FACS FACTS

BCS FACS

The Newsletter of the BCS Formal Aspects of Computing Science Special Interest Group.

Series III Vol. 1, No. 1 Winter 1992

Contents

Editorial

Welcome to this new series of FACS newsletters!

The new series should now appear regularly at quarterly intervals and have a more regular structure, in general. As an example, we are starting a correspondence column with this issue. This will hopefully consist of technical questions and queries, answers and replies, and of course just comments of a general nature. As a further example, the column might be useful for giving informal advance warning of events or in announcing invitations for participation. Finally, there is always a place for humorous writings, (legal) gossip, etc. Our hope is that our correspondence column will encourage further informal contact between academic and industrial interests.

We intend to include a number of technical columns, the first of which are given by the RAISE column (by Chris George) and the HOL column (by Roger Jones). Other possibilities are: *Z, VDM, Term Rewriting, Requirements Analysis, Refinement, Theorem Proving, Formal Methods in Education, etc.*. Please write in and let us know your views.

In addition to the technical columns, we are always interested in short technical articles for publication. These may contain short overviews of technical areas, announcement of results, or contain general reports of activity performed within your group (e.g. site reports from academic and industrial groups). We shall be actively encouraging the production of articles for future issues. If you are interested in writing a short article on some topic for us, please let us know.

Also we plan to allow sponsorship of the newsletter. Sponsors can pay for a mail-out of an issue of the newsletter. The cost depends on the size of the issue and the size of the membership, but is typically of the order of £150. In return the front cover will bear the words: "This issue has been sponsored by ...". If you wish to sponsor an issue please write to or e-mail the editor (see below for details).

Tim Denvir

The newsletter ...

The production deadlines for the coming year will be:

Spring	end of Feburary, 1993
Summer	end of May, 1993
Autumn	end of August, 1993
Winter	end of November, 1993

If you are interested in writing for the newsletter, please see the contributors guidelines.

The views and opinions expressed within articles included in the BCS FACS FACTS newsletter are the responsibility of the authors concerned and do not necessarily represent the opinions or views of the editorial panel.

Acknowledgements

This edition of the newsletter was produced by:

Brian Monahan, University of Manchester
Jawed Siddiqi, Sheffield Hallam University.
Chris Roast, Sheffield Hallam University.

Duplication and distribution by Department of Computing and Management Sciences, Sheffield Hallam University.

Contributions, advice and able assistance were thankfully received from:

David Blyth, John Cooke, Tim Denvir, Chris George, Rajeev Gore, Richard Harter, Roger Jones, Ian Maung, Richard Mitchell, F.X. Reid, Dan Simpson and Ann Wrightson

BCS FACS Annual General Meeting 1992

Minutes of AGM held at the University of Westminster
17 July 1992

Chairman's and Secretary's Report

Membership had dwindled slightly over the year.

FACS members' subscriptions to EATCS will have to increase next year, since it had been discovered that our members were being undercharged.

In the last year two successful workshops had been held, the Formal Aspects of Measurement and the Fifth Refinement Workshop. The proceedings of both are with the publisher (Springer) and should be out imminently.

The Formal Aspects of Computing Journal has been very successful in that a large number of good papers have been received. This has led to an expansion to six issues per year. Consequent disadvantages are a slower turn-around for publication of papers received and an increased cost of subscription. The editorial board is trying to work out a way of pruning the intake so as to shift the emphasis slightly away from the theoretical towards the formal methodological. The subscription list is about 500 which the publishers are happy with. The five year contract between BCS and Springer runs out at the end of the year. This can be renewed year by year if we want. Springer is willing to revise the contract so as to suit FACS.

A full financial report was not available owing to the absence of the treasurer but FACS has money in its bank account and a greater quantity of reserves in a building society. The committee have been considering setting up a separate company to run and publicise meetings.

Election of Officers

Various nominations were received and the new committee were duely elected (see end cover). In general the committee would try to arrange sharing of tasks to ensure coverage (e.g: Newsletter, Publicity, Meeting organisation).

AOB

It was noted that the special issue of the *Computing Bulletin* on formal methods would be deferred because we were unsure of what the new format and policy of the Bulletin would be.

Nineteen members attended the AGM. It concluded with a presentation from Martin Loomes on "Formal Aspects of Chatting about Systems", and a lively discussion followed. Warm thanks to Martin Loomes were expressed.

Tim Denvir, July 1992

The Society Column

"Aha!", you think, "Scandal, glitz and gloss". Alas, no. Not even a manual of glitches in DOS. Just BCS News.

Publications

The big story in this issue is the new publishing deal with Oxford University Press. The BCS and OUP have set up a joint venture company called Itext Ltd. Most of the details were sent out in an insert in *The Computer Bulletin.*

Itext will focus initially on producing the *Computer Journal* and the *Computer Bulletin.* Through Itext, the BCS has gained access to OUP's substantial experience in academic publication and this creates exciting new opportunities.

Whereas in the past, BCS publishing projects were handled via BISL, in future Itext will deal with such projects directly. This means that authors can deal directly with a company whose sole focus is the publishing business. The new joint venture will be well resourced. Proposals put to Itext can be swiftly passed to the appropriate part of OUP for editorial assessment. Once agreements have been reached, authors can deal with OUP through Itext. This should be more efficient than the previous publication route via BISL (No disrespect to BISL staff - they were always overloaded).

If you have a proposal for a new publishing venture, contact Peter Ashby at Itext, care of the BCS.

Formal Methods in Standards

The BCS Formal Methods in Standards Working Party (FMIS) still exists but is presently dormant. As acting chairman, I recently asked Andrew Wilkes to canvas FMIS members on a possible future programme, but what do readers think?

The role of FMIS is to advise the Technical Board so that BCS representatives on BSI committees speak with a coherent voice on FM issues. To this end FMIS prepared a briefing monograph, *Formal Methods in Standards*, which was published by Springer in 1990. There has been no pressing need for meetings since then, but if any reader thinks there is something useful that FMIS could do, please contact either myself or Andrew Wilkes at the BCS.

Branch Funding

BCS members will no doubt have seen the calling notice for the BCS AGM. The branches put up a motion to give them 10% of the Society's budget. Council advised members to vote against it as they considered a percentage formula too inflexible, and the motion was defeated. (Even so, if the SG's had been offered the same deal then the proposal may well have been surported by the membership.) Just to put things in perspective, have a look at the published accounts and compare Branch and SG funding for 1991/92. Poor branches! If you have any strong views on SG funding, please send them to *The Society Column.*

This is *your* forum for all matters concerning FACS and the BCS.

David Blyth, Incord Ltd., Ferndown

Addresses

BCS	The Society Column
PO Box 1454	BCS-FACS FACTS
Station Road	15, Sherwood Avenue
SWINDON	FERNDOWN
Wiltshire	Dorset
SN1 1TG	BH22 8JS
Tel: 0793-480269	0202-896834
FAX: 0793-480270	0202-894834

Report of FACS evening meeting on 17 November 1992 held at Lloyd's Register, 71 Fenchurch Street, London EC1

VDM through Pictures

Jeremy Dick and Jerome Loubersac
Bull Research

(presented by Jeremy Dick)

This was a successful meeting; holding an evening meeting was something FACS has not done for a considerable time, and so resurrecting the practice was an experiment. 16 people attended from a considerable geographical spread.

The work presented was done in the ESPRIT project ATMOSPHERE. Jeremy went through the motivation for formal methods rapidly in view of the audience. The motivation for describing VDM through pictures was for communication with the customer who does not usually understand formal notations and is unwilling to learn; thus to overcome the barriers to technology transfer.

Jeremy used a specification of a lift as an example. Data types are expressed diagrammatically using a notation (TSD's) similar to E-R diagrams. However those are not sufficient: it is desired not to lose the formality of VDM. E-R diagrams are therefore extended to provide a complete visual syntax for VDM types. One of the several limited forms of modular structre in VDM is also supported.

For the operation definitions, classical state transition diagrams are used as the starting point. States become conditions. Conditions are non-disjoint: several may hold simultaneously; mutually disjoint conditions are grouped. The operation specification diagrams are called OSDs.

The result is that a VDM specification can be generated from the two diagrams. Certain aspects of specifications cannot be represented in either diagram, but if a particular style of specification is adopted, it is easy to produce them diagrammatically. Because every diagram transforms to a unique specification, the diagrams have a defined semantics.

Prototype tools have been developed: a picture editor for creating and editing TSDs and OSDs, a parser, a typechecker and a printer. They are currently working on an analysis/design assistant. The tools are integrated with PCTE. The prototype tools were created by systematic translation from their VDM specifications into C++. Integration into PCTE did not take long once he had got familiar with PCTE.

Jeremy saw the weaknesses of the approach as: VDM module structure is unwieldy for an ADT style (but this may improve when further work has been done on proposals for VDM modules - TD), operation frames are always maximum and there becomes a proliferation of small function definitions. Work considered for the future included assessing the effect of the VDM style imposed by the visual notations, considering other types of diagram for VDM, considering diagrams for other approaches, and carrying out case studies.

We are grateful to Jeremy Dick for the time and care which he clearly put into giving this interesting talk.

Tim Denvir

$\phi[\![x]\!]$.Reid
TCS Accessories
221B Hope Park Square,
Basingstoke

Dear Sir,

As you are no doubt well-aware, I have ceased to work in Theoretical Computer Science, because since it has become infested with -

Category Theorists
Topologists
Intuitionalists
Interleaving-Semanticists

- and other such riffraff, the subject is no longer suitable for a man of my dignity and gravitas.

I have therefore incorporated myself into:

F. X. REID: TCS ACCESSORIES PLC.

Our function is to supply items suitable to be put on sale at conferences, workshops, seminars and so on at which TCS purports to be under scrutiny. Your aim is to exploit the circumstances to make a lot of money.

Our first production has been a line of T-shirts with humorously suggestive slogans. For example:

DENOTATIONAL SEMANTICISTS DO IT CONTINUOUSLY

SOFTWARE ENGINEERS DO IT BOTTOM-UP

COMPILER WRITERS DO IT INCREMENTALLY

and so forth. Incidentally, these slogans, were contrived by my collaborator, ex-Professor F. X. Lurk, who assures me that they are amusing. I shall have to take his word for it.

Ex-Professor F. X. Lurk and I are also in the middle of developing what we have provisionally named our 'suddenly !!!' line of products. Our current project is a 'kit' which we intend to market under the title:

Suddenly, you're a Category Theorist !!!

The 'kit' contains an Introduction to Automata Theory[1], and a glossary of technical terms both authentic and of our own manufacture. Using this 'kit', it is possible for practically anyone to translate elementary propositions in automata theory into elaborate-sounding theorems in category theory. We also plan a *de lux* edition which will enable purchasers to generate Ph.D. theses.[2] Other planned titles in this series are:

Suddenly, you're a Formal Methods Expert !!!

Suddenly, you're an Object Oriented Programmer !!![3]

Suddenly, you're an OSI Implementor !!!

Suddenly, you're a Neural Network Trainer !!!

Suddenly, you're a Temporal Logician !!!

We had also projected a title

Suddenly, you're a Software Engineer !!!

but it transpired that as soon as the phrase became popular, every last hacker suddenly *was* a software engineer - a self-styled one, at any rate.

After our initial efforts have met with the success they so clearly

[1] Blackwell Scientific Publications, £19.95 o.n.o.

[2] *pro-formas* will be made available to potential examiners of such theses.

[3] COBOL manual provided.

merit, ex-Professor F. X. Lurk and I propose to found our own press. We have already made tentative arrangements to copywrite the name, **Fexpress**, and are negotiating with authors. Soon to be rolling off the presses will be a distinguished contribution to the art of calligraphy:

Handwriting, the E. W. Dijskrasch Way

soon to be followed by

The F. X. de Wilde-Roeuver[4] Assertiveness Training Course.

A team of Belgian COBOL programmers, writing under the pseudonym F. X. Shields is preparing (or has at least threatened to prepare):

A Bluffer's Guide to Intuitionalistic Type Theory
A Bluffer's Guide to Failure/Divergence Semantics
Fifty New Things to do with Non-standard Set Theory
λ-calculus for Electrical Engineers

and more!

We have other, less well-defined, plans[5] of which we will inform the public as they[6] come to maturity. We would also welcome suggestions of other products.

Thanking your for the hospitality of your organ, I remain

Yours faithfully

Reid

F. X. Reid, BSc, MLitt, DSc, PhD, DPhil, DD, FRS, FD, FRCPS Glasg., FIFA, FET, FM, VLSI, DBE.

[4] (whom God forbid) of Utrecht.

[5] Such as the wall-chart showing Great Theoretical Computer Scientists of the c20 - which only has one name on it at the moment.

[6] The plans, not the public.

WANTED! — Technical Contributions

Ian Maung, University of Brighton

We welcome:

- abstracts of recent technical reports and theses
- articles of general interest
- survey or introductory articles

All FACS technical areas covered - formal specification and verification, concurrency, semantics, abstract data types, applications, etc, etc.

We offer:

- no pedantic red tape about presentation
- rapid and widespread distribution of your ideas

So don't wait months (or years) to get your ideas aired in other broadsheets - submit your latest work now - by post, fax or e-mail to:

Ian Maung
Dept of Computing
University of Brighton
Moulsecoombe
Brighton
East Sussex
BN2 4GJ

Fax: 0273 642405 E-mail: im1@unix.brighton.ac.uk

Any reasonable format accepted for postal contributions, but plain text, LaTeX, PostScript, or Word preferred by E-mail.

ICL *ProofPower*

Roger Jones, ICL Winnersh
November 1992

1 Introduction and brief overview

At the time of writing, the status of research and development is as follows:

- Proof automation for HOL has been further enhanced (see section 2 below). The supplied hierarchy of theories has been extended.
- Syntax and type checking for Z has been supported for some time. Proof support for predicate calculus and set theory in Z is now becoming mature, coverage of all Z language features is nearing completion (see section 3 below).
- The first implementation of support for SAL (SPARK annotation language) is in place, and has been used to prove pre- and post- conditions in SAL against Z specifications.
- First external trials are under way after delivery of a one day tutorial in September.
- A second tutorial in *ProofPower* HOL followed by the first tutorial on *ProofPower* Z are planned for January.

2 *ProofPower* HOL SUPPORT

2.1 Algebraic Normalisation

Automation of term normalisation over arbitrary commutative semigroups and commutative semirings, with specialisation to boolean algebra and natural number arithmetic is now available. This greatly reduces the amount of tedium involved in proving equalities or inequalities in these theories, and provides an essential tool for the implementation of support for linear arithmetic.

2.2 Linear Arithmetic

Automatic proof (amounting to a "decision procedure") for results in linear arithmetic is now provided in *ProofPower*. This will facilitate the introduction of support for other number systems, and will be transferred to these number systems as they become available.

Linear arithmetic in *ProofPower* is implemented entirely in non-critical code, resulting in a logical proof which is checked (behind the scenes) by the *ProofPower* logical kernel. This makes confirmation of the result a little slower than less safe methods but provides immunity to any bugs in the code for linear arithmetic. Where a conjecture is false (or unprovable by these means) it will be rejected rapidly without embarking on the detailed proof construction and checking.

2.3 New Theories

A collection of theories is now provided extending the basic support for SETS in previous versions of *ProofPower* along lines similar to those available for Z in the "Z toolkit". This includes the theory

of binary relations, which are then used to support partial functions. These theories support a style of specification in HOL which is similar to specifications written in Z in an "axiomatic" style. Support for labelled products as analogues for schemas is also available. These features provide a compromise between specification in a less set theoretic HOL style and in full Z which is cost effective for some kinds of application. (They do not form the basis for the support of *ProofPower* Z)

3 *ProofPower* Z Support

The Z supported is an approximation to the evolving Z standard, which is itself a liberalisation and extension of the Z described in Spivey's *The Z Reference Manual.* To facilitate proof further extensions are permitted, though specifications can be checked within the more restricted language.

The extended *ProofPower*-Z language is a higher order language, permitting proof of tautologies using propositional variables. Predicates in this extended language are acceptable as terms of type BOOL, which improves the smoothness of integration with SPARK Annotation Language for refinement to verified code. Mixed language working is supported, allowing a single formal expression to contain parts in distinct languages.

Support for proof in Z is now well advanced. Selection of appropriate proof contexts causes many of the proof facilities (stripping, rewriting, automatic proof) to work with the Z language. These are supplemented where necessary with facilities specific to Z

3.1 Propositional Reasoning

This is inherited from HOL, fully automatic, and well integrated into the normal proof paradigm. This includes automatic proof for "propositional logic with equality".

3.2 Predicate Calculus

Though a little more complex than (first-order) predicate calculus reasoning in HOL, this is now well supported. This includes some automatic handling of quantifiers during stripping (e.g. skolemisation of existential assumptions), forward chaining (similar to HOL88 "RES_TAC"), and automatic predicate calculus proof using resolution.

This is intended to be supplemented by new PROLOG-like backward chaining using unification, targetted at automatic proof of set membership conjectures analogous to type-correctness results.

3.3 Z Toolkit

The Z toolkit is available as a series of four theories entered entirely using standard facilities. These feature include general support for mix-fix notations as needed, for example, for the relational image notation in Z. The user may select or ignore this toolkit by chosing the parents for the theories in which their specification is loaded.

3.4 Set Theory and Relations

Reasoning in elementary set theory and the theory of relations is almost as smooth in *ProofPower*-Z as it is in HOL, with automatic proof of most of the rules in the ZRM.

3.5 Numbers

The theory of integers has been developed in Z to the point at which a sufficient set of results is available for largely manual arithmetic proofs. These results will now permit straightforward implementation of automatic proof for this theory similar to that available for natural numbers in HOL.

3.6 Schema Calculus

A full range of operators in the schema calculus are in the supported language. Derived proof facilities for these constructs are expected to be complete within a few weeks.

4 *ProofPower* SAL Support

In collaboration with Program Validation Limited a mapping of SPARK Annotation Language into HOL has been implemented. A parser and type checker for SAL permit pre and post-conditions expressed in SAL to be accepted by *ProofPower*. In fact this map goes into the image of the Z-HOL mapping, so that SAL quotations can also be viewed expressions or predicates in Z . For this reason a pretty printer for SAL has not been implemented, and SAL pre and post-conditions, when quoted in the body of a Z schema result in intelligible Z schemas. Smooth integration here depends on the higher order features of extended *ProofPower*-Z, permitting boolean expressions in SAL to be identified with predicates in Z.

Small case studies have been conducted showing that with these features pre and post-conditions expressed in SAL can be verified against specifications using schemas in Z. Verification of SPARK programs against these pre and post-conditions may be undertaken using the PVL SPARK examiner and proof tool (and this is being done at PVL).

5 2nd *ProofPower* Trial Package

The first external trials of *ProofPower* are now under way after delivery of a package in September, consisting of a one day hands-on tutorial with a three month trial/evaluation package for *ProofPower* (HOL only).

A second trial package for ProofPower is planned for the beginning of January. This will cover specification and proof both in HOL and in Z, in two consecutive one day courses, which will be held at the University of Reading (UK).

The first course will be a slightly extended version of the previously delivered one day course on specification and proof in *ProofPower*-HOL. This will run from 10:30 a.m. on 6th January through until 12:30 p.m. on the 7th. The main prerequisites for this course are prior acquaintance with first order logic and set theory.

The second course will be a course in the use of *ProofPower* for syntax checking, type checking, and formal reasoning about specifications in Z using *ProofPower*. The prerequisites for this course are prior attendance on the one day course on *ProofPower*-HOL and some prior acquaintance with Z as a specification language.

Each student attending these courses will have their own workstation. Time will be roughly evenly split between lectures, during which the student can illustrate the material by executing the formal parts of the lecture material, and practical sessions, during which the student may work through supplied exercises or otherwise explore the use of the tool.

Either of these is separately bookable, but attendance on the *ProofPower*-HOL course (either the January one or the previous one in September) is a prerequisite for the *ProofPower* Z course.

There are only a few places available (max 10 students, some places already reserved) but we will rerun the courses if necessary to meet demand. The price is £200 +VAT for each course (*ProofPower*-HOL and *ProofPower*-Z counting as separate courses), and a further £150 (+VAT) to cover the costs of issuing a trial/evaluation system. Total for both courses and a trial/evaluation system for *ProofPower* HOL and Z would be £550+VAT for one student, £950+VAT for two students from the same organisation (one set of issue media only).

If you would like further details of these packages, including reservation forms please send a brief message to us:

to: R.B.Jones
International Computers Limited
Eskdale Road,
Winnersh
Wokingham
Berks RG11 5TT
UK

Tel: +44 734 693131 x 6536
Fax: +44 734 697636

Email: R.B.Jones@win0109.uucp
R.B.Jones@win0109.wins.icl.co.uk
R.B.Jones%win0109.uucp@uknet.ac.uk

Priority will be given to people wishing to take a trial/evaluation system. You will need a Sun system to run *ProofPower*.

5.1 *ProofPower* Availability

At present *ProofPower* is available only to people attending courses as part of the trial/evaluation package. The main inhibitor to more general release of *ProofPower* is the quality of documentation available. We hope that the documentation will advance to a point at which we can make *ProofPower* more widely available during 1993.

An announcement on this topic is expected during the first quarter of 1993.

RAISE Column

Chris George, CRI

This is the first RAISE column, and consists mainly of a description of the RAISE tools. Future columns will contain a variety of items that I hope people will find interesting — technical discussions, case studies, industrial experiences, news about further RAISE developments, etc. But the contents will not just be for me to provide. I hope that users of RAISE (actual or potential) will contribute their ideas, opinions, experiences and questions. E-mail them to cwg@csd.cri.dk, or mail them to:

Chris George, CRI A/S, P.O.Box 173, Bregnerødvej 144, DK-3460 Birkerød, DENMARK.

The RAISE Tools

Bent Dandanell
Computer Resources International A/S (CRI),

1 Introduction

RAISE is an acronym for Rigorous Approach to Industrial Software Engineering. The RAISE project was a 120 person year ESPRIT[1] I project running from 1985 to mid 1990. The major results of the RAISE project were: RSL (the RAISE Specification Language), the RAISE Method, extensive documentation and a collection of tools that supports the use of RSL as well as the RAISE method.

The RAISE tools support rigorous and/or formal development of large software systems using RSL and the RAISE method. All of RSL is supported.

Since 1990, effort has succesfully been applied to the tools evolution in order to develop, enhance and industrialise the tools based on user feed-back.

The RAISE method is rigorous rather than formal. This means that the method does not insist on a formal argument of correctness for all components in a system, but enables the formality if desired.

2 The Kernel

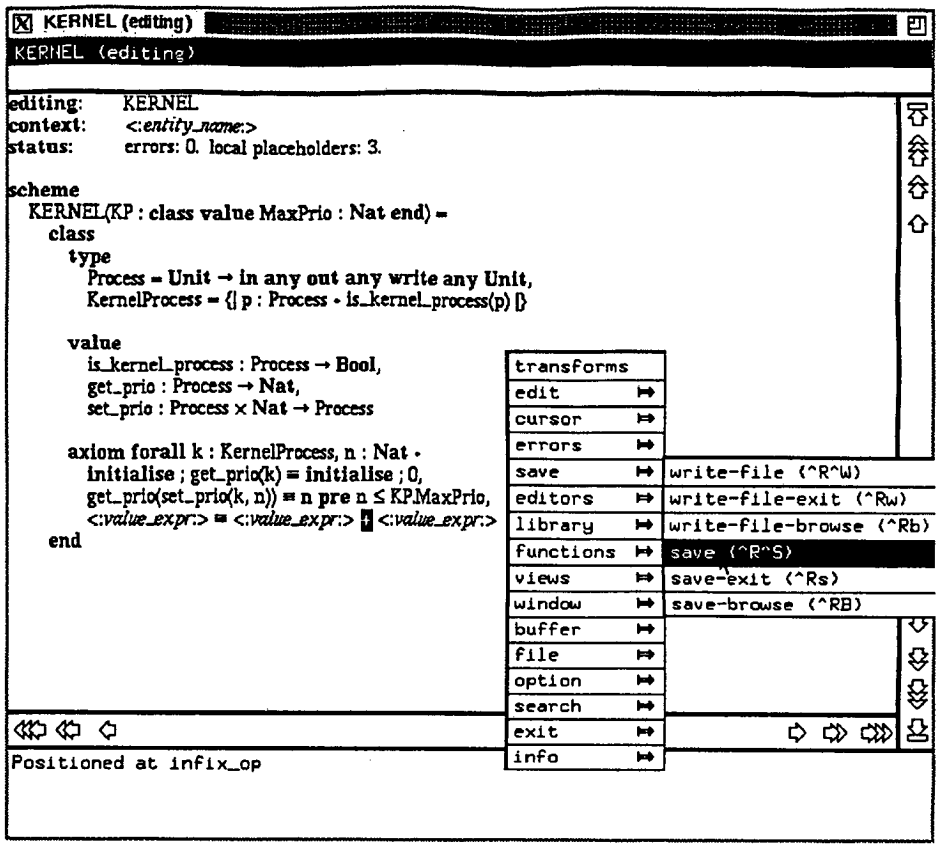
The core tool, 'eden' [2], is a structure-oriented editor supporting writing and continuous type checking of RAISE entities. The editor supports the full symbolic representation of entities including greek letters in identifiers, bolding of keywords, automatic indentation, automatic bracketing, etc. The continuous type-checking provides terse or verbose error messages, structurally linked to erroneous constructs, as well as error-message-only windows and editing windows with error messages suppressed.

The structure-oriented editors can manipulate entities either using pull-down menus, short abbreviated control sequences and/or textual editing. The MMI is homogeneous across the tools. The various tools can be invoked from each other.

The core is encircled by a broad variety of tools which support many different aspects of the software

[1] European Strategic Programme for Research and Development in Information Technology.
[2] *edit entity*

lifecycle and operations on the entities. These tools range from Unix-like commands to interactive editors.



Screen 1: Eden

Entities may be RSL modules, development relations, theories, hints and justifications. In addition to structure-oriented editing, which closely matches the documented syntax for RSL, the editors also support textual editing, as well as the ability to load files prepared using other tools.

RSL modules are either objects or schemes, as defined in RSL. A module may refer to other modules stored in the repository.

3 Library

Oracle as well as the Unix file system can be used as a repository for RAISE entities. The RAISE tools are open ie. they can easily be incorporated into a given development environment.

The ORACLE based repository holds a user defined number of libraries in which entities may be stored under version control and locked against concurrent change. Entities may be referenced across libraries.

The Unix repository enables the usage of standard Unix commands such RCS, make etc. RAISE

entities are stored in ASCII files together with the necessary context information (ie. references to other entities).

An ORACLE and a Unix repository can easily co-exist, ie. an entity stored in the Unix repository may refer to other entities of which some are stored in the ORACLE repository others in the Unix file system.

4 Development Relations and Theories

Development relations are used to express a formal relationship between two modules. The relationship is expressed as an RSL predicate.

The development relation is transitive. This ensures that a detailed specification, closing a chain of developments, conforms to the initial specification. The static properties of the development relation are checked by the development relation editor.

Theories are used to express properties of modules. These properties may be derived properties, which may be used in justifications, or expected properties, which should then be justified.

5 Justification Tools

The construction and writing of formal and rigorous arguments for correctness of a specification is supported by the justification editor. The justification editor also supports the generation and justification of proof opportunities derived from eg. the formal implementation relation.

A justification may be completely informal, or may be a formal proof, or may be a mixture. A justification is developed interactively by application of RSL proof rules and axioms which are in scope. Justifications have a precise syntax and semantics. The justification editor includes a simplifier which automatically makes inferences, and an implementation condition expander which may be used to break a stipulation of the implementation relation into simpler goals. Both forwards and backwards reasoning is possible.

The justification editor keeps track of the number of yet unproved theorems and the number of informal arguments used. Lemmas may be introduced, proved and used as part of a justification.

Only the rules that are applicable to a selected (sub-)expression are displayed. The rules used in the justification editor and the rules listed in the documentation are derived from the same source. This ensures consistency between the tools and the documentation.

6 Pretty Printers

Pretty printers are used to generate LaTeX source for an entity. Such source may then be included in documents together with text, figures or any other related material. The resulting print-out is consistent with the unparsing displayed in the editors.

```
1.0  context:
 .1
 .2  scheme
 .3    KERNEL(KP : class value MaxPrio : Nat end) =
 .4      class
 .5        type
 .6          Process = Unit → in any out any write any Unit,
 .7          KernelProcess = {| p : Process • is_kernel_process(p) |}
 .8
 .9        value
.10          is_kernel_process : Process → Bool,
.11          get_prio : Process → Nat,
.12          set_prio : Process × Nat → Process
.13
.14        axiom
.15          forall k : KernelProcess, n : Nat •
.16            initialise ; get_prio(k) ≡ initialise ; 0,
.17            get_prio(set_prio(k, n)) ≡ n pre n ≤ KP.MaxPrio
.18      end
```

Cross references to be included in a document are generated on-demand. Defined RAISE entities, eg. type, value, scheme and object names etc. may be referenced by line number or page number.

Documents can automatically be printed with current versions of all the entities referenced in them.

For the purpose of easily including fragments of RSL in LaTeX documents, the program 'rslatex' may be used as a preprocessor on files in which such fragments have been written directly using the ASCII representation of RSL.

7 Translators

Translators producing C++ and Ada exist. These are capable of translating vitually all constructive parts of RSL. The translators generate compilable and not just skeleton source code.

Depending on the abstraction level in the translated RSL specification the resulting generated code is useful either for prototyping purposes or for inclusion in production code.

Informative error messages are generated in case a RAISE entity is untranslatable. The message indicates which parts of the specification need to be developed further if automatic translation is needed.

8 Availability

The RAISE tools are available on SUN 3/60 and SPARC computers using the Unix operating system. The tools require at least 12 Mbyte on-board memory and a free disk area of 40 Mbyte.

X11, Sunwindows and OpenWindow interfaces are supported. Licences for commercial purpose are available. Academic licences are available at a reduced cost.

9 Further Reading

The RAISE Specification Language,
The RAISE Language Group,
BCS Practitioner Series
Prentice-Hall International

This is a combined tutorial and reference manual of the RAISE Specfication Language (RSL). It covers the whole language, and features numerous useful examples.

A.E. Haxthausen, J. Storbank Pedersen and S. Prehn,
RAISE Overview,
Computer Resources International A/S

This document describes the ideas and components of RAISE at an introductory level. In particular, various features of the specification language are illustrated by small examples.

B. Dandanell and C. George,
The LaCoS Project,
Computer Resources International A/S

RAISE is among others used in the ESPRIT LaCoS[3] project. The aim of LaCoS is to, try RAISE on industrial applications, improve the tools and document experiences gained using RAISE. "The LaCoS Project" is a public introduction to the project. It covers all the workpackages defined in the project, including a brief description of the applications which are using RAISE.

"RAISE Overview" and "The LaCoS Project" are available from Computer Resources International A/S. Requests for these documents and additional information on RAISE or LaCoS should be directed to `raise@csd.cri.dk`

[3]Large scale Correct Systems using formal methods

Formalising a CORE Requirements Model: An Experimental Investigation

by

M. R. Moulding and L. C. Smith

Software Engineering Group

Royal Military College of Science (Cranfield)

Shrivenham, Wilts. SN6 8LA

Abstract

CORE is an established requirements modelling method which employs diagrams and natural language supporting text to express the requirements of a system. This paper presents an overview of a three year project which has investigated the use of formal specification techniques to strengthen CORE. The work has focussed on the use of the Vienna Development Method (VDM), and a proposed mapping of VDM onto CORE has been applied to an Air Traffic Control (ATC) application in order to assess the utility of the combined CORE/VDM approach. In addition, the role of Communicating Sequential Processes (CSP) to specify the control behaviour of a CORE requirements model has been investigated. The results of these investigations suggest that VDM complements CORE by improving the semantic definition of the resultant requirements model, thus forging a formal link between requirements expression and system specification. Furthermore, it is suggested that CSP may be used to complement VDM in order to portray the dynamic aspects of a CORE model in a more explicit and visible way.

1 Introduction

In 1985, a two-year project was set up at the Royal Military College of Science (RMCS) to investigate the applicability of software fault tolerance techniques to Air Traffic Control (ATC) systems. The project was funded by the Civil Aviation Authority (CAA) and focussed on the potential use of recovery blocks [Randell, 1975] for the Radar Data Processing functions of the London Air Traffic Control Centre (LATCC) at West Drayton. During the initial stages of this project, it was necessary to formulate a *retrospective* requirements model for LATCC, and the Controlled Requirements Expression (CORE) method [Mullery, 1979] was selected for this purpose. This proved to be very successful in that the graphical notations employed by CORE encouraged staff at LATCC to communicate their understanding of the system, and a clear model was established which described both the ATC functions of LATCC and the way these functions interacted with external systems and users.

At the time of embarking upon the LATCC analysis, two variants of CORE were available: the British Aerospace (BAe) version of CORE supported by a *Workstation* tool running on a DEC VAX with Sigmex terminal, and the System Designers (SD) version supported by their CORE *Analyst* product running on an Apple Macintosh XL. Largely because of the lower tooling cost of

the SD variant, this version was selected for the project.

Although CORE had proved successful for the LATCC analysis, certain difficulties were noted [Moulding & Barrett, 1987]. In particular, the graphical notations employed by CORE were not rigorously defined and, in some cases, required significant interpretation. Furthermore, the detailed processing semantics of the requirements model relied heavily upon natural language supporting text, and little guidance was available within the method to define the content of this. It was concluded that formal specification approaches should be investigated to address these problems. Consequently, in 1988, a 3-year project was undertaken at RMCS, with CAA funding, which was concerned with investigating how the SD-CORE method [SD, 1989] might be strengthened by the complementary use of formal specification techniques. The project was proposed in collaboration with Praxis Systems plc, and the Vienna Development Method (VDM) [Jones, 1990] was identified as the primary focus. It was, however, recognised at this stage that other formalisms should be investigated to complement VDM and, during the project, Communicating Sequential Processes (CSP) [Hoare, 1985] was selected for this purpose.

This paper addresses the work of this latter project. The bulk of this work has been concerned with exploring the way in which VDM could be used to describe formally a CORE requirements model and, in order to validate and refine this work, the requirements analysis for a Short Term Conflict Alert ATC function was performed using the combined CORE/VDM approach. During the early stages of the project, it soon became apparent that a computer-based tool was needed to support the generation of VDM specifications and, after an extensive review, the Adelard SpecBox product [Adelard, 1991], which supports the emerging BSI-VDM Specification Language standard, was selected for a PC-compatible platform. In parallel, the CAA had funded System Designers to enhance their CORE Analyst product for an Apple Macintosh platform and, consequently, these two tools provided the basic support for our project.

The remaining sections of this paper provide a brief outline of the technical basis of the project, summarise the results of the experimental investigations, and present our overall conclusions and planned future work.

2. SD-CORE

The SD variant of CORE comprises seven stages which, for the purpose of this paper, we divide into 3 main phases:

(i) **Problem Definition**. This is the first stage of CORE and is concerned with establishing the customer authority and setting the context for the requirements analysis.
(ii) **Functional Modelling**. This corresponds to SD-CORE stages 2-6 and is concerned with describing the functional aspects of the requirement (i.e. the services which the system must provide).
(iii) **Constraints Analysis**. This is the seventh and last stage of SD-CORE and is associated with establishing the non-functional aspects of the requirement (e.g. reliability, performance issues, etc.).

It is the functional modelling phase of CORE which employs the graphical notations discussed above, and which has been the subject of our investigations into the role of formal specification for CORE.

The functional modelling phase of CORE can be considered logically to result in the establishment of a data-flow model depicting the interaction between the functions of the target system, which is the subject of the requirements analysis, and the various systems and users which form the operational environment of that target system. This is illustrated in Figure 1. The five stages of CORE (2-6) can then be considered progressively to construct and analyse this model in the following way:

- **Viewpoint Structuring.** The main environment entities are identified and, together with the target system, form the *viewpoints* which will be the subject of the analysis exercise.
- **Tabular Collection.** Each viewpoint is analysed separately to define the *actions* (functions) it contains and the data upon which these actions operate. The target system is normally analysed last and, when all viewpoints have been so analysed, a connected data-flow model logically exists, and can be checked for consistency.
- **Data Structuring.** The data flows emerging from each viewpoint are described in terms of their temporal relationships and their detailed content.
- **Single Viewpoint Modelling.** At this stage, the information derived from the Tabular Collection stage for each viewpoint is enhanced to provide additional detail such as the internal data flows between actions of a viewpoint and the control behaviour of actions.
  At the end of this stage, the data-flow model which logically exists at the end of Tabular Collection has been enhanced to reveal the detailed characteristics of the actions and the data they operate upon.
- **Combined Viewpoint Modelling.** Having logically established a detailed data-flow model for the target system with its operational environment, important transactions involving the target system and its environmental viewpoints are identified. The model is then checked to ensure that these are present as data processing threads through the relevant actions of the model's viewpoints.

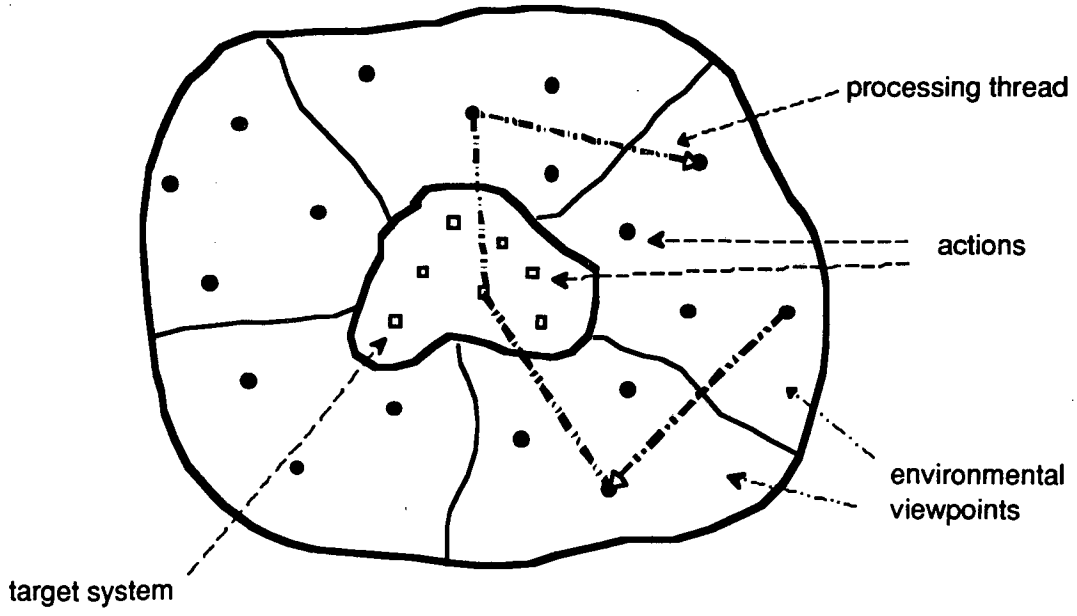


Figure 1 Schematic Overview of CORE

Each of these stages is based upon the use of diagrams to describe the information obtained. However, natural language supporting text is required to add semantics, typically in the form of outline descriptions of the viewpoints, a data and action dictionary, and descriptions of each Single Viewpoint Model and Combined Viewpoint Model.

Necessarily, this description of SD-CORE abstracts away various details of the method and its notations, but there are two complications to this simple picture which are worthy of brief comment. First, it is possible to define *indirect* environmental viewpoints which will not be analysed in terms of their actions; they exist in a model simply as sources and sinks of data flows, and their use significantly weakens the consistency checking which the method supports. Secondly, CORE allows the functional modelling phase to be carried out at various levels of abstraction, defined by a hierarchical viewpoint structure. It does not impose any specific decomposition technique to achieve the abstraction levels, but requires that the processing of a parent viewpoint is a valid abstraction over the collective processing of its sub-viewpoints. The impact of these issues on the formal specification of a CORE model has been addressed by our project, but a discussion of this is beyond the scope of this paper.

3. Vienna Development Method (VDM)

VDM is a member of the *model-based* class of formal specification approaches where a system's state is explicitly represented mathematically, and the functionality of the system is defined in terms of the manipulation of that state. In VDM, system state variables are declared using data types which are derived from the in-built VDM types of scalars, sets, lists, records and mappings, all of which are set-theoretic. The functionality of a system, as perceived by the environment of that system, is represented by a number of VDM *operations* that modify system state variables and are specified in terms of pre- and post-conditions which are expressed in predicate logic. In order to simplify the pre- and post-conditions, operations may refer to (side-effect-free) *functions* which are themselves specified using predicate logic - either implicitly using pre- and post-conditions, or explicitly by expressing a rule to define the result of the function.

In VDM, it is possible to define *invariants* which constrain the values that a data type defines (type invariants), and the values that state variables can hold (state invariants). Proof obligations can then be generated and discharged, to ensure that the specified functions and operations do not violate these invariant properties of the model. VDM also defines proof obligations associated with the progressive refinement of a specification towards an implementation, but these are beyond the scope of this paper.

In addition to the facilities discussed above, it is envisaged that BSI-VDM will introduce a limited *module* scheme for partitioning a large specification. Our understanding [Froome, 1991] is that modules will essentially support the sharing of functions and data types across a large specification, but can also be used to specify abstract data types (rather awkwardly).

BSI-VDM will support both mathematical-symbol and ASCII syntaxes, but in this project we have been constrained, by the SpecBox tool, to use the ASCII syntax.

4. Mapping VDM onto CORE

The final approach adopted for mapping VDM onto a CORE requirements model [Smith, 1992a] is summarised as follows. The data flows of the CORE model form the global state of the VDM specification, and the VDM data types for these state variables are derived from the data structuring information within the CORE model. The target system and other environmental viewpoints are expressed as BSI-VDM modules in which the actions of the viewpoints are modelled as VDM operations. Internal data flows within a viewpoint can be considered as the local state of the viewpoint module, and the triggering conditions for an action are expressed in the pre-condition of the corresponding operation, by testing for the presence of data in the local and global states of the VDM model.

A practical complication to this simple mapping arrangement derives from the limitation of BSI-VDM modules that state variables cannot be shared between modules [Froome, 1991]. This requires that each viewpoint module must declare locally those aspects of the global state which its operations require to access; global data flows must be replicated in the various modules. However, data types can be shared between modules and, consequently, the type definitions for these data flows do not have to be replicated.

It should be noted that the output of the Tabular Collection stage of CORE is subsumed within the Single Viewpoint Modelling stage and, consequently, is not explicitly modelled in VDM. Furthermore, the thread analysis of the Combined Viewpoint Modelling stage of CORE is not modelled in VDM; it is more readily expressed in the CSP modelling of CORE, as discussed in section 7, below.

5. The Short Term Conflict Alert (STCA) Experiment

Following the development of a prototype mapping of VDM onto CORE, using the retrospective analysis of LATCC as a test vehicle, we felt that it was important to apply this mapping to a practical CORE requirements analysis exercise in order to validate the overall approach and to allow the mapping to be revised in the light of practical experience. The STCA function was selected by CAA for this purpose, and CAA staff at LATCC were kindly made available to support the exercise.

The overall purpose of an STCA system is to provide air traffic controllers with early warning of aircraft which are in danger of airborne collision. The system primarily receives input from the radar tracking functions of the associated ATC system and interacts with the controllers via a Human Computer Interface (HCI). Collision alert information is produced in a two-stage process: a course filter identifies aircraft pairs that are in potential conflict; alert confirmation is passed to a controller if any of these pairs pass one of three fine-filter tests.

The way in which the CORE/VDM analysis exercise was conducted is illustrated in Figure 2. It is instructive to note that the natural language supporting text for the normal CORE analysis was insufficiently precise to allow the VDM specification to be constructed. When additional information had been obtained to construct the VDM specification, more comprehensive supporting text could be generated from the VDM, thus providing a means by which this additional information could be validated by the customer.

The final specification produced for the STCA model was some 1600 lines of VDM, distributed across 23 modules. A 400% increase in effort was required for the complete CORE/VDM analysis of STCA, compared with the CORE-only phase of the experiment which was conducted first, and this reflects the additional load of collecting and specifying the extra detail.

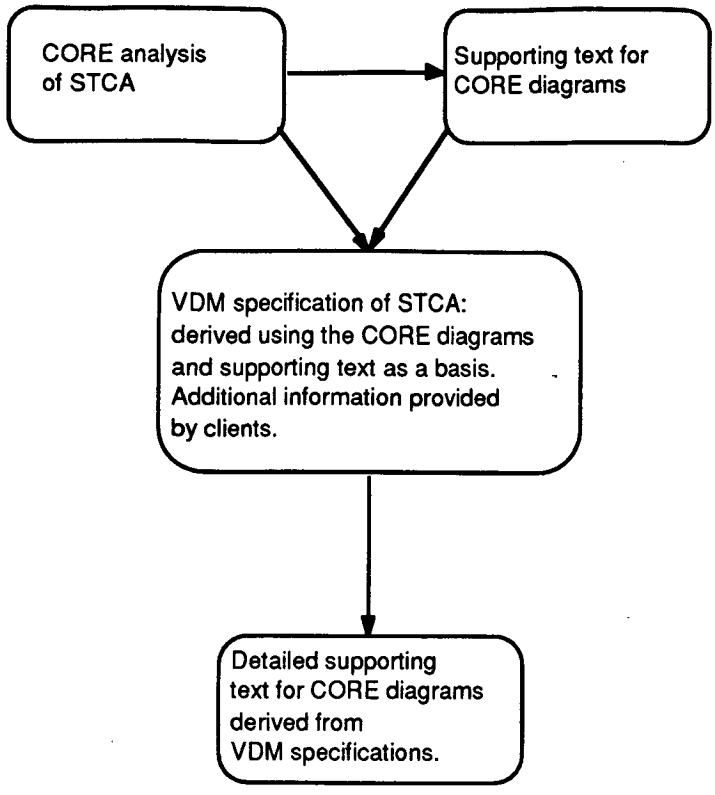


Figure 2 CORE/VDM Approach

6. Discussion of the CORE/VDM approach

Overall, the results of the STCA experiment support the premise upon which the project was instigated, that VDM can provide an effective way of complementing CORE's diagrammatic notations. In using CORE and VDM in this way, we provide a formal link between the requirements modelling and specification phases of a system development. Indeed, one could view the role of CORE, in this combined approach, as providing a method for organising the formal system specification. The viewpoint structuring stage provides a starting point for the development of the VDM specification by guiding the partitioning of the formal specification across a number of modules (and by controlling the abstraction levels of a VDM specification when a hierarchical viewpoint structure is employed). The data structuring stage is used to derive the state of the formal model, and the single viewpoint modelling stage identifies the VDM operations which portray the functionality of the model.

As anticipated, the CORE diagrams proved to be useful in encouraging discussion between the client and the analyst during the process of generating the formal specification, and argues in

favour of the general approach of using diagrams to visualise formal specifications [Dick & Loubersac, 1991].

Although the experiment showed that VDM strengthens the CORE method by forcing the analyst to capture information which was originally missed, it could be argued that this increased detail required by VDM complicates the analysis exercise and may result in poor *information* collection from the problem domain. For this reason, we suggest that the approach we have adopted of first performing a standard CORE analysis, before embarking upon VDM, offers potential benefits. This allows the broad features of the requirement to be discussed in an expressive and informal way, before precisely defining the characteristics of those features.

During the course of the STCA experiment, certain practical difficulties in applying the CORE/VDM approach were experienced. It is clear that, if the approach is to be suitable for large software projects, it will be necessary to introduce tool support to assist in managing the size and complexity of the resulting formal specification. In particular, the need for a tool capable of animating VDM, and of detailed checking of the BSI-VDM module interfaces, has been identified. Of further significant benefit would be a tool which could provide an automatic translation from the CORE diagrams into BSI-VDM, in a similar manner to that proposed by [Dick & Loubersac, 1991].

7. Mapping CSP onto CORE

VDM is essentially a specification approach for sequential systems whereas CORE addresses real-time concurrent systems. In particular, the graphical notations associated with the Single Viewpoint Modelling and Combined Viewpoint Modelling stages of CORE allow both the control and sequencing behaviour of operations to be specified. To some extent, this difference has been reconciled, in the adopted mapping of VDM onto CORE, by capturing the triggering conditions for an action in the pre-condition of that action, as described in section 5, above. Such an approach, however, increases the complexity of the resulting VDM specification, provides poor visibility of the control and sequencing behaviour, and does not readily admit formal analysis of this behaviour. A better approach would be to separate out the control and sequencing information from the VDM specification, and to specify these features in a concurrent system formalism. A CORE requirements model would then be denoted by two formal models: a VDM specification to define the processing semantics of the actions (i.e. to define what individual actions do) and a concurrent system specification to identify the real-time behaviour of those actions (i.e. to define when they execute and how they interact).

Following a brief review of relevant formalisms, CSP was chosen and a pilot investigation into the mapping of CSP onto CORE has been performed [Smith, 1992b]. The results of this work suggest that CSP does provide a suitable approach to complement VDM, with the CSP trace concept mapping conveniently onto the transaction thread of CORE's Combined Viewpoint Modelling stage. Work on integrating the VDM and CSP models of CORE has not been tackled yet, but the key to this would appear to be the CORE data flows which bridge the two models.

8. Conclusions

From the results of this investigation, we conclude that VDM complements the conventional CORE method to provide a requirements model with improved semantic definition of the processing actions. The price paid for this increased semantic precision, in the STCA analysis, was a 400% increase in effort, but this must be weighed against the very high cost of correcting requirements errors which are detected later in the life-cycle [Cohen et al, 1986].

We also conclude that CSP complements the CORE/VDM approach by providing a description of the control and sequencing behaviour of the actions within a CORE requirements model which is both explicit and amenable to formal analysis.

9. Future Work

Work which has not been addressed in this project, but which is to be the subject of future investigations, is as follows:

(a) the role of mathematical proof within both the model-based (VDM) and process-based (CSP) specifications of a CORE model.
(b) refinement to the process-based (CSP) description of CORE and the integration of this with the model-based (VDM) specification.
(c) animation of both model-based and process-based specifications to support requirements validation.

This work is to be addressed by RMCS as part of the DTI-funded *B User Trials* project which also involves Lloyd's Register, Program Validation Ltd and the Rutherford Appleton Laboratory [Shaw, 1991]. The role of VDM will be replaced by the Abstract Machine Notation (AMN) of Abrial [Abrial, 1992] and the new CORE/AMN approach will be investigated for the STCA application using the B Toolkit recently developed by BP International Ltd, and which supports both animation and proof of AMN specifications.

10. Acknowledgements

The authors gratefully acknowledge the contribution of the CAA, without whose funding and support this project would not have been possible. In particular, we would like to express our thanks to Steve Willmott, Andy Price and Paul Cornish. We would also like to extend our thanks to Mel Jackson (Praxis Systems plc) for his guidance, particularly during the early phases of the project, and to Dr Peter Froome (Adelard) for uncovering the proposed features of BSI-VDM modules.

REFERENCES

[Abrial, 1992] J-R Abrial, **Assigning Programs to Meaning**, Prentice Hall International, 1992. (to appear)

[Adelard, 1991] Adelard, **SpecBox User Manual**, Version PC/2.1, Adelard, Coborn House Business Centre, 3 Coborn Rd., London, January 1991.

[Cohen et al, 1986] B. Cohen, W. T. Harwood & M. I. Jackson, **The Specification of Complex Systems**, Addison-Wesley, 1986.

[Dick & Loubersac, 1991] J. Dick and J. Loubersac, **The Visual Presentation of VDM Specifications**, Proc. 4th International Symposium of VDM Europe, Noordwijkerhout, The Netherlands, October 1991.

[Froome, 1991] P. Froome, **Use of BSI-VDM Modules**, Adelard Ref. M/9/0121/1 v1.0, July 1991.

[Hoare, 1985] C. A. R. Hoare, **Communicating Sequential Processes**, Prentice-Hall, 1985.

[Jones, 1990] C. B. Jones, **Systematic Software Development using VDM**, Prentice-Hall, 1990.

[Moulding & Barrett, 1987] M. R. Moulding and P. A. Barrett, **Software Fault Tolerance in Air Traffic Control Systems: Project Final Report**, RMCS Ref. 1049/TD.6, 1987

[Mullery, 1979] G. P. Mullery, **CORE: Method for Controlled Requirements Expression**, Proc. IEEE Fourth International Conference on Software Engineering, New York, 1979.

[Randell, 1975] B. Randell, **System Structuring for Software Fault Tolerance**, IEEE Trans. SE-1 (2), pp. 220-232, 1975.

[SD, 1989] System Designers, **CORE - The Method**, Issue 2.0, SD Software Technology Centre, Camberley, April 1989.

[Shaw, 1991] R. C. Shaw, **B User Trials Proposal**, Submission IED4/1/2182, Lloyd's Ref. BUT/Lloyd's/RCS/1/v8, September 1991.

[Smith, 1992a] L. C. Smith, **A Code of Practice for the Representation of CORE Requirements using BSI-VDM**, RMCS Ref. 1091/TR9, March 1992

[Smith, 1992b] L. C. Smith, **An Initial Investigation into the use of CSP with CORE**, RMCS Ref. 1091/TR10, March 1992.

MATHEGENESIS
The Book of Mathematical Foundations

In the beginning there was Cantor, and he gave us Ordinals and Cardinals and numerable and nondenumerable sets. And the multitude looked on in wonder and marveled, for the vistas opened up were uncountable. But Kronecker cried out, "This is sin and wickedness, God has made the integers, all else is the work of the competition." And some heeded him, but the multitude was beguiled by the infinite pleasures of Cantor's new theory, and they mocked Kronecker, and drove him into the wilderness.

And it came to pass that in those days there was a man named Frege, a most wonderously subtle and wise man. And he said, behold, I have seen a vision. It has been revealed unto me that all mathematics and all reason are but a species of logic. And he built a mighty palace that reached up into the heavens without limit. And within that palace there were no end of wondrous things, including contradictions.

And the Lord said, and I will cast down this palace, for it is not mete that a mortal should thus storm the heavens. And the Lord called unto a man named Russell, and said unto him, "Behold, this man Frege hath displeased me, for he hath made much out of nothing. Thou art my instrument to cast him down. I shall arm you with a dox, nay with a pair of dox, and set you forth". And Russell set forth with his paradox and Frege was cast down.

The multitude saw this and lamented, for they loth to leave garden of paradise of absolute infinities, yet the garden was infested with paradox.

And Russell saw this and felt shamed for he had destroyed much and had built nothing. So he set about to restore the palace of Frege, free of paradox, wherein the Garden of Cantor might be placed. And he labored mightily, and brought forth a new palace. And he guarded against paradox by all manner of walls and barriers, types and orders.

Now there came unto Russell a Spanish Barber, saying, "Thy work is wondrous, though somewhat tedious, but where is this marvelous garden that Cantor gave us." And Russell replied, "Behold, I have added a path within the palace from which all can be reached, for it is of infinite

length." And a great cry went up, for this path was a bit hard to swallow.

Whilst Russell was raising the palace that he had razed, a new prophet arose, saying, "Cantor gave unto a thing of mathematics, and Frege hath made of it a thing of logic. Let us hark unto the vision of Cantor." And he said, "The false prophet, Frege, gave us law. And he said it was good, for law is surely good. But the dox, both of them, hast shown that Frege's law was false law. Now I say that Cantor was a man of truth, and his vision was true. But truth needs law, as law needs truth. So let us lay down law for Cantor's truth." And he did. And his name was Zermelo.

Now Zermelo was a man of great order, yea beyond all measure. But some saw that he was arbitrary in the ways he chose to attain that order, and they rejected his order.

And it came to pass that the spirit of Kronecker whispered unto two men and prevailed on them to take on the quest of restoring the spirit of Mathematics. And each did so, each in his own way, taking separate paths. And their names were Hilbert and Brouwer, and the names of their paths were Formalism and Intuitionism. And they labored mightily, and they traveled far. But in the end, nothing much came of it.

The law of Zermelo took hold in the land, and the people praised it, Theologians formulated it and elaborated it and tested it, and created all manner of axiomatic set theories. And their names were Frankel, and Bernays, and Skolem.

But Skolem was a holy man, and there came unto him a vision. And he saw that the great garden of Cantor was an illusion, a great shadow cast by countable models. And he preached that it was so. And the multitude said, yes it is so, and we will it in our proof theory. But their agreement was with their lips, but not in their hearts. And they said, "None-the-less the continuum is not countable", and they hearkened unto Platonism.

Now the Lord looked down, and he saw that many believed that Man is the measure of all things, and that all problems could be solved. And the Lord grew wroth, for this sort of thing was lese majeste, and He called unto Him a man named Gödel, and He armed him with theorems, and He set Gödel forth to preach.

And Gödel preached unto the multitude, saying "Whatever thou knowest, if it be truly worth knowing, thou canst never be sure of". And he said unto them, "For every truth that thou canst prove, there is another that thou cannot prove." And the multitude was abashed and sore perplexed.

In the wilderness of Harvard there came a man named Quine. And he surveyed the ruins of the palace that Russell had built on the ruins of the palace that Frege had built. And he said, 'I can repair this mess." And he did. And he called his labors, New Foundations, and he cast them unto the multitude. And the multitude said, "Can we do Mathematics in the ways of our Fathers?" And Quine said, "Not exactly." And the multitude asked, "Can we follow your law and the law of Zermelo at the same time?". And Quine replied, "Not completely." And the multitude rejected Quine, and Quine returned to the wilderness of Harvard and wrote books.

The spirit of Kronecker was wroth, for the multitude followed the law of Zermelo and froliced in the garden of Cantor. So he appeared unto a man named Turing, and a man named Church, saying unto them, "The multitude hath donned all manner of fantastic garments, but they can do no more than their forefathers did, for the integers are all there really is." And Turing and Church hearkened unto the spirit of Kronecker and they devised theories of computability. And Church laid down a great Thesis, saying that all theories of computability are equivalent, and that what ye may do is forever limited. And he laid down a universal law prescribing what Man may or may not do. And he shewed that the great law could never be proven.

