FACS FACTS

The Newsletter of the BCS Formal Aspects of Computing Science SIG

Series II Vol 3 No 1 **May '92.**

Editor Jawed Siddiqi

This quarterly issue of FACS FACTS contains information on some formal methods tools. It includes overviews of the Genesis System and the Genesis Z tool. In addition there is a demonstration of how predicate calculus proofs can be simplified using a high order logic proof tool.

If you want to add a little sparkle to your life you might want to try Petri Nets 1992, this year being held at Sheffield. The conference programme and the workshop announcement on Partial Orders are enclosed in your newsletter.

We are hoping to carry further information about formal methods tools, so contributions would be very welcome.

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@cik.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

BCS–FACS Christmas Meeting on Domain Theory

City University, 19–20 December 1991

1 Introduction

Last year, the BCS–FACS annual Christmas meeting covered the topic of Domain Theory. The two day format included introductory and intermediate level tutorials, and more advanced lectures; there was something for everybody. Below, I attempt to give a flavour of the presentations, from my own point of view as an attendee. This summary is necessarily incomplete: it is impossible to compress two days of dense mathematics into a few pages. So at the end there is a list of 'further reading', where more on the various topics can be found.

2 Introduction to Denotational Semantics and Domain Theory — Samson Abramsky (Imperial College)

This introductory tutorial motivated giving formal mathematical meanings to programs, introduced denotational semantics as one such approach, and gave an justification for why domains are a necessary component of the process.

A meaning can be given to a program by interpreting it as a mathematical object. With denotational semantics, the mathematical structure used is an algebra, and each language construct is given an interpretation as an operator in the algebra. The programming language construct 'denotes' the mathematical object.

For a simple imperative language, the algebra can be one of sets and functions. For example, expressions can be interpreted as denoting numbers (the appropriate value of the expression) and commands as denoting state transitions functions. The first problem with this simple approach occurs with the 'while' command, which has a recursive definition with the possibility of non-termination. A partial function approach can be used to solve this problem: 'before states' corresponding to non-terminating loops are not in the domain of the partial state transition function. But eventually, as the language being defined gets more complicated (for example, procedures as parameters to procedures) a richer mathematical model is needed to represent it: domain theory.

Consideration of the desirable properties of a denotational specification language (such as existence of fixed points, and handling of non-termination) leads to an axiomatic definition of the properties required of domains. For example, non-termination is denoted by partial objects, with 'more defined' being formalized by partial ordering, and 'fully defined' by the limit. So, domains are complete partial orders (partial orders that have a least upper bound). Using set theory allows 'too many' functions to be represented: we want only the computable functions between domains, those that preserve the structure of the domains. 'Monotonic' functions preserve the order (which implies the Halting Problem cannot be solved), and 'continuous' functions preserve limits (which implies only a finite amount of information about the input data can be used in the computation).

3 Information Systems — Steve Vickers (Imperial College)

This tutorial presented Scott Information Systems as one approach to solving domain equations (another approach uses category theory).

A Scott Information System (SIS) is a domain presented from an informational point of view. It comprises a set of 'tokens' (units of information), a consistency predicate (to say whether tokens have been combined consistently) and an entailment rule (that says whether one piece of information automatically entails another). These components must satisfy various properties.

The set of all 'ideals' of a SIS form a domain. Although not all domains can be constructed this way, enough 'useful' ones can; the advantage is that when sets of simultaneous equations involving domains are translated into SIS terms, they can be solved more easily.

4 Examples of Domain Equations — Samson Abramsky (Imperial College)

This tutorial presented a number of domain equations for natural numbers, lists, and functions, and discussed their solutions.

Similar data structures can come with a variety of evaluation strategies. For example, lists can be 'strict cons' (eager, finite lists), 'lazy cons' (finite and infinite lists), and 'head-strict tail-lazy cons' (infinite lists). These different computational models are captured by different domain equations.

More complex domains occur when considering 'self-application' in the λ-calculus: terms of the form $x\ x$. Considering the types, x must have a type D and also type $D \to D$. The structure of non-trivial domains that satisfy this requirement is rather complex.

5 The Monadic Approach to Denotational Semantics — Andy Pitts (University of Cambridge)

This lecture presented recent approaches for formalizing the language used to define the denotational semantics of another language (the metalanguage) and for improving the structuring of the definition.

Given a domain, a new domain modelling a new kind of computation can be constructed from it. For example, domains supporting side-effects, exceptions and non-determinism can be constructed from a simpler domain in a systematic manner. It has been noted that all these constructions have the same form, that of a strong monad on the category of semantic domains.

