No. (or sister society name + membership number)

_____

Address _____

_____

_____

Postcode _____ **Country** _____

I would like to take out **membership to FACS** at the following rate:

- ❑ £15 (Previous member of BCS-FACS now retired, unwaged or a student)
- ❑ £15 (Member of BCS or sister society with web/email access)*
- ❑ £30 (Non-member or member of BCS or sister society without web/email access)

In addition I would like to subscribe to **Volume 16 of the FAC journal** at the following rate:

- ❑ £42

For electronic only journal subscription*, please tick here ❑. No further discount given.

The total amount payable to BCS-FACS in **pounds sterling** is **£ 15 / 30 / 57 / 72** (delete as appropriate). I am paying by:

- ❑ Cheque made payable to **BCS-FACS** (**in pounds sterling**)
- ❑ Credit card via PayPal (instructions can be found on the BCS-FACS website)
- ❑ Direct transfer (in **pounds sterling**) to:

  Bank: Lloyds TSB Bank, Langham Place, London
  Sort Code: 30-94-87
  Account Number: 0173977
  Title of Account: BCS-FACS

If a receipt is required, please tick here ❑ and **enclose** a stamped self-addressed envelope.

Send completed forms to:

Dr Paul P Boca
PO BOX 32173
LONDON N4 4YP
UK

For FACS use only

Received by FACS	Date:	Initials:
Sent to Springer	Date:	Initials:
Actioned by Springer	Date:	Initials: