

PROPOSAL TO EPSRC: MATHFIT INITIATIVE 2002

LOGICAL STRUCTURES FOR CONTROL

CASE FOR SUPPORT

PROFESSOR URSULA MARTIN,

DR ROY DYCKHOFF

SCHOOL OF COMPUTER SCIENCE

UNIVERSITY OF ST ANDREWS

ST ANDREWS KY16 6SS

SCOTLAND

um@dcs.st-and.ac.uk

P: +44 1334 463252

F: +44 1334 463278

CONTENTS

A	SUMMARY, AIMS AND OBJECTIVES	1
B	RELEVANCE TO MATHFIT	1
C	BACKGROUND	2
D	PROGRAMME AND METHODOLOGY	3
E	RELEVANCE TO BENEFICIARIES	4
F	DISSEMINATION AND EXPLOITATION	4
G	JUSTIFICATION OF RESOURCES	4
H	PROJECT PLAN AND PROGRAMME OF WORK	5
	REFERENCES	6
	APPENDIX I PROJECT PLAN	A
	APPENDIX II TRACK RECORD, EXPERTISE AND COLLABORATION	B

OCTOBER 2002

A SUMMARY

This proposal, from Ursula Martin and Roy Dyckhoff of St Andrews, concerns the interpretation of the mathematics of feedback control as a computational phenomenon. It is part of a long term program to apply the techniques of computational logic to mathematics and its applications, including pure mathematics [1], symbolic computation [2,3], numerical libraries [19] and mathematical modelling [5].

To control an object means to influence its behaviour so as to achieve a desired goal. Control systems may be natural mechanisms, such as cellular regulation of genes and proteins by the gene control circuitry in DNA. They may be man-made – an early mechanical example was Watt’s steam governor – but today most man-made control systems are digital, for example fighter aircraft or CD drives. In genomics we want to identify the control mechanism from observations of its properties: in engineering we want to solve the dual problem of constructing a system with certain properties. Traditionally, control is treated as a mathematical phenomenon, modelled by continuous or discrete dynamical systems. Numerical computation is used to test and simulate these models, for example MATLAB is an industry standard in avionics. A largely separate process is then used in the implementation of such as continuous model as a digital (discrete) controller.

In the account above, the controller, whether natural or man-made, is treated first as a dynamical system and then represented computationally. We seek rather to view control as a computational phenomenon from the start, so that we can use established notions from computational logic to understand and reason about it. This is a familiar idea in other fields: for example the representation of hardware devices as state-machines allows the use of model-checking to reason about temporal aspects.

Block diagrams are a standard engineering representation of dynamical systems, obtained from a Laplace transform. We have recently completed a pilot project, funded by Qinetiq, in which we added assertions about phase and gain of a signal to block diagrams and devised and implemented a simple Hoare logic. We were able to emulate mechanically engineers informal reasoning about phase and gain [22]. At the suggestion of Qinetiq’s control engineers we used this to prototype symbolic, automated, very general alternatives to some standard numeric tests used in engineering design [26]. This is evidence for the feasibility of our long term goal. Thus the AIM of this project is **to develop logical techniques to model continuous and discrete dynamical systems, with a focus on the needs of practical applications.** Our measurable OBJECTIVES are:

- (i) **to establish a logical framework, assertion language and inference system for understanding and reasoning about continuous and discrete control**
- (ii) **to validate our ideas against examples drawn from avionics and biological applications**
- (iii) **a prototype implementation using existing methods and tools such as Simulink, Clawz and PVS.**

The proposers have a strong track record in mathematics, logic and computer science. They will collaborate with Martin Hyland (Maths, Cambridge), Gordon Plotkin (Computer Science Edinburgh), Peter Saunders (Maths, Kings London), David Gilbert (Bioinformatics, Glasgow), a variety of close-at-hand mathematical experts (algebra, analysis, differential equations) in St Andrews, and industrial control engineers and computer scientists at Qinetiq (O’Halloran, Patel, Hall), SRI International (Tiwari, Rushby) and NASA Langley (Butler). In particular our earlier work benefited enormously from the patience and insight of working engineers at Qinetiq, and we expect to continue this synergy in this project.

B RELEVANCE TO MATHFIT We propose a **novel interaction**: viz the deployment of mathematics and logic on a novel and timely problem at the interface of computer science and mathematics of potentially high impact. Thus, citing from **the aims of MathFit**, the **relevance to MathFit** is that we propose a **new interdisciplinary interaction** including **a dynamic exchange of ideas, relevant to research ... in computer science** leading to **applications of mathematical research that address computer science challenges and problems...**and (we hope) to **new developments in computer science inspired and motivated by research activity in the other domain** and, through the appointment of a suitable RA and research student, and our activities in the EPSRC NETCA network, **a transfer or training of people with research expertise in one domain to apply these skills to research in the other.**

C BACKGROUND

Control engineering is a large subject and depends on many parts of pure and applied mathematics: we intend initially to focus on those aspects which are fairly standard and widely used in practice [5,6]. *Optimal control* assumes that a model of the system is available and one wants to optimise its behaviour, using the calculus of variations and so forth: for example pre-computing a desired flight-path for a spacecraft. *Feedback control* compensates for uncertainty in the model by using feedback to correct for deviations for desired behaviour: for example if the spacecraft strays off course. Models vary according to the application: differential equations are used when modelling a continuous signal, but these are replaced by difference equations when modelling a sampled signal as used in digital systems. In reasoning about such systems we are interested not only in the solutions, but in other properties, for example reachability, transient behaviour, limiting behaviour (stability), behaviour under perturbation, or chaotic phenomena. In practice systems are rarely linear: non-linear systems are generally treated “locally” by linearising at points of interest. “Global” behaviour of non-linear systems is subject of much research and raises subtle questions in differential and algebraic geometry: for example Liapunov methods involve constructing a suitable function that bounds the solution, another standard method involves Lie rings.

In “classical” control a Laplace transform is applied to a linear system to obtain a representation as a transfer function, a rational function over the complexes. Analysis of properties, such as frequency and response of the control system, is in terms of the position of its roots and poles in the complex plane. So called “modern” control considers a state-space representation, which replaces a single differential equation with a system of simultaneous equations in the state variables, and analyses the system via properties of the eigenvalues of a related matrix. Both frameworks can be extended from SISO (single input) to MIMO (multiple input) systems.

Block diagrams are often used to represent systems with feedback graphically, for example in classical control a block diagram is a directed graph whose edges are labelled by rational functions over the complexes. They also allow more general representation of components described only by their input/output behaviour, corresponding to a more general notion of state.

Software such as the widely used Mathworks Simulink [7], the industry standard in avionics and automotive applications, supports numerical tests and simulations. A number of standard tests are used for prediction and analysis: for example the Nichols plot displays the steady state behaviour of a “classical” control system in terms of the phase and gain of a sinusoidal input. The control requirements of fighter aircraft, which strike a balance between aircraft responsiveness and pilot discomfort, are specified in terms of acceptable paths in this plot [8].

Man-made control systems are typically digital embedded software systems, which use sampled, rather than continuous time. These can be modelled as discrete dynamical systems (difference equations), which again admit a transform representation via the z-transform, and an analogous state-space representation, investigated as before using matrix algebra. The design of a digital controller, for example in avionics applications, typically involves analysis as above in continuous time: it is then passed to a software team for implementation as a discrete digital system. It has been suggested that this process is a likely source of error: indeed apparently “similar” continuous and discrete systems may have very different stability properties. The ubiquity of such embedded controllers, for example in cars and domestic appliances, has led to increased interest in methods of generating assured code straight from a high level design [9].

In biological applications the process is reversed: we want to infer the model from knowledge of its properties. Typically in investigating transcriptional control (the cellular regulation of genes and proteins), microarray analysis is used to gather data on response to perturbations, for example in temperature or concentration of a substance [10]. For example our collaborator Saunders [11] was able to postulate a control mechanism for blood sugar, and why it failed (essentially a feedback which changed sign) in people with type 2 diabetes.

Our thesis is that a control system can be regarded directly as a computational process, of which the various mathematical descriptions, whether of a continuous or discrete system, can be regarded as a high-level specification or representation. Thus it is natural to think of this process in logical form, and to extend and apply familiar computer science techniques to understand, model and reason about it. In general terms one might

expect to annotate a representation, for example nodes in a block diagram, with assertions, and use a logic to reason about the assertions. Thus, for example, the numeric plotting described above can be viewed as an assertion about the output from a given real input to a complex function: one might hope to replace it with an automated analysis of properties of more general state variables.

The study of control in the context of computer science is an emerging area, we identify some strands of work which complement this proposal:

- The most well developed is the field of hybrid systems, which models certain control systems as state-transition diagrams, whose nodes represent continuous control modes and whose edges represent mode switching between them. Such models are amenable to model checking [12] and theoretical analysis [13], especially in terms of semi-algebraic sets, which can express, for example, reachability. Alur and Dill [29] introduced timed automata, state-transition diagrams annotated with timing constraints using finitely many real-valued clock variables which can be used to model discrete dynamical systems
- In the 1970s Arbib and Manes [14] studied categorical models of linear control: more recently various categories with feedback have been much studied, especially traced monoidal [15] categories, which are models for linear logic. These seem to obey similar algebraic laws to block diagrams with feedback, though as far as we know the connection has not been developed formally. Anderson and Tourlas [16] have studied reasoning about general diagram languages.
- Less attention has been paid to the classical dynamical systems representations. Perhaps the closest foundational work is Edalat's [17] extension of classical domain theory to analysis and dynamical systems. Tiwari [18] allows abstraction of dynamical systems to a level where model checking can be used.
- Our own work on light formal methods for mathematical systems [19,20] was a precursor to this proposal: NAG Ltd funded us to devise an assertion language and lint-like checker for their AXIOM system.
- The widespread use of Simulink suggests that effective formal verification techniques for block diagrams could have significant impact. The Clawz system of Arthan et al [9] is a first step: it translates discrete-time models, described using Simulink, into formal specifications in Z. A controller implementation in an Ada-like programming language, with assertions in the form of the "compliance notation", can then be verified against these Z specifications using the ProofPower mechanised proof assistant. Concurrency is being studied in this framework, using CSP/FDR. Mahony [21] has used similar ideas in his DOVE system.

In a pilot study [22] we developed a Hoare-style logic and a verification condition generator for a restricted class of block diagrams, essentially those with a tree structure. Hoare logics [23] were originally studied by Hoare, Floyd and others to give an axiomatic basis for programming, and continue to be used for a variety of applications, for example Java byte-code verification [24]. As far as we know our work is the first to investigate Hoare-style logics for feedback systems. We attached assertions to nodes in the diagram: the key observation was that phase and gain were compositional, and hence we could reason about them locally, and propagate our reasoning through the diagram to deduce properties of a classical frequency-response analysis. Following Gordon's approach [25] the logic was mechanised in the HOL98 theorem proving system, allowing goal-directed reasoning, machine assistance in the details of the proof, and automatic generation of verification conditions, the logical formulas that must ultimately be proved to justify an assertion in the Hoare logic. The verification conditions themselves are pure predicate logic formulas, that is, they do not involve the constructs of our logic.

These ideas also enabled us to prototype an automated alternative to Nichols plots [26], which replaced numeric plots with symbol manipulation in the computer algebra system Maple and the theorem prover PVS (we could also have used quantifier elimination). This in turn exploited our Maple-PVS system, and Gottlieb's PVS continuity checker [27].

D PROGRAMME AND METHDODOLOGY

PHASE 1 PROLOGUE AND EVALUATION Preliminary study and assessment, control and reasoning

There are a variety of representations of linear control phenomena: continuous, discrete, feedback, state space, transform, block diagram: giving rise to representations as differential equations, complex functions and linear

algebra. Our first task will be to assess these, and the relationships between them, and in particular to understand the kinds of properties and assertions that are relevant in applications, for example temporal properties. We will start with linear continuous SISO and MIMO control, bearing in mind extensions to the non-linear and discrete cases. At this stage, since we are trying to extend existing practical techniques, we envisage using classical mathematical approaches. However their amenability to computation suggests we should assess the alternative models implied by Edalat's [17] work extending domain theory to handle dynamical systems, or Fleuriot's non-standard analysis [28] work.

Our second and closely related task will be to survey the logical background, in particular to establish the relationship between categorical models, especially the various categories with trace and feedback [14, 15], and the mathematics considered in Task 1.

PHASE 2 MAIN THEMES PART I

Task 1 Semantics and assertions Our main task is to digest the material surveyed in Phase 1 and identify a suitable semantical basis and assertion language, initially for the continuous linear case. Our plan is to have a small clear base system, suitable for further experimentation, possibly at the expense of omitting some constructs or assertions. This may involve staying at the dynamical systems level, or conversely abstracting to the transform, block diagram or state space view (at time of writing the most plausible candidate). To handle the applications we have in mind one will expect verification conditions and assertions which involve explicit statements about roots and poles of transfer functions, or behaviour of eigenvalues, depending on whether a complex variable, or matrix representation is used. Traces provide a characterisation of fixed point operators in traditional denotational semantics [15], so if our hunch about traced monoidal categories is right, a modal mu-calculus may be the right way to express the assertions.

Task 2 Inference systems In our pilot study we used phase and gain as state variables for our Hoare logic: these worked, but it was clear that they were not the best solution. Developing a more generic Hoare logic, and proving its soundness against our chosen semantics, will require us to characterise state variables with the required compositionality properties. Thus we would expect characterisations of compositionality and inference across feedback loops in terms of dynamical systems, and hence of the associated complex variable and linear algebra. Again, if there are nice category theoretic properties these should manifest themselves in good descriptions of compositionality. We also restricted to tree-structured block diagrams only, but Nipkow [24] shows how to reason about jumps, which will enable us to tackle a larger class.

Task 3 Prototypes While this is not an implementation project we hope to produce prototypes based on tools (Clawz, ProofPower, PVS) being developed by our collaborators: in particular of our inference rules and the necessary verification condition generator. If time we will also investigate analogues of our Nichols plot support: handy tools that exploit our work to provide automated checks. Even something as simple as a parameterised Nichols plot is likely to be useful in practice.

PHASE 3 MAIN THEMES PART II Our third phase will depend on earlier progress, though we will bear these extensions in mind from the start. **Extensions: discrete time and non-linearity** As we have seen the behaviours of continuous and discrete time systems can differ markedly: likewise linearization can mask complex behaviour. When passing between two models in this way it is important to be able to document the properties of each through assertions, and have inference rules for moving between them: we will need to investigate what kinds of consistency conditions we would expect. It may be that we should base our work on existing theories for discrete systems, such the theory of timed automata [29]. While the main thrust of our work above is for linear systems it may be that the assertion language, compositionality conditions and reasoning framework carry over: so we hope that later in the project we will be able to draw a road map for non-linearity research along these lines.

PHASE 4 EVALUATION

In this phase we hope to produce case studies of simple practical examples in avionics and bioinformatics.

E RELEVANCE TO BENEFICIARIES

The immediate beneficiaries of our work are other researchers both academic and industrial, in particular our collaborators at Qinetiq, NASA and SRI International, who will have a secure foundation on which to build

practical experiments in the use of formal methods for the kinds of tasks we have described above. Longer term our work will benefit computer scientists, engineers, biologists and others seeking to understand and apply control engineering techniques.

F DISSEMINATION AND EXPLOITATION

Results will be disseminated through the usual academic channels of journal publication, conference and workshop presentations and by making relevant material available on the Internet (subject to the usual copyright provisos). Experimental software will be distributed to other academic sites without charge, in so far as this can be done in accordance with the rights of our collaborators, and of EPSRC and the University of St Andrews. We will address IPR issues early, set up formal collaboration agreements as necessary (we already have these in place with Qinetiq and SRI for other projects), and undertake regular reviews of technology transfer, IP and exploitation involving EPSRC and the St Andrews IPR team.

G JUSTIFICATION OF RESOURCES

We seek one research assistant: while we are not in a position to name an RA in this proposal we have had informal contact with one or two individuals from Europe who have a strong background in the relevant mathematics, and are interested in working on a project of clear commercial and computer science interest, while keeping good links with the mathematical community. We note in this regard that our group has an excellent track record in recruiting mathematicians and turning them into computer scientists. A project funded **PhD student** will support the project. We expect the student to concentrate on the category theoretic aspects, supervised by Dr R Dyckhoff. We anticipate that investigating categorical representations of control as above, identifying the most appropriate, and developing suitable inference rules and links with other models of control will form a stand-alone project, of value and interest independent of the rest of the project. The **equipment** is costed as high-end LINUX PCs (quote from Gateway), exact purchase to be decided according to market and needs at the time. The amount allocated to **consumables** covers the purchase of necessary software: WMI Maple, Mathworks MATLAB and further components as necessary, necessary incidental expenses: stationery; production and distribution of reports; specialist publications; inter-library loan fees; recurrent computing costs inc. paper and printer supplies; network and infrastructure support. We seek **travel funds** for organising meetings with our project partners, conferences and workshops to include a selection from: CADE 2003/4/5, ISSAC 2003/4/5, TPHOLS 2003/4/5, CAV 2003/4/5, FME 2003/4/5, specialist workshops and meetings as need arises and for research visits to collaborators and industrial sites named in the proposal and to researchers at Cambridge [Gordon], Newcastle [Jones] in the UK, U Penn [Alur], CMU [Clarke, Krogh] SRI [Shankar, Rushby] in the USA and Munich [Nipkow] in Germany: to be combined where possible.

H PROJECT PLAN AND PROGRAM OF WORK

The research team will comprise the project leaders Ursula Martin and Roy Dyckhoff, the project research assistant and PhD student (RS1), with support from a further PhD student (Ruth Hardy) (RS2) funded by our EPSRC DTA 2001-2004. We will work in close collaboration with our listed collaborators at St Andrews and elsewhere. The project will be organised in three phases, each with a list of tasks corresponding to a technical report or research paper deliverable, accompanied where appropriate by demonstrator implementations.

PHASE 1 PROLOGUE AND EVALUATION

Task 1	Preliminary study and assessment, control	Months 1-6	All
Task 2	Preliminary study and assessment, reasoning	Months 1-6	All

PHASE 2 MAIN THEMES PART I

Task 1	Semantics and assertions	Months 7-30	All, Research assistant focus
Task 2	Inference systems	Months 7-30	All, Research student focus RS1
Task 3	Prototypes	Months 7-18	All, Research student focus RS2

PHASE 3 MAIN THEMES PART II

Task 1	Extensions: discrete time	Months 19-30	All, Research assistant focus
--------	---------------------------	--------------	-------------------------------

Task 2	Extensions: non-linearity	Months 19-30	All, Research assistant focus
PHASE 4		EVALUATION	
Task 1	Review of case study	Months 31-36	All
Task 2	Recommendations and final evaluation	Months 31-36	All

MANAGEMENT The project will be managed by Ursula Martin. Regular meetings of the local research team and others as appropriate are anticipated, working with them more intensively as needs dictate, and liaising with our academic and industrial partners. Assessments of progress will take place at **six-monthly project milestones**. Research assistants and students are subject to University of St Andrews policies on appraisal, performance monitoring and review, and submit monthly progress reports. Meetings will be minuted and action lists and follow ups maintained on a project www page.

REFERENCES

- | | | | |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | 16 | C Gurr and K Tourlas, Towards the principled design of software engineering diagrams, in ICSE 22, pp 509-520, ACM Press 2000 |
| 1 | Ursula Martin, Computers, reasoning and mathematical practice, Proc 1997 NATO ASI Summer School, Springer Verlag 1999 | 17 | A Edalat and A Lieutier, Domain theory and differential calculus, Proc IEEE LICS 17, IEEE Press 2002 |
| 2 | A Adams, M Dunstan, H Gottliebsen, T Kelsey, U Martin and S Owre, Computer Algebra meets Automated Reasoning: Integrating Maple and PVS. in TPHOLS 2001, LNCSxxxx, Springer Verlag, 2001 | 18 | A Tiwari and G Khanna, Series of abstractions for hybrid automata, in 5 th HSCC 2002, LNCS 2289, Springer 2002 |
| 3 | Andrew Adams et al, VSDITLU: a verified symbolic definite integral table look-up, CADE 16, LNAI 1632, Springer 1999 | 19 | Martin Dunstan et al, Lightweight formal methods for computer algebra systems, ISSAC'98, ACM Press, 1998 |
| 4 | Ursula Martin, Towards formal methods for mathematical modeling, Proc 5th NASA Formal Methods Workshop, NASA Press 2000 http://shemesh.larc.nasa.gov/fm/Lfm2000 | 20 | Martin Dunstan et al, Formal Methods for Extensions to CAS, FM'99, LNCS 1709, Springer 1999 |
| 5 | K Ogata, Modern control engineering, Prentice Hall 1997 | 21 | B Mahony, The DOVE approach to the design of complex dynamic processes, NASA/CP-2002-211736, August 2002 |
| 6 | E Sontag, Mathematical control theory, Springer 1998 | 22 | R Boulton, R Hardy and U Martin, A Hoare logic for single-input single-output continuous-time control systems, submitted to HSCC 2003, www-theory.st-and.ac.uk/~um/HSCC |
| 7 | MATLAB, www.mathworks.com | 23 | C A R Hoare, An axiomatic basis for computer programming, Communications of the ACM, 12, 576--580 1969 |
| 8 | R Pratt (ed), Flight control systems, IEE Press 2000 | 24 | Tobias Nipkow, Hoare Logics in Isabelle/HOL, in Proof and System-Reliability, Kluwer, 2002 |
| 9 | R Arthan, C O'Halloran et al, ClawZ: Control Laws in Z, ICFEM 2000, IEEE Press 2000 | 25 | M Gordon, Mechanizing programming logics in higher order logic, in Current trends in hardware verification and automated theorem proving, pp 387--439, Springer 1989. |
| 10 | E Davidson, Genomic regulatory systems, Academic Press 2001 | 26 | R Hardy, Symbolic analysis of design requirements for control laws, Technical report 2002, www-theory.st-and.ac.uk/~um/HSCC |
| 11 | P Saunders, J Koeslag and J Wessels, Integral Rein Control in Physiology , J. Theor. Biol. 194 (1998) 163-173. | 27 | H Gottliebsen, Transcendental functions and continuity checking in PVS, TPHOLS 2000, LNCS 1869, Springer 2000 |
| 12 | Bruce Krogh, Approximating Hybrid System Dynamics for Analysis and Control, HSCC 1999, LNCS 1569, Springer, 1999 | 28 | J Fleuriot, On the mechanisation of real analysis in Isabelle/HOL, TPHOLS 2000, LNCS 1869, Springer 2000 |
| 13 | M Jirstrand, Nonlinear Control System Design by Quantifier Elimination, J Symbolic Comput, 24, 137-152, 1997. | | |
| 14 | M Arbib and E Manes, Machines in a category, SIAM review 57 (1974), 163-192 | | |
| 15 | M. Hasegawa, Models of Sharing Graphs, Springer 1997 | | |