And thus it came to pass that inconsistency was removed from set theory in many divers and inconsistent ways. And the Lord looked upon this and saw that it was good. And the multitude wrote papers and taught courses and garnered grants, and saw that that was good too.

Translated from the original by: Richard Harter
SMDS Inc., PO Box 55, Concord MA 01742

Typesetting for BCS FACS FACTS by Brian Monahan

The 1992 BCS-FACS Christmas Workshop

Turning a Formal Eye on Requirements

Imperial College, Department of Computing
16-17th December 1992

The workshop aims to survey the current state of R&D in the requirements engineering stage of the development of software-intensive systems; the elicitation of requirements, how they are expressed and how they are analysed.

Included are reports of on-going research; perspectives on industry developments and related problems. It is intended to encompass a variety of different approaches being persued in this problematic area.

Speakers so far confirmed (in alphabetical order):

Tom Anderson and colleagues (Newcastle/York)
The BAe Dependable Computer Systems Centre

Matthew Bickerton, Oxford Centre for Requirements

Albert Camilleri, HP Labs Bristol

Anthony Finkelstein, Imperial College
Methodology in formal requirements capture

Anthony Hall, Praxis Systems plc.
Formal methods in the requirements for an air traffic control project.

Sara Jones, City University
G-MARC

Axel van Lamweerde, Catholic University of Leuven
The ICARUS project, ESPRIT

Tom Maibaum, Imperial College
Formalization of the Engineering Process

Bill Quirk, Harwell
Validation : 3 Aspects of the FOREST project

Forthcoming Events

January 4 - 6 *IEEE Int'l Symp. on Requirements Eng* San Diego, Calif., USA Contact: Stephen F. Fickas, Univ. of Oregon, Comp. and Information Sci. Dept., Eugene, OR. 97403, Tel: (503) 346-3964. E-Mail: fickas@cs.uoregon.edu

January 11 - 13 POPL'93
The Twentieth Annual ACM SIGPLAN-SIGACT Symp. on Principles of Prog. Lang. Charleston, South Carolina, USA Contact: (Program Chair) Susan L. Graham, Comp. Sci. Division - EECS, 571 Evans Hall, Univ. of California, Berkeley, CA 94720, USA, Tel: (510) 642-2059. E-Mail: graham@cs.berkeley.edu *or* (Local Arrangements Chair) Dee Medley, Augusta College, Tel: (404) 737-1672. E-Mail: dmedley@uscn.uga.edu

February 25 - 26 ICSP
Int'l Conf. on Software Process Berlin Sponsored by : Rocky Mountain Inst. of Software Eng Contact: Herbert Weber, Soft-Tech., Univ. of Dortmund, PO BOX 500-500, D-4600, Dortmund 50, Germany, Tel: 49 (231) 775-2780, Fax: 49 (231) 755-2047.

February 25 - 27 STACS '92
10th Symp. on Theoretical Aspects of Comp. Sci. '93 Congress Centrum Wurzburg, Germany Sponsored by : GI, AFCET Contact: Prof. Dr. Klaus W. Wagner, Lehrstuhl fur Theoretische Informatik, Universitat Wurzburg, Am Exerzierplatz 3, 8700 Wurzburg, Germany, Tel: : +49-931-8878 10 E-Mail: stacs@informatik.uni-wuerzburg.de

March 16 - 18 TLCA
Int'l. Conf. on Typed Lambda Calculi and Applications Utrecht, The Netherlands Contact: Mr Frans Snijders, CWI, PO BOX 4079, 1009 AB Amsterdam, The Netherlands, Tel: +31-20-5924171, Fax: +31-20-5924199. E-Mail: franss@cwi.nl

March 21 - 23 *5th Annual Oregon Workshop on Software Metrics* Silver Falls State Conf. Center, Oregon, USA Sponsored by : Oregon Center for Advanced Tech. Education and State Univ. Center for Software Quality Research. Contact: Warren Harrison, Center for Software Quality Research, Portland State Univ., Portland, *or* 97207-0751; Tel: (503) 725-3108. E-Mail: warren@cs.pdx.edu

March 24 - 26 IWSR 93
Second Int'l Workshop on Software Reusability Lucca, Italy Co-sponsored by : ACM SIGSoft et al Contact: Ruben Prieto-Diaz, Software Productivity Consortium, 2214 Rock Hill Rd., Herndon, VA 22070, USA, Tel: (703) 742-7107, Fax: (703) 742-7200.

April 13 - 16 IPPS 93
Seventh Int'l Parallel Processing Symp Newport Beach, California, USA Co-sponsored by : ACM SIGArch Contact: Viktor K. Prasanna, EE Systems Dept., EEB 244, Univ. of Southern California, Los Angeles, CA 90089-2562, Fax: (213) 740-4449. E-Mail: ipps93@halcyon.usc.edu

April 13 - 17 TAPSOFT 93
TAPSOFT 93 (CAAP FASE Advanced Seminar) Orsay, France Sponsored by : AFCET, EATCS Contact: Program Chair of FASE: Marie-Claude Gaudel, Program Chair of CAAP: Jean-Pierre Jouannaud, TAPSOFT 93, LRI Batiment 490, Universite Paris XI, 91405 Orsay Cedex, France, Fax: 33 1 69 41 65 86 E-Mail: tapsoft@lri.lri.fr *or* jouannaud@margaux.inria.fr *or* AFCET, 156 Bd Pereire, 75017 Paris, Fax: 33 1 42 67 93 12

April 19 - 23 FME'93
Industrial Strength Formal Methods Odense Technical College, Denmark Contact: Programme Chairman, Jim C.P. Woodcock, Oxford Univ. Computing Lab., Prog. Research Group, 11 Keble Road, Oxford OX1 3QD, UK, tel. +44 865 272576, Fax: +44 865 273839. E-Mail: jimw@prg.ox.ac.uk *or* Organising Chairman, Peter Gorm Larsen, The Inst. of Applied Comp. Sci. (IFAD), Forskerparken 10, DK-5230 Odense M, Denmark, Tel: +45 65 93 23 00, Fax: +45 65 93 29 99. E-Mail: peter@ifad.dk

April 19 - 23 ICDE
9th Internationcal Conf. on Data and Eng Vienna, Austria Contact: Eruch J. Neuhold, GMD-IPSI, Dolivostrasse 15, D-6100 Darmstadt, Germany; Tel: (+49) 6151 869 803. E-Mail: darmstadt.gmd.de

April 20 - 23 *History of Prog. Lang* Boston, Mass., USA Sponsored by : SIGPLAN Contact: Jan Lee, CIT ITT 133 McBryde Hall, Blacksburg, VA 24061-0119; Tel: (703) 231-5780. E-Mail: janlee@vtmi.bitnet

May 16 - 18 STOC'93
25th Annual ACM Symp. on the Theory of Computing 1993 San Diego, Calif., USA Sponsored by : SIGACT Contact: David S. Johnson, AT&T Bell Labs, 600 Mountain Ave., Rm. 2D-150, Murray Hill, NJ 07974; Tel: (908) 582-4742. E-Mail: dsj@research.att.com

May 17 - 21 ICSE
15th Int'l. Conf. on Software Eng Baltimore, Maryland, USA Contact: Victor R. Basili, Dept. of Comp. Sci., Univ. of Maryland, College Park, Maryland 20742, USA; Tel: (301) 405-2668. E-Mail: basili@cs.umd.edu

May 24 - 26 *1993 IEEE Symp. on Research in Security and Privacy* Oakland, California, USA Sponsored by : IEEE Comp. Soc. and Technical Committee on Security and Privacy Contact: Richard Kemmerer, Comp. Sci. Dept., Univ. of California, Santa Barbara, CA 93106, Tel: (805) 893-4232, Fax: (805) 893-8553. E-Mail: kemm@cs.ucsb.edu *or* John Rushby, SRI Int'l., EL254, 333 Ravenswood Avenue, Menlo Park, CA 94025, Tel: (415) 859-5456, Fax: (415) 859-2844. E-Mail: rushby@csl.sri.com *or* Jeremy Jacob, Oxford Univ. Computing Lab., 11 Keble Road, Oxford, England OX1 3QD, Tel: +44 865 272562, Fax: +44 865 273839. E-Mail: jeremy.jacob@prg.oxford.ac.uk

May 25 - 28 *The 13th Int'l. Conf. on Distributed Computing Systems* Sponsored by : IEEE Comp. Soc. Pittsburgh Hilton, Pittsburgh, Pennyslvania, USA Contact: Benjamin W. (Ben) Wah, Coordinated Sci. Lab., Univ of Illinois, MC228, 1101 W. Springfield Avenue, Urbana, IL 61801-3082, Tel: (217) 333-3516; Fax: (217) 244-7175. E-Mail: b-wah@uiuc.edu

June 9 - 11 *Functional Prog. Lang. and Comp. Architectures* Copenhagen, Denmark Sponsored by : SIGPLAN and SIGARCH in cooperation with IFIP WG 2.8 Contact: John Williams, IBM Almaden Research Center K53-803, 650 Harry Road, San Jose, CA 95120; Tel: (408) 927-1888. E-Mail: williams@ibm.com

June 14 - 17 *The Fifth Asian Logic Conf* National Univ. of Singapore, Republic of Singapore Contact: The 5th ALC, Dept. of Mathematics, National Univ. of Singapore, Singapore 0511, Republic of Singapore. E-Mail: matlogic@nuscc.nus.sg *or* matlogic@nusvm.bitnet

June 15 - 18 *7th Int'l. Symp. on Methodologies for Intelligent Systems* Trondheim, Norway Contact: Jan Kmorwski, Univ. of Trondheim, Norwegian Inst. of Tech., Dept. EE and Comp. Sci, N-7034 Trondheim, Norway. E-Mail: janko@idt.unit.no *or* Zbigniew W. Ras, UNC-Charlotte, Dept. of Comp. Sci., Charlotte, NC 28223.