This research is continuing, with the aim of providing a structured way of adding new language features.

6 **An Introduction to Concurrency — Steve Schneider (University of Oxford)**

This tutorial presented the denotational semantics of the parallel language CSP (Communicating Sequential Processes).

The denotation of programs in CSP is the set of all possible observations. In the simplest model, the 'traces' model, the observations are the communications between processes. However, this model is not rich enough to distinguish some undesirable programs from desirable ones: it does not handle deadlock properly. A more sophisticated model, the 'failures–divergences' model has to be used, where the communications processes refuse to engage in are also considered.

7 **Fixed Points without Completeness — Geoff Barrett (Inmos)**

This lecture presented the problem with an infinite traces model of CSP (as opposed to the standard, finite, traces model), and its resolution.

The failures–divergences model of CSP cannot model 'fairness' properties. An infinite traces model, dealing with infinitely non-deterministic processes, in needed. Unfortunately, this model does not have a *complete* partial order: it does not have upper bounds required by the domain axioms. A more sophisticated treatment is needed.

8 **Introduction to Powerdomains — Edmund Robinson (University of Sussex)**

This tutorial presented powerdomains, used in the denotation of concurrent languages (except for CSP, which has its own, hand-crafted domains).

Powerdomains model (finite) non-determinism. Informally, a non-deterministic command can be considered as a mapping from the before state to a *set* of after states: $S \rightarrow \mathcal{P}(S)$. Powerdomains are the domain-theoretic analogues of the sets of finite subsets of a set.

When reasoning about liveness, non-determinism is 'good': a larger set of possible outcomes means more good things might happen. When reasoning about safety, non-determinism is 'bad': a larger set of possible outcomes means less control over the process. Hence there are various different order relations necessary on powerdomains.

Recent work on powerdomains has been in the area of databases. A database can be considered as a set of records. When different records with identical fields are allowed, 'bag domains' can be used, but these are outside conventional domain theory.

9 Powerdomains and Predicate Transformers: a Topological Approach — Mike Smyth (Imperial College)

This lecture presented the connection between denotational semantics using power domains, and axiomatic semantics using Dijkstra's weakest preconditions, when viewed from a topological perspective.

10 Further Reading

The numbers before the references refer to the presentations where they were mentioned.

(2) S. Abramsky. The lazy λ-calculus. In D. Turner, editor, *Research Topics in Functional Programming*. Addison Wesley, 1990.

(8) M. Hennesy and G. D. Plotkin. Full abstraction for a simple parallel programming language. In J. Becvár, editor, *Proc. Mathematical Foundations of Computer Science*, volume 74 of *Lecture Notes in Computer Science*, pages 108–120. Springer-Verlag, 1979.

(5) E. Moggi. A Modular Approach to Denotational Semantics. preprint. 1990.

(5) E. Moggi. *Inf & Comp*. 93:55–92, 1991.

(4) L. C. Paulson. *Logic and Computation: Interactive Proof with Cambridge LCF*, volume 2 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1987.

(9) M. B. Smyth. Power domains and predicate transformers: a topological view. In P. Diaz, editor, *Proc. 10th International Colloquim on Automata, Languages and Programming*, volume 154 of *Lecture Notes in Computer Science*, pages 662–675. Springer-Verlag, 1983.

— R. D. Tennent. *Semantics of Programming Languages*. Prentice Hall, 1991.

(8) S. Vickers. *Topology via Logic*. Cambridge University Press.

Susan Stepney
Logica Cambridge

What are Formal Methods?

Formal methods are techniques based on the application of mathematical and logical concepts to computing. Most commonly the term is applied to system specification methods and notations such as Z (developed by the Programming Research Group at Oxford University) and VDM (the Vienna Development Method, developed at the IBM Vienna Laboratories).

The varying emphases and needs of different organisations have lead to the development of a number of methods and notations. Recognising this, IST has developed 'Genesis', a widely applicable, generic tool capable of supporting most formal methods. Genesis has already been successfully used by IST to create tools to support Z and VDM and can be tailored to support other approaches.

Why use Formal Specification?

Formal specifications, unlike those using English language descriptions and diagrams are precise and unambiguous. Formal specifications use the precision, conciseness and universality of mathematics to produce specifications which are demonstrably correct.

Users of formal methods produce rigorously checkable specifications, unlike those depending on (often ambiguous) English language descriptions or vaguely defined diagrammatic notations. The greater confidence in the correctness of designs that this produces has lead to an increasing interest in such methods, especially in the areas of Safety Critical and High Reliability Systems. The new MoD standard 00-55 on Safety Critical Systems specifically identifies formal methods as an essential part of the design of such software.

An extra benefit of formal methods is the increased visibility of the design. Incompleteness or inconsistency in user requirements can be identified earlier and dealt with then, rather than leaving unexamined areas to later stages of development where they can be much more costly to fix.

How do Formal Methods fit into the Development Life-Cycle?

Formal methods do not replace other methods, but they do provide a technique which builds upon the results of the initial stages of specification to give greater clarity to the design at an earlier stage.

This early clarity and preciseness helps to avoid the design errors which, if undetected, can have a calamitous impact on later development stages.

Support for Formal Methods

To make the best use of formal techniques, there is a clearly identified need for appropriate tools. The rigour demanded by these methods requires computing support if users are to realise their maximum potential.

Tools can provide aids to the creation, modification and maintenance of formal specifications; they should also provide support for verification of the correctness of designs (theorem proving).

What does Genesis do?

Genesis is a tool which can be configured to provide a support environment for most formal methods. Genesis has already been tailored to support VDM and Z (these two environments are available from IST as separate products).

Once configured for a particular method, the Genesis tools can be used to create, edit and browse through specifications using syntax directed editors appropriate to the method in question. The use of such an editor ensures that all input is syntactically correct as it is entered.

If a concrete syntax is defined for the method (ie an ASCII representation of the symbols it employs and rules for their use) then a parser can be produced which allows text files containing specifications to be imported and syntax checked. Once imported, specifications can be manipulated using the language editor.

To enable specifications to be displayed and printed in a clear and consistent style, Genesis allows the required layout of specifications to be defined. The editor automatically maintains the required format as specifications are entered and modified.

The development process involves demonstrating that the specification is:

correct - it correctly captures what was intended,

complete - it includes all the required functions, and

consistent - its parts do not contradict each other.

With a formal method, this process is aided by the existence of logic (inference) rules for proving theorems about the specification. Once configured, Genesis provides a proof editor which can be used interactively to prove theorems using the inference rules of the method. The user can 'help' the proof editor by suggesting possible classes of inference rules to apply. Theorems can be recorded and used to simplify later proofs.

The Components of Genesis

Genesis comprises three main components:

KENSHO a language for defining the user interface. Kensho takes ideas from LISP and LaTeX(the powerful layout and typesetting language) to provide a SunView-based window interface.

JADE a logic programming language for proving theorems. JADE is used to implement rules describing how statements about the system can be manipulated. This allows assertions about systems behavior to be proved correct.

An LL(1) parser generator for creating parsers for the languages supported by the tools.

A developer configuring Genesis for a particular method (referred to as a 'Meta User') uses these components to implement tools to support the method. Such tools provide the complete environment for the ordinary user of the method (the 'Application User'). At this level the details needed for KENSHO and JADE are invisible to the application user who communicates via the normal SunView mouse and keyboard interface.

Support for Genesis

A Genesis Meta User (a creator of a new support environment) needs a knowledge of several areas of computing. Naturally they will need an in depth knowledge of the method being implemented, but to make the best use of Genesis they will also need to know something of language definition (to make the best use of the parser generator) and proof theory and logic programming (to use JADE effectively).

IST's Formal Methods Group provide consultancy and support for Genesis users. Our expertise in Genesis (as its original developers and from using it to create support environments for Z and VDM) enable us to advise on its application and assist in the creation of new tools.

Our consultants have, between them, over 40 years of formal methods experience in industry covering a wide range of software engineering projects. We have a high level of expertise in methods, such as VDM, Z and CSP, as well as considerable experience in applying them to such areas as secure and high integrity systems.

From using formal methods on a variety of projects, we are aware of the need for care in their introduction and use. We can advise on migration to formal methods, their applicability to client's needs and project management for formal methods development.

We also provide training in formal methods, from the level of management overviews, through courses for project leaders and line managers, to in-depth technical courses on specific methods.

The Genesis Z Tool - An Overview

What is the Genesis Z Tool?

Genesis Z is a tool which facilitates the creation and verification of formal specifications using the Z notation. Genesis Z has been built on the foundations of Genesis which was designed to provide generic mechanisms for tools to support formal methods.

The Genesis Z Tool Components

The Specification Editor

The Genesis Z Tool Specification Editor allows users to create and modify Z specifications which are stored internally as Abstract Syntax Trees (AST's).

Specifications may be input in one of two ways:

- by using the syntax-directed editor
- by writing the specification in ASCII and parsing it into the editor

The syntax-directed editor is operated from menus. Users select meta-variables from the specification and expand each one until only terminals remain. Each choice for expansion appears in a menu and corresponds to a production rule in the grammar for the Z language. Because users can only choose a valid production, the syntactic correctness of the specification is assured.

ASCII text can be typed into a separate Text Window. A corresponding syntactic category is then selected and the ASCII text is parsed to produced the AST which is subsequently displayed on the Specification Canvas. A facility is also provided to unparse an AST back to its ASCII equivalent.

These methods are complementary and users may freely switch between them. Each specification may be stored as a file for subsequent examination and/or modification.

Because Z provides no mechanism for structuring text, a specification written in the Z language can often be difficult to assimilate. To overcome this deficiency, the Genesis Z Tool allows users to structure their specifications into *Chapters*.

Associated with each chapter is a list of names of the chapters it references; the specification editor has facilities to maintain and view these references. A chapter referenced within another chapter is considered to be part of that chapter and can be referred to within it. When a chapter is type checked, the type checker is initialised with type information stored with the chapters referenced.

Each chapter has a name associated with it. The full chapter name is the absolute pathname of the corresponding Unix file. The Genesis Z Tool provides a *nickname* facility which allows names to be declared which map onto full chapter names.

The Mathematical Toolkit

The Genesis Z Tool includes the Z Mathematical Toolkit as defined in Spivey's "The Z Notation: A Reference Manual". The toolkit has been divided into chapters which may be referenced by users within their specifications.

The Type Checker

The type checker uses a set of rules to verify that a Z specification is correctly typed. Type checking is carried out with respect to the scope of references of a chapter; i.e. the chapter itself, the chapters it references and all chapters referenced by those chapters and so on. If any errors are detected, the location and nature of those errors will be reported.

The Tactical Proof System

The Tactical Proof System consists of a Tactical Logic Machine, a Proof Window and a Secure Lisp Reader. The Proof Window displays the goal to be proved. The user selects a proof goal and applies a tactic. If successful, this results in removing the original goal and replacing it with the new proof goals needed to complete the proof. The original goal is proved when no more proof goals remain.

The tactic applications are supplied by the user as Lisp expressions which are read from a Lisp Window. These expressions are read in via the Secure Lisp Reader which ensures that no illegal expressions can be executed which may subvert the integrity of the proof state.

The tactics provided implement inference rules and there is a term rewriting capability. Term rewritings are called *conversions*. The conversions provided implement basic simplifications and the ability to rewrite with respect to an equation or logical equivalence from the goal.

The Tactical Logic Machine contains the rules necessary to carry out a proof. A number of basic tactics and conversions are supplied. New tactics and conversions can be built up from those provided to capture re-usable proof techniques. The Genesis Z Tool does not allow proofs to be carried out on specifications which have not been type checked.

Each chapter may have proof-related information stored with it. This information is stored within *notes*. There are currently three kinds of notes:

Tactical program notes - which are source code files for tactical proof programs.

Demonstration notes - which record the proof trees generated by the tactical proof system.

Workspace object notes - which are used to store derived rules and may also be used to hold arbitarary user data.

Summary

The Genesis Z Tool has been designed to support the creation of specifications written in the Z language. The provision of a syntax-directed editor ensures that the production rules for the Z grammar are always obeyed. Users are not restricted to using the editor in syntax-directed mode; they may also enter the text in ASCII and then parse it into the specification.

A chapter mechanism allows large specifications to be split into more manageable parts. Chapters may also include references to other chapters. Notes can be used to associate proof-related information with each chapter. When a chapter is type checked, the system will report the exact nature and location of all type-related errors.

An extensible tactical proof system allows users to prove the validity of assertions made in their specifications.

Further Information

For further information please contact:

Karam Ashoo
Imperial Software Technology
3 Glisson Road
Cambridge CB1 2HA, England
Telephone: (0223) 462400
Fax: (0223) 462500
Electronic mail: ka@ist.co.uk

SIMPLIFYING PREDICATE CALCULUS PROOFS

Though HOL is a higher order logic proof tool, a significant part of the reasoning undertaken in industrial applications (especially if you are using it to support Z!) is essentially first order. The ease with which users can prove results in the first order predicate calculus is therefore significant to overall productivity.

Some of the innovations we have made in ICL HOL simplify these proofs, and make it easier for beginners to learn how to prove first order results. A new user who is familiar with first order logic should be able to sit down in front of ICL HOL and learn within five minutes how to use the tool to prove any (true) first order result. Though we don't guarantee the proof will be easy, use of the tool for these problems is easy, and the tool does help a lot in finding the proof.

The simplest proof strategy is to use the resolution based facilities recently developed for ICL HOL. These solve many simple pure predicate calculus results automatically. However, since some results may not be provable automatically (or may take too long), it is useful to have a good method to fall back on if completely automatic proof fails.

The "two tactic story" is a simple but effective method which can be used either to discharge completely a predicate calculus result, or to progress it until resolution is able to finish the job.

The first of the two tactics, "contr_tac", sets up the user for a classical `proof by contradiction', aggressively taking apart the goal and yielding a set of sub-goals each requiring proof of "F" from a set of contradictory assumptions. If the original goal is a propositional tautology "contr_tac" will complete the proof without further intervention by the user.

The second tactic, "list_spec_asm_tac" may then be used to specialise one of the assumptions of the current sub-goal to yield an explicit contradiction. All necessary inference other than the specialisation of universally quantified assumptions (viz: propositional reasoning and skolemisation of existential assumptions) is undertaken automatically, so that the user is only required to identify the necessary substitutions. In typical proof most of the top level sub-goals would be discharged after just one specialisation step.

The following example proof is the proof of proposition *11.71 from Principia Mathematica (after translation into our notation). The script is presented in ASCII as if it were Cambridge HOL (`?' means "there exists", `!' means "for all"), though the original proof script was in the extended character set used by ICL HOL. The result is automatically provable with the resolution facilities, but is used here to illustrate the two tactic method.

```
push_goal([],"
        (?z. p z) /\ (?z. q z) ==>
                ((!z. p z ==> r z) /\ (!z. q z ==> s z)
                <=>
                (!z w. p z /\ q w ==> r z /\ s w))
");

(* this sets up the goal *)

a contr_tac;

(* initiates proof by contradiction,
                        yielding four sub-goals *)

(* *** Goal "1" *** *)

(* 7 *) "p z"
(* 6 *) "q z'"
(* 5 *) "!z. p z ==> r z"
(* 4 *) "!z. q z ==> s z"
(* 3 *) "p z''"
(* 2 *) "q w"
(* 1 *) "~ r z''"

(* ?|- *) "F"

a (list_spec_asm_tac "!z. p z ==> r z" ["z''"]);
```

Current goal achieved, next goal is:

```
(* *** Goal "2" *** *)

(* 7 *) "p z"
(* 6 *) "q z'"
(* 5 *) "!z. p z => r z"
(* 4 *) "!z. q z => s z"
(* 3 *) "p z'"
(* 2 *) "q w"
(* 1 *) "~ s w"

(* ?|- *) "F"

a (list_spec_asm_tac "!z. q z => s z" ["w"]);
```

Tactic produced 0 sub-goals:
Current goal achieved, next goal is:

```
(* *** Goal "3" *** *)

(* 5 *) "p z"
(* 4 *) "q z'"
(* 3 *) "!z w. p z /\ q w => r z /\ s w"
(* 2 *) "p z''"
(* 1 *) "~ r z''"

(* ?|- *) "F"

a (list_spec_asm_tac
"!z w. p z /\ q w => r z /\ s w" ["z''","z'"]);
```

Tactic produced 0 sub-goals:
Current goal achieved, next goal is:

```
(* *** Goal "4" *** *)

(* 5 *) "p z"
(* 4 *) "q z'"
(* 3 *) "!z w. p z /\ q w => r z /\ s w"
(* 2 *) "q z''"
(* 1 *) "~ s z''"

(* ?|- *) "F"

a (list_spec_asm_tac
"!z w. p z /\ q w => r z /\ s w" ["z","z''"]);
```

Tactic produced 0 sub-goals:
Current and main goal achieved

The theorem may now be retrieved from the goal-package and stored in a theory or bound to an ML name.

The main merits of this "two tactic method" are that it is systematic and requires very little knowledge of proof machinery. You do need to understand the predicate calculus to be able to identify what to instantiate, but you won't be frustrated by not knowing what tactics/rules/conversions to use.

The approach described above covers only first order predicate calculus without equality. Good general methods adequate for most applications require the user to be familiar with only a small subset of the wide range of facilities available in ICL HOL. We are continuing work on reducing the knowledge required by the user of the system, and on clear and straightforward systematic approaches to proof with the ICL HOL system.

Though the proof above is in the HOL language rather than in Z, a similar proof style will suffice for pure predicate calculus results in Z.

PETRI NETS 1992

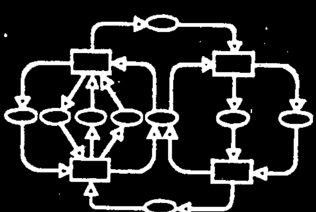
in Sheffield

THE WELCOMING CITY

13th International Conference on Application and Theory of Petri Nets
Wednesday 24 to Friday 26 June

Introductory and Advanced Tutorials
Monday 22 and Tuesday 23 June

FOR FURTHER DETAILS CONTACT:
Conference Services
Sheffield City Polytechnic
36 Collegiate Crescent
Sheffield S10 2BP
England
Tel: +44 742 532576/7
Fax: +44 742 532579



Hosted by
School of Computing and
Management Sciences

Sheffield City Polytechnic

Sponsored by BT

Conference Programme - Wednesday 24 to Friday 26 June

Wednesday 24 June

08.00 - 08.45	Registration at Sheffield City Polytechnic
08.45 - 09.00	Opening
09.00 - 10.00	**Guest Speaker - G Balbo** **Performance Issues in Parallel Programming**
10.00 - 10.30	Refreshments
10.30 - 12.00	Robert Shapiro (Meta Software Corporation, Massachusetts, USA) and Hartmann Genrich (GMD, Bonn, Germany) **Formal Verification of an Arbiter Cascade**
	Carlos A Heuser (Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil) and Gernot Richter (GMD, Bonn, Germany) **Constructs for Modelling Information Systems with Petri Nets**
	Søren Christensen (Aarhus University, Denmark) and Laure Petrucci (CEDRIC-IIE, Paris, France) **Towards a Modular Analysis of Coloured Petri Nets**
12.00 - 13.30	Lunch
13.30 - 15.00	Peter Kemper and Falko Bause (Universität Dortmund, Germany) **An Efficient Polynomial-Time Algorithm to Decide Liveness and Boundedness of Free-Choice Nets**
	Vanio M Savi and Xiao-lan Xie (INRIA, Metz, France) **Liveness and Boundedness Analysis for Petri Nets with Event Graph Modules**
	Kamel Barkaoui (CNAM, Paris, France) and Michel Minoux, (Université Pierre et Marie Curie, Paris, France) **A Polynomial-Time Graph Algorithm to Decide Liveness of Some Basic Classes of Bounded Petri Nets**
15.00 - 15.30	Refreshments
15.30 - 17.00	Susanna Donatelli and Matteo Sereno (Università di Torino, Italy) **On the Product Form Solution for Stochastic Petri Nets**
	Günter Klas (Siemens AG, München, Germany) **Hierarchical Solution of Generalised Stochastic Petri Nets by Means of Traffic Processes**
	Giorgio Bruno, Andrea Castella, Gianpaolo Macario, and Marco P Pescarmona (Politecnico di Torino, Italy) **Scheduling Hard Real Time Systems Using High-Level Petri Nets**

Wednesday 24 June - Social Programme

18.15	Coach leaves Novotel for visit to Chatsworth
19.00 - 20.30	Tour of Chatsworth House, home of Their Graces the Duke and Duchess of Devonshire
20.30 - 23.00	Dinner in the Carriage House Restaurant, Chatsworth

Thursday 25 June

09.00 - 10.00	**Guest Speaker - W Reisig** **Combining Petri Nets with Other Formal Methods**
10.00 - 10.30	Refreshments
10.30 - 12.00	**Project Presentations**
	Geoff Cutts (Sheffield City Polytechnic, UK) and Shaun Rattigan (Systec CS Limited, Sheffield, UK) **Using Petri Nets to Develop Programs for PLC Systems**
	Linda Wilkens, James Canning and Patrick Krolak (University of Massachusetts, Lowell, USA) **Modelling Fine Grain Computation via the Fusion of Two Extended Petri Nets**
	K Lemmer and E Schnieder (Technische Universität Braunschweig, Germany) **Modelling and Control of Complex Logistic Systems for Manufacturing**
	Annie Guevel (IXI, Toulouse, France) **Using Generalised Stochastic Petri Nets for Systems Assessment: Operational Availability and Exploitation Costs of a Constellation of Satellites**
12.00 - 13.30	Lunch
13.30 - 15.00	Carolyn Brown and Douglas Gurr (Aarhus University, Denmark) **Refinement and Simulation of Nets - A Categorical Characterisation**
	Johan Lilius (Helsinki University of Technology, Finland) **High-Level Nets and Linear Logic**
	Hans Fleischback (Universität Oldenburg, Germany) **P-Superfairness in Nets**
15.00 - 15.30	Refreshments
15.30	**Work in Progress** In this session the Conference Participants are invited to make short presentations of their ongoing work. Each presentation will be 10-15 minutes. Those who wish to contribute should contact the Programme Committee Chairman before or during the Conference.

Friday 26 June

09.00 - 10.00	**Guest Speaker - M Hennessy** **Action Refinement in Process Algebras**
10.00 - 10.30	Refreshments
10.30 - 12.00	C Autant and Ph Schnoebelen (Institut IMAG, Grenoble, France) **Place Bisimulations in Petri Nets** Kunihiko Hiraishi (Fujitsu Laboratories Limited, Numazu, Japan) **Construction of Petri Nets by Presenting Examples** A V Kovalyov (Byelorussian Academy of Sciences, Minsk, Russia) **Concurrency Relations and Safeness Checking for Petri Nets**
12.00 - 13.30	Lunch
13.30 - 14.30	**Project Presentations** J C Lloret, J L Roux, B Algayres and M Chamontin (Université Toulouse, France) **EVAL: A Petri Net Based Industrial Tool for System Modelling and Evaluation** William W McLendon Jr (Draper Laboratory, Massachusetts, USA) and Richard F Vidale (Boston University, Massachusetts, USA) **Analysis of an Ada System Using Coloured Petri Nets and Occurrence Graphs** Kimmo Varpaaniemi and Marko Rauhamaa (Helsinki University of Technology, Finland) **The Stubborn Set Method in Practice**
14.30 - 15.00	Refreshments
15.00 - 16.30	Jörg Desel (Technische Universität, München, Germany) **A Proof of the Rank Theorem for Extended Free Choice Nets** E Teruel (Universidad de Zaragoza, Spain), P Chrzastowski-Wachtel (Warsaw University, Poland), J M Colom and M Silva (Universidad de Zaragoza, Spain) **On Weighted T-Systems** Greg Findlow (Telecom Australia) **Obtaining Deadlock-Preserving Skeletons for Coloured Nets**
16.30	Closing session

Friday 26 June - Social Programme

19.30 - 24.00	Conference Dinner at the Cutlers' Hall

See you in Chicago for Petri Nets 1993

Conference Information

Venue
The Tutorials and Conference will take place in the New Teaching Block on the Collegiate site of Sheffield City Polytechnic (see map inside back cover).

Arrival and reception
Upon arrival in Sheffield, delegates should make their way to the Novotel to register and collect their Conference papers. The Petri Nets reception and information desk will be situated in the Huntsman suite of the hotel and will be open at the following times:

Sunday 21 June	14.00 - 21.00 (Tutorials and Conference)
Tuesday 23 June	14.00 - 21.00 (Conference)

Participants who arrive later than these times may collect their Conference papers the following morning from the reception and information desk at the Conference venue.

Contact address, telephone and fax numbers

Hotel	Novotel Sheffield Arundel Gate Sheffield S1 2PR Telephone +44 742 781781 Fax +44 742 787744
Conference office	Sheffield City Polytechnic 36 Collegiate Crescent Sheffield S10 2BP Telephone +44 742 532290 Fax +44 742 532290

Conference office
The Conference office will be located in the New Teaching Block. Once the Conference is underway, all enquiries should be directed to this office.

Office services
Telephone, fax and E-mail facilities will be available in the Conference office. The Novotel Business Centre offers photocopying, fax and credit card telephone facilities.

Conference proceedings
Registered delegates will receive a copy of the Conference proceedings upon arrival. Additional copies will be on sale at the registration desk and in the Conference office.

Fire regulations
You are asked to familiarise yourself with the action to be taken in case of fire or emergency, both at the hotel and the Conference venue.

Medical
In case of a medical emergency, contact the Conference office. The nearest accident/emergency unit is at the Royal Hallamshire Hospital on Glossop Road (see map inside back cover) telephone 766222.

Workshop Announcement

"What Good Are Partial Orders?"

June 22,1992, Sheffield (England)

Description

Partial order semantics is an alternative to interleaving semantics for the description of the behaviour of concurrent systems. Recent developments have moved from ideological reasoning to more objective investigations of the relative usefulnesses of both (or intermediate) approaches. The Esprit Basic Research Working Group 6067 CALIBAN (Causal Calculi Based on Nets) organises a one-day workshop devoted to this topic on June 22 in Sheffield (England).

This workshop takes place just prior to a 1-day Tutorial on the Relationship between Different Models of Concurrency (June 23, by Mogens Nielsen, Aarhus University), an the Annual Petri Net Conference (June 24–26), both also in Sheffield.

The workshop aims to provide a forum at which stock can be taken of the developments; at which the actual advantages (or disadvantages) of partial order semantics can be assessed; and at which further directions of research can be identified. It seemed appropriate to exploit the vicinity of the Petri Net Conference, since the potential usefulness of partial orders have frequently been stressed in net theory. However the themes of the workshop stretch across theories and are not limited to Petri nets.