- 29 D Dill. A theory of timed automata. Theoretical
Computer Science , 126(2):183-235,1994

APPENDIX 1 PROJECT PLAN

	1-6	7-12	13-18	19-24	25-30	31-36
P1, T1						
P1, T2						
P2, T1						
P2, T2						
P2, T3						
P3, T1						
P3, T2						
P4, T1						

APPENDIX II TRACK RECORD, EXPERTISE AND COLLABORATION

This proposal is part of a long-term programme in the effective use of computational logic techniques in pure and applied mathematics led by the proposer, and extends and complements recent work on computational logic to support research in pure mathematics and symbolic computation. It builds on a long-standing relationship with developers and users of computational mathematics systems of all kinds.

Professor Ursula Martin, the project leader, has a strong background in symbolic computation and computational logic, has published widely on both theoretical and practical matters, has held a series of research grants from EPSRC, EEC, NAG Ltd, Qinetiq and Microsoft in the application of computational logic to pure and applied mathematics. In 1999-2000 she held an International Fellowship with the PVS group at SRI in Menlo Park, funded by an industry fellowship from the Royal Academy of Engineering, to develop the pilot work on this project [5]. She serves on numerous international program committees, for example LICS 03, CADE 99,00,02 and ISSAC 00.

Dr Roy Dyckhoff has a strong background in logic for computer science and also in category theory, classical topology and mathematics, and an international reputation for his recent work on proof search. He has been program chair or co-chair of several international meetings, most recently Tableaux 2000 [28], and a member of a number of international program committees, for example IJCAR 2001 and TAB 02. He works in particular on the development of proof systems (in the form of sequent calculi or analytic tableau systems) for non-classical logics, of interest in computer science or artificial intelligence [26,27].

Collaborators

Dr Martin Hyland, Cambridge Logician working in category theory and

Professor James Davenport, Bath, a leading expert in computer algebra and applications in engineering

Dr Colin O'Halloran, Dr John Hall, Dr Yoge Patel Qinetiq Ltd (formerly DERA) control engineers supporting certification and verification activities via the ADA compliance notation

Professor Peter Saunders, Kings London Mathematician working on control in biological systems

Professor David Gilbert, Glasgow, Computer scientist working on control in Bioinformatics

Dr Ricky Butler, NASA Langley VA USA are active in this area and we enjoy excellent informal working relationships through their recent recruitment of Gottlieb, a former PhD student of Ursula Martin.

Dr John Rushby, Dr Ashish Tiwari, SRI International CA USA develop the PVS system and run a DARPA Mobies project applying it to MATLAB

Research environment The School of Computer Science at St Andrews was one of only three 5*/5A departments in the 2001 RAE, and supports leading research groups in Theoretical and Experimental computer

science. The group led by Ursula Martin has developed over the past 10 years and now has a strong international reputation for our work on foundations, our computational discrete mathematics system GAP, our pioneering work in using techniques from computational logic to extend the power of research and commercial computational math systems, and our work in the OpenMath consortium. Over the past five years we were supported by around £1 million in announced grant income from EPSRC, EPSRC ROPA, EC, industry and other sources. The methods and technologies that we have been developing are all key to this project, particularly our emphasis on the hidden use of computational logic in support of heterogeneous mathematical applications.

An early theme was the evaluation of the use of computational logic techniques to support pure mathematics. Martin's influential survey paper [1] drew attention to the many methodological and social aspects of mathematical research that we needed to support if our work was to have significant impact, and drew a road map for future research which has had major influence on research projects such as EC Calculemus.