June 16 - 18 RTA-93

Fifth Int'l. Conf. on Rewriting Techniques and Applications Montreal, Canada Contact: Claude Kirchner, RTA-93, INRIA Lorraine & CRIN, Campus scientifique, 615 rue du Jardin Botanique, BP 101, 54602 Villers-les-Nancy CEDEX, France, Tel: (33) 83 59 30 11, Fax: (33) 83 27 83 19. E-Mail: Claude.Kirchner@loria.fr *or* Mitsuhiro Okada, RTA-93, Dept. of Comp. Sci., Concordia Univ., H3G1M8 Montreal, Quebec, Canada, Tel: (1) (514) 848 30 48, Fax: (1) (514) 848 28 30. E-Mail: RTA93@concour.cs.concordia.ca

June 16 - 18 SEKE'93

5th Int'l. Conf. on Software Eng. and Knowledge Eng. San Francisco, Calif., USA Contact: Bruce I. Blum, Applied Physics Laboratoty, Johns Hopkins Univ., Laurel, MD 20723-6099; Tel: (301) 953-6235; Fax: (301) 953-6904. E-Mail: bib@aplcomm.jhuapl.edu *or* C.L. Chang, Lockheed S/W Tech Center, Org 96-10, Bldg. 254E, 3251 Janover St., Palo Alto, CA 94304; Tel: (415) 424-5379; Fax: (415) 424-2999. E-Mail: chang@stc.lockheed.com

June 21 - 25 Petri Nets'93

The 14th Int'l. Conf. on Application and Theory of Petri Nets Bismarck Hotel, Chicago, USA Contact: Prof. T. Murata, Dept. of EECS (m/c 154), Univ. of Illinois at Chicago (UIC), P.O. Box 4348, Chicago, IL 60680, USA. E-Mail: pn93@uicbert.eecs.uic.edu

June 22 - 24 FTCS 23

The Twenty Third Annual Int'l. Symp. on Fault-Tolerant Computing Toulouse, France Sponsored by : IEEE Comp. Soc. and LAAS-CNRS in cooperation with AFCET and IFIP WG 10.4 Contact: Marie-Therese Ippolito, LAAS-CNRS, Tel: +(33) 61 33 62 74, Fax: +(33) 61 55 35 77. E-Mail: Marie-Therese.Ippolito@laas.fr

June 22 - 25 AMAST

Third Int'l. Conf. on Algebraic Methodology and Software Tech. Univ. of Twente, Enschede, The Netherlands Abstract to: AMAST Conf., Univ. of Twente, Fac. Informatica, PO BOX 217, NL-7500AE Enschede, The Netherlands. *or* Canada: V.S. Alagar, Concordia Univ., Dept. of Comp. Sci., 1455 De Maisonneuve Blvd. West, Montreal, Quebec H3G 1M8, Canada, Tel: +1 514 8483022, Fax: +1 514 8483000. E-Mail: alagar@concour.cs.concordia.ca *or* Europe: Charles Rattray, Univ. of Stirling, Dept. of Mathematics and Computing Sci., Stirling, Scotland, FK9 4LA, Great Britain, Tel: +44 786 73171, Fax: +44 786 64551. E-Mail: cr@cs.stir.ac.uk *or* USA: Teodor Rus, Univ. of Iowa, Dept. of Comp. Sci., Iowa City, IA 52242, USA, Tel: +1 319 3350742, Fax: +1 319 3350627. E-Mail: rus@cs.uiowa.edu

June 28 - 30 LP&NMR-93

2nd Int'l. Workshop Logic Prog. and Non-Monotonic Reasoning Lisbon, Portugal Contact: Anil Nerode, Mathematical Sci.s Inst., Cornell Univ., Ithaca, NY 14853.

June 28 - 30 ISSTA 1993

Int'l. Symp. on Software Testing and Analysis Cambridge, Massachusetts, USA Sponsored by : ACM SIGSOFT Contact: John Gannon, Dept. of Comp. Sci., Univ. of Maryland, College Park, MD 20742, USA. Tel: : (301) 405-2671. E-Mail: gannon@cs.umd.edu

July 5 - 9 ICALP'93

20th Int'l. Coll. on Automata, Lang., and Prog. Lund, Sweden Contact: Prof. Rolf Karlsson, Dept. of Comp. Sci., Lund Univ., S-221 00 Lund, Sweden. E-Mail: icalp93@dna.lth.se

August 23 - 27 FCT'93

Fundamentals of Computation Theory Szeged, Hungary Contact: T. Gaizer *or* J. Viragh, FCT'93 Bloyai Inst., A. Jozsef Univ., 6721 Szeged, Aradi v. tere 1., Hungary, Fax: 36-62-12292. E-Mail: h754esi@ella.hu, h1299gai@ella.hu, J68A004@HUSZEG11

Guidelines for Newsletter Contributions

Contributions may be in the form of single-sided camera-ready copy, suitable for layout and sub-editing. They can also be sent to us using electronic media (i.e. by floppy disk (MS DOS or Mac)/e-mail/etc.), to be formatted in the house style. As a rule, we generally accept pure ASCII text or TEX/LATEX in order to avoid complications involving interchange between wordprocessing formats. We regret that we are unable to offer typesetting facilities for handwritten material.

If contributions are sent using proprietary wordprocessor/markup language formats (i.e. MicroSoft Word 5, FrameMaker), then these will be treated as though they were camera-ready copy. If we are unable to print them adequately or to otherwise convert to another more suitable form then the authors may be asked to provide paper copies of appropriate reproduction quality.

Artwork can be provided for appropriate inclusion, either using general formats (such as DVI files or Encapsulated PostScript[1]) or by sending camera-ready paper copy. Generally, line drawings and other high-contrast graphical diagrams will be acceptable.

Material must be of adequate quality for reproduction. Output from high quality printers with at least 300 DPI resolution is generally acceptable. Output from printers with lesser resolution (i.e. dot-matrix printers) tends not to reproduce very well and will not be of sufficiently good print quality. The Editorial Panel reserves the right to refuse publication for contributions which cannot be reproduced adequately.

Page definition information

If possible, contributions should be designed to fit standard A4 paper size, leaving a margin of at least one inch (1") on all sides. Camera ready copy should be sent in **single-sided** format, with page numbers written lightly **on the back**. Ideally, all fount sizes used should be no smaller than 10pt for clarity. Contributions should attempt to make adequate use of the space, filling at least 60% of each page, and including the final page. Authors should note that all contributions will be sub-edited appropriately to make efficient use of space.

Addresses

General Correspondance to the editor:
The Editor, BCS FACS FACTS Newsletter,
c/o Department of Computer Studies,
Loughborough University of Technology
Loughborough, Leicestershire
LE11 3TU
United Kingdom

Tel: (0509) 222676
E-mail: FACS@lut.ac.uk

Technical Contributions Coordinator:
Ian Maung
BCS-FACS Technical Contributions Coordinator
Dept of Computing
University of Brighton
Moulsecoombe
Brighton, East Sussex
BN2 4GJ
United Kingdom

Tel: 0273 642492
Fax: 0273 642405
E-mail: iml@unix.brighton.ac.uk

[1]PostScript is a trade mark of Adobe Systems, Inc.

BCS FACS Committee 1992/93

General

General enquiries about the BCS FACS group, the newsletter or its meetings can be made to:

BCS FACS
Department of Computer Studies
Loughborough University of Technology
Loughborough, Leicestershire
LE11 3TU
Tel: 0509-222676
E-mail: FACS@lut.ac.uk

Membership fees 1993

Standard (i.e. non-BCS members) : £25
BCS members : £10

Discount subscription rates 1993

EATCS : £10
FACS Journal : £33 (6 issues, Vol. 5)

Officers

Chair	Tim Denvir
Treasurer	Roger Stone
Committee Secretary	Richard Mitchell
Membership Secretary	John Cooke
Newsletter Editor	Jawed Siddiqi (Dan Simpson)
Publicity	Brian Monahan
BCS SIG representative	David Blyth
BCS SE TC representative	John Boarder (Roger Shaw)
Liaison with FACS Journal	John Cooke
Liaison with BCS FMIS group	Ann Wrightson

Committee Members

Name	Affiliation	Tel:	E-mail
R. Barden	Logica Cambridge Ltd	0223-66343	rosalind@logcam.co.uk
D. Blyth	Incord Ltd.	0202-896834	DBlyth@cix.compulink.co.uk
J. Boarder	Buckinhamshire	0494-22141	
Dr. D.J. Cooke	Loughborough	0509-222676	D.J.Cooke@lut.ac.uk
B.T. Denvir	Translimina Ltd.	081-882 5853	timdenvir@cix.compulink.co.uk
Prof. S.J. Goldsack	Imperial	071-589-5111x5099	sig@ic.doc.ac.uk
Dr. A.J.J. Dick	Bull Research		J.Dick@uk03.bull.co.uk
R.B. Jones	ICL Winnersh	0734-693131x6536	R.B.Jones%win0109.uucp@uknet.ac.uk
Dr. R.J. Mitchell	Brighton	0273-642458	rjm4@unix.brighton.ac.uk
Dr. B.Q. Monahan	Manchester	061-275-6137	brianm@cs.man.ac.uk
Prof. A. Norcliffe	Sheffield Hallam	0742-720911x2473	A.Norcliffe@scp.ac.uk
R.C.F. Shaw	Lloyd's Register	081-681-4040	ttercs@aie.lreg.co.uk
Dr. J.I.A. Siddiqi	Sheffield Hallam	0742-533171	jawed@cms.scp.ac.uk
Prof. D. Simpson	Brighton	0273-600900x2273	ds33@unix.bton.ac.uk
Dr. R.G. Stone	Loughborough	0509-222686	R.G.Stone@lut.ac.uk
D.R. Till	City	071-477-8552	till@cs.city.ac.uk
Dr. Ann Wrightson	Central Lancashire	0772-893242	annw@sc.lancsp.ac.uk