FACS FACTS

The Newsletter of the BCS Formal Aspects of Computing Science SIG

Series II Vol 3 No 3 **Aug 92**

Continuing out theme on reports on Formal Methods Tools, this issue includes an overview of the Mural System.

We also have the results of two surveys about the penetration of Formal Methods in industry and academia. The results might not be statistically useful or indeed the surveying methods used not rigorous but they do however provide interesting reading.

Happy Summer

Jawed Siddiqi

FACS Committee Officers

Chairman:

Dr D John Cooke
Department of Computer Studies
Loughborough University of Technology
LOUGHBOROUGH
Leicestershire
LE11 3TU
Tel: (0509) 222676
Fax: (0509) 610815
E-mail: D.J.Cooke@uk.ac.lut

Secretary:

B Tim Denvir
Translimina Ltd
37 Orpington Road
Winchmore Hill
LONDON
N21 3PD
Tel: (081) 8825853

Treasurer:

Dr Roger G Stone
Department of Computer Studies
Loughborough University of Technology
LOUGHBOROUGH
Leicestershire
LE11 3TU
Tel: (0509) 222676
Fax: (0509) 610815
E-mail: R.G.Stone@uk.ac.lut

Newsletter Editor:

Dr Jawed I A Siddiqi
Department of Computer Studies
School of Computing and Management Sciences
Sheffield City Polytechnic
100 Napier Street
SHEFFIELD
S11 8HD
Tel: 0742 533155 direct/ msg 533171
E-mail: JAWED@uk.ac.scp.cms
FAX: 0742 533161

Publicity:

Dr Brian Q Monahan
Department of Computer Science
The University of Manchester
Oxford Road
MANCHESTER
M13 9PL
Tel: (061) 275 6137
Fax: (061) 275 6280
E-mail: bqm@uk.ac.man.cs

Specialist Groups Management Committee Representative:

David Blyth
INCORD Ltd
15 Sherwood Avenue
Ferndown
WIMBORNE
Dorset
BH22 8JS
Tel: (0202) 896834
Fax: (0202) 894834
E-mail: DBlyth@cix.compulink.co.uk

FACS Committee

Dr A Jeremy J Dick
Centre de Research BULL
F3/2GO7
Fue Jean Jaures
78340 LES CLAYES-SOUS-BOIS
FRANCE
Tel: -33 1 30 80 69 23
E-mail: Jeremy.Dick@crg.bull.fr

Professor Stephen J Goldsack
Department of Computing
Imperial College
180 Queens Gate
Kensington
LONDON
SW7 2BZ
Tel: (071) 589 5111
Fax: (071) 581 8024
E-mail: sjg@uk.ac.ic.doc

Dr Richard J Mitchell
Department of Computing
Brighton Polytechnic
Moulsecoomb
BRIGHTON
Sussex
BN2 4GJ
Tel: (0273) 588641
Fax: (0273) 571701
E-mail: rjm4@uk.ac.bton.vms

Roger Shaw
Performance Technology Group
Lloyds Register of Shipping
Lloyds Register House
29 Wellesley Road
CROYDON
CR0 2AJ
Tel: (081) 681 4818
E-mail: rcs@uk.co.lreg.aie

Professor Dan Simpson
Department of Computing
Brighton Polytechnic
Moulsecoomb
BRIGHTON
Sussex
BN2 4GJ
Tel: (0273) 600900 ext. 2273
Fax: (0273) 681752
E-mail: ds33@uk.ac.bton.vms

Dr David Till
Department of Computer Science
City University
Northampton Square
LONDON
EC1V 0HB
Tel: (071) 477 8552
Fax: (071) 477 8560
E-mail: till@uk.ac.city.cs

Stephen Webster
Department of Computing and Cognition
Bournemouth Polytechnic
Fern Barrow
Dorset
Tel: (0202) 595250
E-mail: swebster@bmthpoly.uucp.ac.uk

Professor Allan Norcliffe
Department of Mathematical Sciences
Sheffield City Polytechnic
SHEFFIELD
S1 1WB
Tel: (0742) 720911 ext. 2473

John C Boarder
"Cartreff"
New Inn Road
Beckley
OXON
OX3 9SS
Tel: (086735) 475 Home

Discrete Mathematics and Software Development
Some results from a survey

The following notes document the results of an attempt to discover the extent of the penetration of formal mathematical approaches into the teaching of software development in degree level courses in the U.K. The survey was carried out under the auspices of BCS FACS; we were interested to see how our 'special interest' was dealt with by Higher Education curricula.

All Computing departments in Universities, Polytechnics (as they used to be called) and H.E. institutions were circulated with a questionnaire. One hundred and five institutions were targetted and thirty eight responded. One immediate concern was that the replying institutions would be those for whom formal approaches to software development were the preferred approaches, whereas the institutions that did not reply would be those that rejected such approaches. Telephone approaches to colleagues in institutions which did not originally reply tended to confirm that the responses received were typical. In fact, we even received a few null responses from departments whose computing courses contained no formal mathematical inputs. We have reason to believe, therefore, that the replies are fairly representative of the state of computing courses in Higher Education.

The questionnaire was deliberately flexible in structure as it had to cope with the many varied course structures that are to be found nationally. Many courses are completely modular and some computing courses have a wide variety of routes to a computing qualification. We interpreted the results in such cases by assuming routes through courses which deliberately selected the formal computing options. This will have introduced some bias into the figures. Many (most) of the routes through computing courses that we selected were called 'software engineering' or something very similar. If you assume that this study is concerned broadly with undergraduate software engineering courses rather than computer science courses then the bias will be minimal.

The questionnaire is appended to this report. The results that follow are outlined in the same sequence as the questions that were put to the institutions. This is followed by a caricature of a 'typical' undergraduate computing course and then some more general conclusions, comments and observations.

Separation or integration?

Over two thirds of the responses indicated that discrete mathematics and formal notations were taught as separate units to those concerned with software development. The use of the formal mathematics was largely in the role of making specifications more precise. There were only two responses which suggested that their courses saw software development as an activity rooted in mathematics and organised their curriculum accordingly. Roughly one third of the respondents suggested that they were actively considering/investigating ways to increase the integration of mathematics with software development.

How much formality?

Typically this is seen as a decreasing proportion of a computing course as the course progresses through the (usually) 3 years. The following figures are averaged from all the replies received. Approximately one fifth (20.4%) of the first year is given over to discrete mathematics or formal program development. One sixth (16.1%) of the final year is given over to mathematical formality. The second year turns out to be in between these two (18.4%). It is clear that mathematics is considered an important component.

Notations

The propositional and predicate calculus were almost universal although there was no agreement on precise syntax. The following table outlines the number of references to particular notations.

Notation	Number of references
Z	23
VDM	10
OBJ/algebraic specification	5
CCS/CSP	5
Functional language (eg. SML)	4
Prolog	1

A glance through the latest publisher's catalogues will confirm the accuracy of this table.

More or less formality

There do not appear to be any plans to reduce the formal component of any courses. Approximately one third of the respondents claimed to be considering an increase in the proportion of time spent.

Industrial feedback

The plain truth here is that there has been very little feedback of any kind. The little positive feedback that has come has been as a result of industrial liaison committees or from joint research projects.

A 'typical' undergraduate computing course

A typical course will contain a first year discrete mathematics unit which will cover propositional and predicate calculus and some set theory. A second year course entitled something like 'formal specification', which will cover a particular formal notation for specification, either Z or VDM. A final year unit would then address something like 'formal approaches to concurrency' or 'program derivation'.

Conclusions, comments and observations

BCS FACS Questionnaire

Formal Methods of Software Development in U.K. Higher Education

Introduction
We are attempting to discover the extent to which Discrete Mathematics has penetrated the teaching of Software Development. We are adopting a broad interpretation of the phrase Software Development, including all the traditional life cycle phases of analysis, design and implementation. We have an equally broad view of the Discrete Mathematics and we include the following areas of interest;

Program Derivation - Gries, Djikstra, Dromey sort of stuff.
Formal specification notations, Z, VDM etc..
Refinement
Executable specifications - Prolog, SML, OBJ.

Courses come in many varieties and modes of delivery, any questionnaire (including this one) looking at this topic area will of necessity ask questions which do not quite seem to fit the teaching structure that you operate. We have adopted the solution of asking very flexible questions thus postponing the hard bit until we have your answers.

We assume that it will take about half an hour to answer these questions. Your patience will be rewarded with a copy of the results.

The questions assume that your answering will refer to one course, most likely your undergraduate computing/software engineering course. I f you have more than one course to describe would you please keep the details for each separate. Highly modular courses could indicate compulsory areas and typical student choice profiles.

Questions

1. What is the role of discrete mathematics in software development in your course? You could possibly send a standard course outline here.
   - Is the mathematics separate from the analysis or design or programming?
   - What proportion of the course is spent on formal methods?
   - Is this a first year, second year or final year topic?
   - Which notations do you teach your students?

2. From your current position, do you plan to increase the level of formality, leave it as it is or raise it?

3. Can you describe any industrial feedback that you have had as a result of teaching or not teaching formal notations to your students.

Software development and formal mathematics are not as integrated in undergraduate computing courses as one might hope with one or two notable exceptions. Mathematics is used largely as a tool for supporting the specification process and Z and VDM are the most popular notations by a long way.

The responses we received concerning higher degrees painted a slightly different picture but there was not enough feedback to make more than a very general comment. Many MSc courses, both conversion and follow-on, had a high percentage of mathematical input. These were anywhere between 25% and 50% of the courses. We seem, therefore, to believe that mathematics is important in software development but more important at the higher academic levels.

One very useful idea that emerged was that we should use the FACS journal as a forum for dissemination of ideas, particularly concerning the reporting of attempts to integrate mathematics and software development.

We learnt a lot about the construction of questionnaires and internal departmental communications through this exercise. (Our own institutions failed to respond!) We send our thanks to all those people who replied to the questionnaire and hope that the results will be helpful to you. If you wish to discuss any matters further please contact the authors at the addresses below.

Steve Webster
Dept. of Computing and Cognition
Bournemouth University
0202 595250
swebster@uk.ac.bournemouth

Dan Simpson
Department of Computing
Brighton University
0273 600900
ds33@uk.ac.bton.vms

Report on BCS Bristol Branch Spring Lecture Series on Formal Methods

In March and April of this year, the Bristol Branch of the BCS ran a Spring Lecture Series on Formal Methods. The Series was spread over 6 Tuesday Evenings and the Programme was as follows:

1 Intro to Formal Methods (A. Hall, Praxis); Intro to Discrete Maths (M. Bartley, Praxis)
2 Intro to Z (J. Woodcock, PRG Oxford)
3 Intro to VDM (M.Bartley, Praxis)
4 VDM Workshop (a number of Praxis employees)
5 Case Studies (P. Harry, HP; D. Clutterbuck, Program Validation; I. Houston, IBM)
Tools demonstrations: Mural (Manchester University), Cadiz (York Software Engineering)
6 The Future and Open Forum (M. Thomas, Praxis; I. Holyer, Bristol University)

The cost was 40 pounds to BCS members and 55 pounds to non-members. 72 people registered for the course (we were oversubscribed) and the average attendance each evening was 55 to 60. A questionaire was distributed on the last evening and 48 replies were received. The results are below.

① What was your main reason for attending?

Personal Interest	Work	Industry Standards Specify Formal Methods	Professionalism
36	8	3	1

② In what sector of industry are you employed?

Defence	Education	S/W Tools	Financial	Engineering	Transport	Security	End User
11	5	5	6	14	3	2	2

③ Do you use Formal Methods at work?

Yes - a lot	Yes - a little	No
4	4	40

(4 of the yes's were from people in Education)

If so - what language/method?

Z	VDM	CSP
6	2	3

④ Do you intend to start/continue using Formal Methods?

Yes	Probably	Depends on clients	Forced to by Law	No	Don't know
21	8	2	4	6	7

⑤ Do you think Formal Methods are sufficiently mature to be used on real projects?

Yes	Only for the Specification	Depends on Tools	Yes - but ltd applications	Not at the moment	No	Don't know
15	2	4	8	7	6	6

⑥ What, in your opinion, is the main driving force behind the adoption of Formal Methods?

Safety Requirements	Getting the Spec right	Client demand	Legislation	The need for proofs	Quality	Better designs
15	15	4	3	3	5	3

In my opinion a number of points should be drawn to the readers attention:

1 The number of people attending for work reasons was relatively low. Most people attended out of personal curiosity and a desire to learn something new. Having said that the level of interest was very high.
2 50% of the people using Formal Methods at work are simply teaching it.
3 I personally feel that the number of people answering yes to ④ is high due to the fact that they had just finished a course on Formal Methods and were therfore keen to employ them.
4 They were not prompted at all for any of the answers.

I personally don't feel that these results are statistically very useful and there cannot be a high level of confidence in them. However, having said that I think they make interesting reading.

Machine Support for Formal Methods
An Overview of Mural

Richard Moore
Dept. of Computer Science
University of Manchester
Manchester M13 9PL

21 July 1992

1 Introduction

As computers are being used more and more widely in situations where the reliability of their software is critical, so attention is being turned to formal methods as a possible means of increasing this reliability. Many people nowadays realise that formal specification can play an important rôle in helping software designers to gain a greater understanding of the system they are building by providing an abstract, unambiguous language in which the functionality of a piece of software can be described. In most cases, however, the formal specification simply forms an isolated "stepping stone" between some informal requirements document written in (possibly structured) natural language and the actual code forming the implementation of the system, and little attempt is made to show either that the specification is itself sound or that it represents a valid description of the system defined by the informal requirements document. The mathematical basis of specification languages means that mathematical reasoning can be used to help with both of these tasks. This, by its very nature, is readily amenable to computer support. This paper describes how the mural system addresses these issues.

Basically there are two parts to mural, a *specification support tool* where specifications are constructed and a mathematical reasoning environment or *proof assistant* where mathematical reasoning about a specification can be performed. The two are linked by a "translator" whereby an environment supporting mathematical reasoning about a particular specification can be generated automatically from that specification. At present, the specification support tool and the translator only support VDM, though any other specification language could be treated entirely analogously. The proof assistant, on the other hand, is generic and will support reasoning about specifications written in any of a wide range of specification languages as it stands.

2 The Specification Support Tool

The current version of the specification support tool provides a series of structure-editor-based operations for constructing specifications using a large subset of the emerging standardised specification language VDM-SL. Mural uses the mathematical syntax of VDM-SL, though specifications can also be generated from the ascii syntax by way of an interface to the related tool SpecBox. SpecBox reads a text file written in this ascii syntax and checks that it conforms to this syntax. It also performs some consistency checks on the specification, and provides a mechanism whereby any errors discovered can be corrected interactively. Once the file passes all these tests it can be converted to a form which is directly readable by mural. Both these methods of building specifications ensure that only syntactically correct specifications can be constructed in mural.

3 The Proof Assistant

Mural's proof assistant basically consists of a collection of mathematical theories. These theories fall into two categories: those supporting reasoning about the basic data-types of the specification language itself (e.g. the logic of partial functions, sets, maps, sequences, etc. for VDM-SL) and those supporting reasoning about a particular specification. The system is generic in that it can be instantiated with a basic set of theories describing the data-types of other specification languages if so desired.

Each theory contains a set of declarations of the mathematical symbols used to describe the properties of the data-type or specification in question, so that, for example, the theory of sets will include the set type constructor -set, the operators for set union ($\cup$), set intersection ($\cap$), cardinality (card), and many more. A theory also contains a set of *inference rules* embodying the various properties of these symbols. Thus, for example, the inference rule stating that the set union operator is commutative has the form:

$$\boxed{\cup\text{-comm}}\ \frac{s_1 : A\text{-set},\ s_2 : A\text{-set}}{s_1 \cup s_2 = s_2 \cup s_1}$$

Here the interpretation is that the formula below the horizontal line is true if all the formulae above the line are true. Thus, in the above example, $s_1 \cup s_2 = s_2 \cup s_1$ provided that both s_1 and s_2 are sets.

One very important feature of mural's proof assistant is that new inference rules can be added to any theory at any time. These can be proved (using already existing inference rules), thus extending the reasoning power of the system.

Some subset of a theory's inference rules are designated as its *axioms* and these are taken to embody the most primitive reasoning steps which can be made in a theory in that they are taken to be true without proof. Inference rules which are not axioms have an associated proof, though this might be incomplete as mural will allow an inference rule to be used in a proof even if it has not been proved. This is an important feature as it admits a notion of rigorous proof in which "obvious" steps in a fully formal proof can be omitted by stating them as unproved inference rules.

Mural keeps track of everywhere that unproved inference rules have been used to prove other inference rules, and proved rules are thus only proved modulo some (possibly empty) set of these unproved inference rules (often called lemmas). This same dependency mechanism is also used to ensure that no circularities in reasoning are introduced into the system, so that, for instance, it is impossible to use a rule r_1 in the proof of some other rule r_2 and then to prove rule r_1 using rule r_2.

This ability to leave steps in a proof unproved is particularly helpful when proving properties of specifications. These proofs generally tend to be fairly long and tedious, though for the most part they are conceptually fairly lightweight with possibly only a few points requiring any major amount of thought. The facility means that the lightweight parts can be factored out as unproved lemmas and attention can be concentrated around those parts of the proof which contain the crux of the reasoning. One can easily envisage a proof review process based on this idea, in which someone with specialist mathematical knowledge simply looks at the lemmas left unproved by someone else trying to prove some properties of some specification and can pronounce them as either acceptable (in the sense of believed to be valid) or unacceptable. Further work could then be done on the proofs of those lemmas flagged as unacceptable, possibly by reducing them to other lemmas, and any desired degree of formality in the reasoning process can thus be obtained.

Proofs in mural are performed in natural deduction style and use the full mathematical syntax. However, no particular order of working is imposed, so the user is free to adopt whatever style seems more natural. Thus, it is possible to use *forward reasoning*, where new facts are deduced from assumptions and already known facts, or *backward reasoning*, where some goal that is still to be proved is reduced to subgoals from which it can be proved.

Mural provides a pattern-matching facility with which it can decide whether or not a particular inference rule can be applied to a selected set of facts and/or a selected goal. This forms the basis of its *controlled search* facility in which the user can designate an area of search (a set of theories and/or a set of inference rules) and mural will display those inference rules in the search area which can be applied to the selected facts/goal alongside all the possible instantiations of each applicable rule. This facility is particularly useful for new users who are unfamiliar with all the inference rules available in the system (currently around 800).

Another important facility of the proof assistant is its *tactic language*. Using this, the user can "program" commonly-used proof strategies as parametrized tactics, using which a whole series of steps of a proof can be generated at once. As an example, one of the existing tactics takes as its parameter a list of inference rules and effectively just plods through the list trying to apply each rule in turn somewhere in some proof until none of them can be applied any more.

One problem with allowing the user the freedom to write new tactics and to pass essentially arbitrary arguments to existing tactics is that it is then impossible to guarantee either that the tactic will terminate or that it will have generated any useful proof steps even if it does terminate. For this reason, mural copies the current state of a proof before a tactic starts execution and applies the tactic to the copy. As the tactic runs changes to the proof are displayed. Execution can be terminated if these changes look unreasonable, and work can continue on the original proof.

In fact this copying of proofs is just one aspect of a more general facility which allows more than one proof to be attached to a given inference rule. This facility allows the user to start a completely new proof from scratch if some proof seems to be getting nowhere, or to duplicate an existing partially complete proof, for instance to try out different ideas on it. When an inference rule has one complete proof attached to it a "garbage collection" facility is offered which not only discards all incomplete proofs attached to the same inference rule but also removes any redundant lines from its complete proof (e.g. facts which might have been deduced by forward reasoning but which were never needed to justify any of the subgoals generated by backward reasoning).

4 The Translator

Given a specification in the specification support tool, mural's translator will automatically construct a theory in the proof assistant which will form the basis for reasoning about that specification. This theory will contain a set of symbol declarations corresponding to the type definitions, the auxiliary functions, etc. etc. defined in the specification, together with a set of axioms describing their basic properties. Symbols and axioms which describe auxiliary constructs implicitly defined in the specification but not appearing explicitly there (such as the *mk-* function and the selector functions for a composite type definition) are also generated as part of this process. In addition, the translator builds an unproved rule for each of the proof obligations associated with the specification. These proof obligations embody the conditions under which the specification is internally self-consistent or sound. The proof assistant would then be used to show that these conditions are satisfied.

Once generated, of course, this theory has exactly the same properties as any other theory in the proof assistant. This means that the user can extend it freely by adding new, unproved inference rules to it by hand. These can then be used to make the task of discharging the proof obligations easier. They can also be used to state "obvious" properties of the specification and left unproved, thus allowing the user to decide the degree of formality appropriate for each part of the specification in the way described above.

Another important use of this extensionality is in trying to show that a formal specification is a faithful representation of the system described in the informal requirements document. The idea is to annotate the informal requirements document with statements about fundamental properties that the desired software is expected to exhibit. Then, when the informal requirements document is transformed, by whatever means, into the abstract formal specification these annotations are simultaneously transformed to *formal* annotations on the specification, these formal annotations stating the equivalent properties that the specification should exhibit in order that the desired property of the system be preserved. These formal annotations can finally be added as unproved rules to the theory generated from the specification by mural's translator. Proving these rules would then show that the specification possesses the desired property. Of course this process does not, and can not, provide proof that the formal specification is a faithful representation of the informal requirements, but it can at least serve to increase one's confidence that this is so.

5 Data-type Refinement

Mural's specification support tool offers a facility whereby one specification can be designated as a refinement of another. When this is done, the user defines a retrieve function and indicates for each operation in the concrete specification the corresponding operation in the abstract specification of which it is claimed to be an implementation. Having done this, the translator can be invoked exactly as described above to generate a theory supporting reasoning about the refinement step. This theory will contain a symbol declaration and axioms describing the retrieve function plus unproved rules stating the proof obligations of the refinement (adequacy of retrieve function plus domain and result obligations for each operation pairing for VDM). These are again proved with the help of the proof assistant.

6 Further Reading and Availability

Further information on mural can be found in a recent book: "mural : A Formal Development Support System" by C.B. Jones, K.D. Jones, P.A. Lindsay and R. Moore (Springer Verlag, June 1991; ISBN 3-540-19651-X and 0-387-19651-X).

Further information on SpecBox can be found in: P.K.D. Froome, B.Q. Monahan and R.E. Bloomfield: "SpecBox - a checker for VDM specifications", in Proceedings of Second International Conference on Software Engineering for Real Time Systems, Cirencester, UK, IEE 1989.

A large (at least 12Meg of memory) workstation running version 2.5 of Smalltalk-80 is required to run mural. Academic licences are available at a cost of £100 (handling charge). For non-academics, an evaluation licence limiting use to non-commercial exploitation costs £250 at the time of writing. Licences for commercial purposes can also be supplied, but are subject to individual negotiation. Mural can be supplied to non-academics with an embedded run-time licence for Smalltalk-80 at an additional cost of around £400. Training and consultancy services based around mural are also available.

Call for Papers
Third International Conference on Algebraic Methodology and Software Technology, AMAST

Goals and Organization

The goal of the third AMAST conference to be held on June 22–25, 1993, at the University of Twente, Enschede, The Netherlands, is to consolidate the trend towards using algebraic methodology as a foundation for software technology, and to show that universal algebra provides a practical mathematical alternative to the common, ad-hoc approaches to software engineering and development. Academia and industry are both beneficiaries of such a formal foundation.

Organizing Committee:

General chairman:	Maurice Nivat	(University of Paris VII, Paris, F)
Programme chairman:	Giuseppe Scollo	(University of Twente, Enschede, NL)
Publicity chairman:	Charles Rattray	(University of Stirling, GB)
	Teodor Rus	(University of Iowa, Iowa City, IA, USA)
	V.S. Alagar	(Concordia University, Montreal, Canada)
Local chairman:	Ed Brinksma	(University of Twente, Enschede, NL)
Finance chairman:	Frans van der Avert	(University of Twente, Enschede, NL)
Members:	Mohammed Bettaz	(University of Constantine, Algeria)
	Christine Choppy	(University of Paris-Sud, LRI, Orsay, F)
	Pierre Deransart	(INRIA, Rocquencourt, F)
	Arthur Fleck	(University of Iowa, Iowa City, IA, USA)
	Luigi Logrippo	(University of Ottawa, Ontario, Canada)
	Michael O'Donnell	(University of Chicago, IL, USA)
	Juan Quemada	(University of Madrid, E)
	Ralph Wachter	(Office of Naval Research, Arlington, VA, USA)

Local Committee: Herman Balsters, Han Bäumer, Pim van den Broek, Rolf de By, Maarten Fokkinga, Pim Kars, Mark van de Voort, Job Zwiers

Secretariat: Kirstin Reinink, Yvonne Rokker

Programme Committee:

Martin Abadi, Egidio Astesiano, Choukri-Bey Ben-Yelles, Michel Bidoit, Wim Blok, Chris Brink, Pierre-Luis Curien, Kokichi Futatsugi, Steven Givant, William S. Hatcher, Douglas J. Howe, Bjarni Jónsson, Hans-Jörg Kreowski, Giancarlo Mauri, Ali Mili, Michael Mislove, Ugo Montanari, Peter D. Mosses, Istvan Németi, Don Pigozzi, Don Sannella, R.K. Shyamasundar, John Staples, Andrzej Tarlecki, Frits Vaandrager, Paulo A.S. Veloso, Eric Wagner, Bob Walters, Martin Wirsing.

Invited Speakers

To achieve the goal of the conference we aim to provide a forum in which leading researchers in mathematics, computer science, and software development, will come together to identify algebraic methodologies that are applicable as viable alternatives to the present software development approaches and to discuss the appropriateness of such alternatives with a view to implementation. Invited speakers, with a tentative indication of respective topics, include:

Hubert Comon	(LRI, Orsay)	: Constraint Solving in Term Algebras
Rob van Glabbeek	(University of Stanford)	: Process Algebra
Nicolas Halbwachs	(INPG, Grenoble)	: Reactive Program Synthesis
Mike Johnson	(Macquarie Univ., Sydney)	: Category Theory for Software Engineering
Giorgio Levi	(University of Pisa)	: Logic Programming
Roger D. Maddux	(Iowa State University)	: Relation Algebra

Submissions

Talks reporting research in algebra and logic, suitable as a foundation for software technology, as well as software technologies developed by means of algebraic methodologies, are welcome. Demonstrations of systems showing the improved effectiveness of software developed on a mathematical basis (with or without a talk) will also be considered. To be more precise the organizers wish contributions on, but not limited to, algebraic methods for language design and compiler construction (e.g. for Algol-like programming languages), algebraic methodology for software engineering, algebraic specifications and algorithms to automatically build programs from such specifications, extraction of programs from constructive proofs, categorical, algebraic and logic programming, deductive databases, query language design, algebraic specification of concurrent systems, distributed operating systems, reactive systems, practical techniques and examples for verification of program and/or specification properties. We invite you to submit a two-page abstract (including a few citations of relevant work) of your talk to

AMAST Conference, University of Twente, Fac. Informatica, Att. Mrs. Y. Rokker,
P.O. Box 217, NL-7500AE Enschede, The Netherlands (phone: + 31 53 893701)

Four-page abbreviated papers of the talks presented at the conference, together with the invited talks, will be collected in the participants' edition of the proceedings, which will be available to the attendees upon their arrival in Twente. The authors are expected to bring full versions of their papers at the conference, for further review and inclusion in the AMAST'93 Proceedings, to be published by Springer-Verlag, London. A special issue of *Theoretical Computer Science* will be dedicated to this conference; participants will be invited to submit their full paper for possible publication in this journal.

Important Due Dates

- Two page abstract submission by December 15, 1992.
- Notification of acceptance by March 31, 1993.
- Camera-ready four-page paper to appear in participants' proceedings by May 1, 1993.
- Full paper for AMAST'93 Proceedings, at the conference.

Further information can be obtained from:

In Canada:	*In Europe*:	*In U.S.A*:
V.S. Alagar	Charles Rattray	Teodor Rus
Concordia University	University of Stirling	University of Iowa
Dept. of Computer Science	Dept. of Mathematics	Dept. of Computer Science
1455 De Maisonneuve Blvd. West	and Computing Science	Iowa City
Montreal, Quebec H3G 1M8	Stirling, Scotland, FK9 4LA	IA 52242
Canada	Great Britain	USA
phone: + 1 514 8483022	phone: + 44 786 73171	phone: + 1 319 3350742
fax : + 1 514 8482830	fax : + 44 786 64551	fax : + 1 319 3350627
e-mail: alagar@concour.cs.concordia.ca	e-mail: cr@cs.stir.ac.uk	e-mail: rus@cs.uiowa.edu

FORTHCOMING EVENTS

1992

Date: July 27 - 29
Title: **International Symposium on Symbolic Algebraic Computation**
Acronym: ISSAC.
Location: Berkeley, California, USA.
Sponsor: SIGSAM.
Contact: Erich Kaltolfen, Rensselaer Polytech. Inst., Dept. of Comp. Sci., Troy, NY 12180; (518) 276-6907.
Email: kaltofen@cs.rpi.edu

Date: July 27 - 29
Title: **5th Annual Workshop on Computational Learning Theory**
Location: Pittsburgh, Penn., USA.
Sponsor: SIGACT and SIGART.
Contact: Robert Daley, Univ. of Pittsburgh, Dept. of Comp. Sci., Pittsburgh, PA 15260; Tel: (412) 624-5930.
Email: daley@cs.pitt.edu

Date: July 28 - August 9
Title: **International Summer School on Program Design Calculi**
Location: Marktoberdorf, Germany.
Contact: Institut für Informatik, Technische Universität München, Summer School, PO Box 202420, W-8000 München 2, Germany.

Date: August 10 - 12
Title: **Eleventh Annual ACM Symposium on Principles of Distributed Computing**
Location: Vancouver, British Columbia, Canada.
Acronym: PODC '92.
Sponsor: ACM.
Contact: Maurice Herlihy, Digital Equipment Corporation, Cambridge Research Laboratory, One Kendall Square, Cambridge, MA 02139.
Email: herlihy@crl.dec.com

Date: August 17 - 21
Title: **21st International Conference on Parallel Processing**
Contact: T. Feng, E.E. East Bldg., The Pennsylvania State Univ., University Park, PA 16802; tel: (814) 863-1469; fax: (814) 865-7065.
Email: tfeng@ecl.psu.edu

Date: August 24 - 27
Title: **Third International Conference on Concurrency Theory**
Location: Stony Brook, New York, USA.
Acronym: CONCUR '92.
Contact: S.A. Smolka, Department of Computer Science, SUNY at Stony Brook, Stony Brook, NY 11794, USA, tel: +1 516 632 8453, fax: +1 516 632 8334.
Email: sas@cs.sunysb.edu

Date: August 24 - 28
Title: **Seventeenth International Symposium on Mathematical Foundations of Computer Science**
Location: Prague, Czechoslovakia.
Acronym: MFCS'92.
Contact: MFCS'92 OC Secretary, Milena Zeithamlová, Agency Action M, Kazasská 1426, 101 00 Praha 10, Czechoslovakia.

Date: August 26 - 28
Title: **Fourth International Symposium on Programming Language Implementation and Logic Programming**
Location: Leuven, Belgium.
Acronym: PLILP 92.
Contact: Maurice Bruynooghe, Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium.
Email: maurice@cs.kuleuven.ac.be

Date: September 13 - 17
Title: **Software Eng. Standards Symp.**
Acronym: SESS
Location: Brighton, UK.
Contact: Takis Katsoulakos, Lloyd's Register, Lloyd's Register House, 29 Wellesley Rd., Croydon, CR0 2AJ, UK, 'phone (081) 681-4774.

Date: August 30 - September 5
Title: **The VIth BANFF Higher Order Workshop**
Location: The Banff Centre, Banff, Alberta, Canada
Contact: Graham Birtwistle, Department of Computer Science, University of Calgary, Calgary T2N 1N4, Canada.
Email: graham@cpsc.ucalgary.ca

Date: August 31 - September 2
Title: **International Workshop on Hardware-Software Codesign**
Location: Estes Park, Co., USA.
Sponsor: SIGDA, SIGSOFT, IEEE CS, and IEEE C&SC.
Contact: Joanne Degroat, Ohio State Univ., 205 Neil Ave., Columbus, OH 43210; Tel: (614) 292-2439.
Email: degroat@ee.eng.ohiostate.edu.

Date: September 2 - 4
Title: **3rd International Workshop on VLSI for Artificial Intelligence and Neural Networks**
Location: Oxford, England.
Sponsor: Univ. of Oxford.
Contact: Jose G. Delgado-Frias, Dept. of Electrical Engineering, SUNY@Binghamton, NY 13902-6000; Tel: (607) 777-4806; Fax: (607) 777-4822.
Email: delgado@bingvaxu.cc.binghamton.edu.

Date: September 2 - 4
Title: **Third International Conference on Algebraic and Logic Programming**
Location: Pisa, Italy.
Contact: Hélène Kirchner, CRIN & INRIA-Lorraine, BP 239, Campus Scientifique, 54506, Vandoeuvre-lès Nancy Cedex, France.
Email: hkircher@loria.crin.fr
Or: Giorgio Levi, Dipartimento di Informatica, Università di Pisa, Corso Italia 40, 56125 Pisa, Italy.
Email: levi@dipisa.di.unipi.it

Date: September 7 - 8
Title: **2nd European Workshop on Software Process Technology**
Location: Trondheim, Norway.
Sponsor: AFCET, AICA, Norwegian Computer Society, et al.
Contact: Jean Claude Derniame, Centre de Recherche en Informatique de Nancy, CRIN, Campus Scientifique, B.P.O. 239, F-54506 Vandoeuvre Les Nancy, France; Tel: +33 83.413052; Fax: +33 83.413079.
Email: derniame@loria.crin.fr

Date: September 7 - 11
Title: **12th World Computer Congress, IFIP Congress 92: "From Research to Practice"**
Location: Madrid, Spain.
Sponsor: International Federation for Information Processing (ISIP).
Local Organiser:
Grupo Geyseco, IFIP'92, Mauricio Legendre 4, 8th Floor G, E-28046, Madrid, Spain. Fax: (+34-1) 3234936.
Email: ifip92@dit.upm.es.

Date: 7 - 11 September
Title: **Second International Logic Programming Summer School**
Acronym: LPSS'92
Location: Zurich, Switzerland
Contact: Gerard Comyn, ECRC, Arabellastr. 17, D-8000 Munich 81, fax +49-89-926 99 1 170.
Email: gerard@ecrc.de
Or: Norbert E. Fuchs, Institut fuer Informatik, University of Zurich, CH-8057 Zurich, fax +41-1-363 00 35.
Email: fuchs@ifi.unizh.ch

Acronym: CSL '92.
Contact: Prof. Dr. E. Börger/Dr. S. Martini, CSL'92, Dipartimento di Informatica, Università di Pisa, Corso Italia 40, I-56125 PISA, Italy, fax: +39-50-510226.
Email: csl92@di.unipi.it

Date: **September 14 - 16**
Title: **3rd IFIP Working Conference on Dependable Computing for Critical Applications**
Acronym: DCCA-3.
Location: Mondello (Palermo), Sicily, Italy.
Sponsor: IFIP Working Group 10.4.
Contact: Luca Simoncini, Dipartimento Di Ingegneria dell'informazione, Univ. of Pisa, Diotisalvi 2, 56100 Pisa, Italy; +(39) 50 593443 or 550100; fax: +(39) 554342.
Email: simon@icnucevm.cnuce.cnr.it.

Date: **September 20 - 23**
Title: **Seventh Knowledge-Based Software Eng. Conf.**
Acronym: KBSE 92.
Location: Tysons Corner, Va., USA.
Sponsor: Rome Lab.
Contact: Lewis Johnson, USC Information Sciences Inst., 4676 Admiralty Way, Marina del Rey, CA 90292-6695, tel: (310) 822-1511, fax: (310) 823-6714.
Or: Barbara Radzisz, Data and Analysis Center for Software, PO Box 120, Utica, NY 13503, tel: (315) 336-0937.
Email: kbse7-request@cs.rpi.edu

Date: **September 21 - October 2**
Title: **International Summer School on Constructive Algorithmics**
Location: Hollum, Ameland, The Netherlands.
Contact: Summer School on Constructive Algorithmics, Vakgroep Informatica, Rijks Universiteit Utrecht, PO BOX 80.089, 3508 TB Utrecht, The Netherlands.
Or: Dr Erik Meijer, RU Utrecht, tel. +31 30-533989, fax. +31 30-513791.
Email: stop@cs.ruu.nl

Date: **September 22 - 24**
Title: **Information Systems Developers Workbench Methodologies, Techniques Tools & Procedures**
Location: Gdansk, Poland.
Contact: Stanislaw Wrycza, The Third Internation Conference on Information Systems Developers Workbench, University of Gdansk, Department of Information Systems, Armii Krajowej 119/121, 81-824 Sopot, Poland, tel: (48 58) 51-00-61 ext. 400, fax: (+ 48 58) 52-03-11, (+ 48 58) 52-22-12.

Date: **September 23 - 25**
Title: **5th International Conference on Putting Into Practice Methods and Tools for Information System Design**
Location: Nantes, France.
Contact: Henri Habrias, Liana, IUT, 3 rue Ml Joffre 44041 Nantes Cedex 01 (France); 'phone: (33) 40 30 50 56; fax: (33) 40 30 60 01.
Email: habrias@naiut.dnet@ciripa.circe.fr.
Paper Submission Details:
Submit five copies of their papers (15 pages maximum, double-spaced, (English or French) to Henri Habrias.

Date: **September 24 - 25**
Title: **International Workshop on Object Orientation in Operating Systems**
Acronym: IWOOOs '92.
Location: Paris, France.
Sponsor: Inst. National Recherche en Informatique et Automatique, INRIA, IEEE Technical workshop on Operating Systems and Application Environments.
Contact: Roy Campbell, Univ. of Illinois, Dept. of Comp. Sci., 2413 Digital Lab, 1304 W. Springfield Ave., Urbane, IL 61801; Tel: (217) 333-3328.
Email: roy@uiuc.edu.

Date: **September 28 - October 2**
Title: Computer Science Logic
Location: San Miniato (Pisa), Italy.

Location: Boston, Mass., USA.
Sponsor: SIGARCH, SIGOPS and SIGPLAN.
Contact: Barry Flahive, Hewlett-Packard-Apollo Systems, 300 Apollo Drive, MS CHR 02 DE, Chelmsford, MA 01824; 'phone (508) 256-6600.
Email: flahive@apollo.hp.com

Date: **September 29 - October 1**
Title: **16th International Symposium on Computer Performance Modelling, Measurement and Evaluation**
Location: Rome, Italy.
Acronym: Performance '92.
Paper Submission Details and Contact:
Submit six copies of full paper (20 double spaced pages max.) to (North America) S.S. Lavenberg. IBM TJ Watson Research Ctr., P.O. Box 704, Yorktown Heights, NY 10598.
Email: sslaven@watson.ibm.com.
Or: (Europe and all others): G. Iazeolla, II Univ. di Roma, Dipart. di Ingegneria Ellettronica, via dela Ricera Scientifica, I-00173 Roma, Italy.
Email: iazeolla@irmias.bitnet

Date: **September 30 - October 2**
Title: **International Workshop on Hardware-Software Codesign**
Location: Ohio, USA.
Co Sponsor: SIGDA, SIGSOFT, IEEE-CS and IEEE-C&CS.
Contact: Joanne Degroat, Ohio State Univ., 205 Neil Ave., Columbus, OH 43210.
Email: degroat@ee.eng.ohio-state.edu

Date: October 5 - 7
Title: 2nd International Conference on Software Quality Control
Location: Research Triangle Park, N.C., USA.
Sponsor: American Society for Quality Control Software Division.
Contact: Sue McGrath, SAS Inst., SAS Campus Dr., Cary, NC 27513, USA.
Email: sassam@dev.sas.com *or* John E. Lowe, Litton Computer Services, 4020 Executive Dr., Dayton, OH 45430, USA; Tel: (513) 429-6458.

Date: October 5 - 7
Title: **6th SEI Conference on Software Engineering and 11th SEI Educator Development Workshop**
Acronym: SCEE 92.
Location: San Diego, California, USA.
Sponsor: SEI and IEEE-CS in coop. SIGCSE, SIGSOFT.
Contact: Carol Sledge, Software Engineering Institute, Rm 4206, 4500 5th Ave., Pittsburgh, PA 15213; 'phone (412) 268-7708.
Email: cas@sei.cmu.edu

Date: October 5 - 7
Title: **11th Symposium on Reliable Distributed Systems**
Location: Houston, Texas, USA.
Sponsors: IEEE Computer Society TC on Distributed Processing, IEEE Computer Society TC on Fault-Tolerant Computing, IFIP WG 10.4 on Dependable Computing.
Contact: Prof. Kishor S. Trivedi, Department of Electrical Engineering, Duke University, Durham, N.C. 27706 USA, tel: (919) 660-5269.
Email: kst@egr.duke.edu
Or: Dr. Edgar Nett, GMD, P.O. Box 1316, Schloss Birlinghoven, D-5205 Sankt Augustin 1, Germany, tel: (49) 2241-142311.
Email: nett@gmdzi.gmd.de

Date: October 7 - 9
Title: **Third International Symposium on Software Reliability Engineering**
Acronym: ISSRE-92.
Location: North Carolina, USA.
Contact: Mladen A. Vouk, Computer Science Department, Box 8206, North Carolina State University, Raleigh, NC 27695-8206, Tel: (919)-515-7886, Fax: (919)-515-5839.
Email: vouk@adm.csc.ncsu.edu

Date: **October 12 - 15**
Title: **5th Architectural Support for Programming Languages and Systems**
Acronym: ASPLOS V.

Email: eoog@cui.unige.ch

Date: October 15 - 18
Title: **Seventh International Software Process Workshop**
Location: Yountville, Calif., USA.
Sponsor: Rocky Mountain Inst, of Software Eng.
Contact: Ian Thomas, Software Design and Analysis, 444 Castro St., Suite 413, Mountain View, CA 94041, phone (415) 694-1464.

Date: October 18 - 22
Title: **Object Oriented Programaming Systems, Languages, Architectures**
Acronym: OOPSLA '92.
Location: Vancouver, Canada.
Sponsor: SIGPLAN.
Contact: John Pugh, School of Comp. Sci., Carlton Univ., Colonel By Drive, Ottowa, Ont., Canada K1S 5B6; 'phone (613) 788-4330.
Email: john_pugh@Carleton.ca

Date: October 19 - 23
Title: 4th International Workshop on Foundations of Models and Languages for Data and Objects: Modelling Database Dynamics
Location: Volkse, Germany.
Contact: Udo Lipeck, Institut für Informatik, Universität Hannover, Lange Laube 22, D-W 3000 Hannover 1, Germany, tel: (++4910) 511-762-4950.
Email: ul@informatik.uni-hannover.[dbp.]de

Date: October 25 - 27
Title: **1992 Symposium on Foundations of Computer Science**
Location: Pittsburgh, Pennsylvania, USA.
Acronym: FOCS'92.
Contact: Gary Miller, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213-3890, USA.
Email: glmiller@cs.cmu.edu

Date: October 26 - 29
Title: **Fifth Workshop on Software Reuse**
Location: Palo Alto, California, USA.
Contact: Martin Griss, Hewlett-Packard Labs, 1501 Page Mill Rd., Palo Alto, CA 94304-1126, tel: (415) 857-8715, fax: (415) 857-8526.
Email: griss@hpl.hp.com

Date: October 26 - 29
Title: **3rd International Conference on Principle of Knowledge Representation and Reasoning**
Acronym: KR'92.
Location: Cambridge, Mass., USA.
Sponsor: AAAI and CSCSI in coop. w/ECCAI and IJCAII.
Contact: William Swartout, USC/Information Sciences Inst., 4676 Admiralty Way, Marina del Rey, CA 90292-6695; tel: (213) 822-1511; fax: (213) 823-6714.
Email: swartout@isi.edu

Date: October 26 - 30
Title: **Joint 9th WADT - Compass Workshop, Ninth Workshop on Specification of Abstract Data Types joint with the Fourth COMPASS Workshop**
Location: Caldes de Malavella, Spain.
Contact: Fernando Orejas, Dept. Lleng. i Sist. Inf., Univ. Polit. Catalunya, Pau Gargallo 5, 08028 Barcelona, Spain.
Email: orejas@lsi.upc.es

Date: October 28 - 30
Title: **3rd Eurographics Workshop on Object-Oriented Graphics**
Location: Switzerland.
Contact: Eurographics Workshop on Object-Oriented Graphics, Centre Univ. d'Informatique, 12 rue du Lac, CH-1207, Geneva, Switzerland, Tel: 41 (22) 787 65 85, Fax: 41 (22) 735 39 05.

Location: Nagoya, Japan.
Acronym: ISAAC '92.
Sponsors: ACM SIGACT, EATCS.
Contact: Kazuo Iwama, Dept. of Computer Science and Communication Engineering, Kyushu University, Hakozaki, Higashi-ku, Fukuoka, 812, Japan.
Email: iwama@cscs.kuyshu-u.ac.jp

Date: October 28 - 30
Title: **4th Italian Conference on Theoretical Computer Science**
Location: L'Aquila, Italy.
Sponsor: Italian Chapter of EATCS.
Contact: Prof. Marisa Venturini Zilli, Dip. di Matematica Pure ed Applicata, Universita' di L'Aquila, Via Vetolo, 67010 Coppito, L'Aquila, Italy.

Date: November 3 - 6
Title: **3rd European Conference on Software Quality**
Location: Madrid, Spain.
Sponsor: EOQ Software Committee.
Contact: Julio Gonzalez Sanz, AECC-CONGHRISA, c/o Velazquez 90 5, 28006 Madrid (SPAIN); tel: 34 1 575 25 80; fax: 34 1 577 38 74.

Date: November 9 - 12
Title: **Conference on Software Maintenance**
Sponsors: IEEE Computer Society - Technical Committee on Software Engineering.
Location: Orlando, Florida.
Contact: Dave Card, Computer Sciences Corporation, Systems Sciences Division, 4061 Powder Mill Road, Calverton, MD 20705, tel: (301) 572-3815, fax: (301) 572-7502.
Email: dc@sei.cmu.edu

Date: November 9 - 13
Title: **1992 Joint International Conference and Symposium on Logic Programming**
Location: Ramada Renaissance Hotel, Washington, D.C., USA.
Sponsor: The Association for Logic Programming.
Contact: Prof. Krzysztof R. Apt, Program Chair, CWI, Krulslaan 413, 1098 SJ Amsterdam, The Netherlands, fax: (+31-20) 592-4199.
Email: apt@cwl.nl

Date: November 23 - 25
Title: **European Symposium on Research in Computer Security**
Acronym: ESORICS 92.
Location: Toulouse, France.
Sponsor: AFCET.
Contact: AFCET, 156 Boulevard Péreire, 75017 Paris, France.
Email: deswarte@laas.fr

Date: December 1 - 4
Title: **Fourth IEEE Symposium on Parallel and Distributed Processing**
Location: Dallas, Texas, USA.
Sponsors: IEEE-Computer Society, ACM-SIGARCH, IEEE-Dallas Section.
Contact: Mikhail Atallah, Department of Computer Science, Purdue University, West Lafayette, IN 47907, tel: (317) 494-3653.
Email: mja@cs.purdue.edu

Date: December 9 - 11
Title: **5th Symposium on Software Development Environments**
Location: Washington DC, USA.
Contact: Herbert Weber, Fachbereich Informatik, Univ. of Dortmund, Baroperstrasse 301, 4600 Dortmund 50, Germany.

Date: December 14 - 15
Title: **7th Annual Z User Meeting**
Location: Department of Trade and Industry, Kingsgate House, 66-74 Victoria Street, London.
Contact: Secretary Z User Group Committee, Oxford University Computing Laboratory, Programming Research Group, 11 Keble Road, Oxford OX1 3QD. Tel: (0865) 272579.
Email: Joan.Arnold@comlab.oxford.ac.uk

Date: December 16 - 18
Title: **Third Annual International Symposium on Algorithms and Computation**

Contact: Program Chair of FASE: Marie-Claude Gaudel, Program Chair of CAAP: Jean-Pierre Jouannaud, TAPSOFT 93, LRI Batiment 490, Université Paris XI, 91405 Orsay Cedex, France, fax: 33 1 69 41 65 86
Email: tapsoft@lri.lri.fr or jouannaud@margaux.inria.fr
Or: AFCET, 156 Bd Péreire, 75017 Paris, fax: 33 1 42 67 93 12.

Date: December 18 - 20
Title: **Twelfth Conference on the Foundations of Software Technology and Theoretical Computer Science**
Location: New Delhi, India.
Contact: Prof. R.K. Shyamasundar, FST & TCS 12, Tata Institute of Fundamental Research, Bombay - 400 005, India, fax: 091-22-215 2110, telex: 011-83009.
Email: fsttcs@tifrvax.bitnet

1993

Date: **January 11 - 13**
Title: **The Twentieth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages**
Location: Charleston, South Carolina, USA.
Contact: (Program Chair) Susan L. Graham, Computer Science Division - EECS, 571 Evans Hall, University of California, Berkeley, CA 94720, USA, tel. (510) 642-2059.
Email: graham@cs.berkeley.edu
Or: (Local Arrangements Chair) Dee Medley, Augusta College, tel. (404) 737-1672.
Email: dmedley@uscn.uga.edu

Date: **February 25 - 27**
Title: **10th Symposium on Theoretical Aspects of Computer Science '93 Congress Centrum**
Location: Würzburg, Germany.
Acronym: STACS '92.
Sponors: GI, AFCET.
Contact: Prof. Dr. Klaus W. Wagner, Lehrstuhl für Theoretische Informatik, Universität Würzburg, Am Exerzierplatz 3, 8700 Würzburg, Germany, 'phone: +49-931-8878 10
Email: stacs@informatik.uni-wuerzburg.de

Date: **March 16 - 18**
Title: **International Conference on Typed Lambda Calculi and Applications**
Location: Utrecht, The Netherlands.
Acronym: TLCA.
Contact: Mr Frans Snijders, CWI, PO BOX 4079, 1009 AB Amsterdam, The Netherlands, tel. +31-20-5924171, fax. +31-20-5924199.
Email: franss@cwi.nl

Date: **March 24 - 26**
Title: **Second Int'l Workshop on Software Reusability**
Acronym: IWSR 93.
Location: Lucca, Italy.
Cosponsors: ACM SIGSoft et al.
Contact: Ruben Prieto-Diaz, Software Productivity Consortium, 2214 Rock Hill Rd., Herndon, VA 22070, tel: (703) 742-7107, fax: (703) 742-7200.

Date: **April 13 - 16**
Title: **Seventh Int'l Parallel Processing Symp.**
Acronym: IPPS 93
Location: Newport Beach, California, USA.
Cosponsor: ACM SIGArch.
Contact: Viktor K. Prasanna, EE Systems Dept., EEB 244, Univ. of Southern California, Los Angeles, CA 90089-2562, fax: (213) 740-4449.
Email: ipps93@halcyon.usc.edu

Date: **April 13 - 17**
Title: **TAPSOFT 93 (CAAP FASE Advanced Seminar)**
Location: Orsay, France.
Acronym: TAPSOFT 93.
Sponsors: AFCET, EATCS.

Date:	June 16 - 18
Title:	Fifth International Conference on Rewriting Techniques and Applications
Location:	Montreal, Canada.
Acronym:	RTA-93.
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.
Email:	Claude.Kirchner@loria.fr
Or:	Mitsuhiro Okada, RTA-93, Department of Computer Science, Concordia University, H3G1M8 Montreal, Quebec, Canada, tel. (1) (514) 848 30 48, fax. (1) (514) 848 28 30.
Email:	RTA93@concour.cs.concordia.ca

Date:	June 16 - 18
Title:	5th International Conference on Software Engineering and Knowledge Engineering
Acronym:	SEKE'93.
Location:	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.
Email:	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.
Email:	chang@stc.lockheed.com

Date:	June 21 - 25
Title:	The 14th International Conference on Application and Theory of Petri Nets
Location:	Bismarck Hotel, Chicago, USA.
Acronym:	Petri Nets'93.
Contact:	Prof. T. Murata, Dept. of EECS (m/c 154), Univ. of Illinois at Chicago (UIC), P.O. Box 4348, Chicago, IL 60680, USA.
Email:	pn93@uicbert.eecs.uic.edu

Date:	June 22 - 25
Title:	Third International Conference on Algebraic Methodology and Software Technology
Location:	University of Twente, Enschede, The Netherlands.
Acronym:	AMAST.
Abstract to:	AMAST Conference, University of Twente, Fac. Informatica, PO BOX 217, NL-7500AE Enschede, The Netherlands.
Or Canada:	V.S. Alagar, Concordia University, Dept. of Computer Science, 1455 De Maisonneuve Blvd. West, Montreal, Quebec H3G 1M8, Canada, tel. +1 514 8483022, fax. +1 514 8483000.
Email:	alagar@concour.cs.concordia.ca
Or Europe:	Charles Rattray, University of Stirling, Dept. of Mathematics and Computing Science, Stirling, Scotland, FK9 4LA, Great Britain, tel. +44 786 73171, fax. +44 786 64551.
Email:	cr@cs.stir.ac.uk
Or USA:	Teodor Rus, University of Iowa, Dept. of Computer Science, Iowa City, IA 52242, USA, tel. +1 319 3350742, fax. +1 319 3350627.
Email:	rus@cs.uiowa.edu

Date:	June 28 - 30
Title:	2nd International Workshop Logic Programming and Non-Monotonic Reasoning
Acronym:	LP&NMR-93.
Location:	Lisbon, Portugal.
Contact:	Anil Nerode, Mathematical Sciences Institute, Cornell Univ., Ithaca, NY 14853.

Date:	July 5 - 9
Title:	20th International Colloquium on Automata, Languages, and Programming
Location:	Lund, Sweden.
Acronym:	ICALP'93.
Contact:	Prof. Rolf Karlsson, Department of Computer Science, Lund University, S-221 00 Lund, Sweden.
Email:	icalp93@dna.lth.se

Date:	April 19 - 23
Title:	Industrial Strength Formal Methods
Location:	Odense Technical College, Denmark.
Acronym:	FME'93.
Contact:	Programme Chairman, Jim C.P. Woodcock, Oxford University Computing Laboratory, Programming Research Group, 11 Keble Road, Oxford OX1 3QD, UK, tel. +44 865 272576, fax. +44 865 273839.
Email:	jimw@prg.ox.ac.uk
Or:	Organising Chairman, Peter Gorm Larsen, The Institute of Applied Computer Science (IFAD), Forskerparken 10, DK-5230 Odense M, Denmark, tel. +45 65 93 23 00, fax. +45 65 93 29 99.
Email:	peter@ifad.dk

Date:	April 19 - 23
Title:	9th Internationcal Conference on Data and Engineering
Location:	Vienna, Austria.
Contact:	Eruch J. Neuhold, GMD-IPSI, Dolivostrasse 15, D-6100 Darmstadt, Germany; tel: (+49) 6151 869 803.
Email:	darmstadt.gmd.de

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

Date:	May 17 - 21
Title:	15th International Conference on Software Engineering
Location:	Baltimore, Maryland, USA.
Contact:	Victor R. Basili, Department of Computer Science, University of Maryland, College Park, Maryland 20742, USA; tel: (301) 405-2668.
Email:	basili@cs.umd.edu

Date:	May 25 - 28
Title:	The 13th International Conference on Distributed Computing Systems
Sponsor:	IEEE Computer Society.
Location:	Pittsburgh Hilton, Pittsburgh, Pennyslvania, USA.
Contact:	Benjamin W. (Ben) Wah, Coordinated Science Laboratory, University of Illinois, MC228, 1101 W. Springfield Avenue, Urbana, IL 61801-3082, tel: (217) 333-3516; fax: (217) 244-7175.
Email:	b-wah@uiuc.edu

Date:	June 14 - 17
Title:	The Fifth Asian Logic Conference
Location:	National University of Singapore, Republic of Singapore.
Contact:	The 5th ALC, Department of Mathematics, National University of Singapore, Singapore 0511, Republic of Singapore.
Email:	matlogic@nuscc.nus.sg OR matlogic@nusvm.bitnet

Date:	June 15 - 18
Title:	7th International Symposium on Methodologies for Intelligent Systems
Location:	Trondheim, Norway.
Contact:	Jan Kmorwski, Univ. of Trondheim, Norwegian Institute of Technology, Dept. EE and Comp. Sci, N-7034 Trondheim, Norway.
Email:	janko@idt.unit.no
Or:	Zbigniew W. Ras, UNC-Charlotte, Dept. of Comp. Sci., Charlotte, NC 28223.

Date:	**August 23 - 27**
Title:	**Fundamentals of Computation Theory**
Location:	Szeged, Hungary.
Acronym:	FCT'93.
Contact:	T. Gaizer or J. Virágh, FCT'93 Bloyai Institute, A. József University, 6721 Szeged, Aradi v. tere 1., Hungary, fax 36-62-12292.
Email:	h754esi@ella.hu, h1299gai@ella.hu, J68A004@HUSZEG11