The unique and novel thrust of our work since 1996 has been the application of computational logic to scientific and mathematical computation. This has been exploited in widely used commercial systems such as Maple (WMI, Canada, 1 million users) and the NAG library (NAG Ltd, Oxford, 0.5million users), and in our own open source research system GAP (1000 user sites). Currently we are focussing on mathematical challenges in high assurance applications such as avionics with collaborators at SRI International (USA), NASA (USA), WMI (Canada) Qinetiq (UK) and NAG Ltd (UK). Our key advance is providing computational support to increase the dependability and scope of computational mathematics, and assist in the design and certification process: the road map developed at sabbatical at SRI and set out in [4] has led to new work by ourselves and our collaborators in computational support for the MATLAB Mathworks toolset for applications in avionics and automotive engineering.

An earlier project assessed the needs of practitioners and identified abstractions for embedding computational logic in COTS environments, shielding the user from its complexities with automated search and specialist libraries. Examples include a novel verifiable symbolic look-up [3] and assertions to support mathematical modelling [19]. Our prototype Maple-PVS [3] system augments Maple with calls to the computational logic engine PVS for symbolic computations where highest degree of assurance is required. With funding from NAG Ltd we introduced similar techniques to extend Aldor, a NAG internal development language, with "smart comments" that serve as interface specifications for trusted components [20], and may be used, for example, for method selection and compiler optimisation.

RECENT EPSRC GRANTS AND OUTCOMES

2001-2004 EPSRC, Symmetry and constraints, [PI U Martin, S Linton, I Gent, RA T Kelsey] Results so far have included use of the GAP system to speed-up constraint solving problems with symmetry

1999-01 EPSRC ROPA with NAG Ltd, GR/M98340 Evaluation, Design and Validation of Exact Real Computation in a Problem Solving Environment, £84,182 [PI U Martin, S Linton, RA T Kelsey] Results so far have used pre-processing via symbolic computation in Maple to speed up exact real arithmetic.

1998-01 EPSRC JREI GR/M32351 Distributed Software Systems £74,826 [PI R Morrison, U Martin, S Linton et al] Beowulf cluster for experiments including distributed symbolic computing

1997-01 EPSRC GR/L48256, Embedded verification techniques for computer algebra systems, £215,000 [PI U Martin] Verified look-up tables, automatic continuity checking and Maple-PVS calls for DEs

1993-7 UK SERC/SBCC GR/J31230 Algebraic applications of automated reasoning techniques £99,593 [PI U Martin, RA S Linton] We obtained a variety of theoretical and practical results in pure mathematics

RECENT INDUSTRIAL GRANTS

2002-2005 Microsoft Research, Cambridge: £54,000 PhD studentship

2001-2002 Qinetiq, £30,000 [PI St Andrews: U Martin, R Boulton] Formal methods for control engineering: results form the basis of this proposal

1999-00 Royal Academy of Engineering industry Fellowship, Safety of Scientific Systems, £30,000 [PI U Martin] Support for pilot study in the application of formal methods

1998-01 NAG Ltd Advanced compilation and analysis for AXIOM/Aldor £84,000 [PI U Martin, S Linton, RA M Dunstan] Incorporating annotation and static analysis based on CL into NAG's AXIOM/Aldor compiler

RECENT EEC AND OTHER GRANTS

1997-00 EC Esprit Multimedia Project 24.969 OpenMath: accessing and using mathematical information Electronically. ECU 1,600,000 in total, St Andrews ECU120,000 [PI St Andrews S Linton, U Martin] NAG Ltd plus academic and industrial partners to develop interface language for web-based mathematical communication

1992-8 EEC Esprit WG 7232 GENTZEN: Extensions of Logic Programming (180,000 ecu, St Andrews 74,000 ecu) St Andrews [PI R Dyckhoff, U Martin] Tübingen, Gothenburg and Chalmers University of Technology. Extended logic programming using proof theory, inductive definitions and non-standard logic techniques.