The workshop is informal and will contain time for breaks and discussion.

Programme

June 22, 1992.

9:30 – 10:20 Jos Baeten, University of Technology Eindhoven:
What Good is Interleaving?

The talk will consider the relative merits and place of partial order semantics and interleaving semantics.

10:20 – 11:00 Marta Z. Kwiatkowska, University of Leicester:
Use of Order and Metric in Partial Order Semantics.

We consider Mazurkiewicz traces as a representation of concurrent behaviour. Two constructions of infinite traces are presented, one based on partial order and one on a metric, and their relationship discussed.

11:00 – 11:30 Coffee Break.

11:30 - 12:10 Javier Esparza, Universität Hildesheim:
Fast Model Checking Using Partial Order Semantics.

Partial order semantics can be used to palliate the state space explosion problem caused by the representation of concurrency by interleaving. We present a model checker based on the theory of nonsequential processes, and compare it with other ones existing in the literature.

12:10 - 12:50 Raymond Devillers, Université Libre de Bruxelles:
General Refinement and Recursion for the Petri Box Calculus.

New general definitions are given for the refinement and recursion operators in the calculus of Petri Boxes. It is shown that not only recursion, but also other operators such as sequence, choice and iteration can be viewed as based on refinement. Various structural properties of these operators can be deduced from a general property of (simultaneous) refinement. A partial order based denotational approach for recursion is presented, which yields a unique fixpoint even in unguarded cases. The construction is based on partially ordered place name sets. It is discussed how the partial orders used in this construction interact with each other and are related to the idea of causality.

12:50 - 14:20 Lunch Break.

14:20 - 15:10 Robin Milner, University of Edinburgh:
Commitment and the Polynomial Pi-Calculus.

Actions in the pi-calculus are so far (as in CCS) very simple; they can be composed from just one or two particles. These particles carry information, so the action structure cannot be enriched exactly as in ACP, Esterel or SCCS. But it can in fact be enriched in such a way that actions themselves form a rich algebraic structure over which there is a partial order - commitment - representing interaction or reduction. The commitment order reflects causal independence, in this sense: precedence holds between two disjoint interactions only when one of them provides information needed to perform the other.

15:10 - 15:50 Walter Vogler, Technische Universität München:
Interval Orders and Action refinement.

A semantics supports the top-down design by action refinement if it induces a congruence for this operator. It can be shown that partial order semantics is necessary for this purpose. A special class of partial orders, the class of interval orders, turns out to be necessary and sufficient. Interval semantics is a partial order semantics, but does not model causality. Instead, it has a temporal flavour. Furthermore, it more or less allows a sequential representation.

15:50 - 16:20 Tea Break.

16:20 - 17:00 Maciej Koutny, University of Newcastle upon Tyne:
Extensions of Partial Order Semantics.

Partial orders describe concurrent histories which capture causality and concurrency invarants. We show that it is possible to express other invariants, such as weak causality and commutativity, by suitably extending the notion of a partial order.

17:00 - ?? Discussion (Moderator: Eike Best, Universität Hildesheim):
What Good, Then, are Partial Orders?

Organisation and Registration

Participants of the workshop will be charged a moderate fee of 30 Pounds to cover organisational overhead. Registered participants will receive a booklet containing abstracts of the speakers' talks, and a transcription of the discussion session, as well as a summary of the discussion during and in between talks. The fee also includes lunch and refreshments on Monday, June 22.

) If you wish to register by electronic mail, .

Prof. Eike Best
Institut für Informatik
Universität Hildesheim
e-mail: E.Best@informatik.uni-hildesheim.de
Tel. (49) 5121 883 741/0, FAX (49) 5121 860˜475.

Send a request for registration to Ms. Grace Roberts

Ms. Grace Roberts
Sheffield City Polytechnic, Conference Services
36 Collegiate Crescent
Sheffield, S10 2BP, United Kingdom
Tel. (44) 742 532 577, FAX (44) 742 532 579.

AI AT THE CROSSROADS
A REPORT OF THE FIRST INTERNATIONAL CONFERENCE INTO THE FUNDAMENTALS OF ARTIFICIAL INTELLIGENCE.

In the Crossroads Motel, at the junction of Adenauerstrasse and Karlmarxstrasse in Aachen, a rectangular negligent conference centre in the style of the "nouvelle brutalisme", a cross-disciplinary team have been examining the foundations of Artificial Intelligence research. This is the first fruit of the new interdisciplinary initiative funded by the NADIR directorate of the SERC. The problems this distinguished team have been addressing were first recognised by students studying advanced AI courses at the Universities of Margate and Auchtermuchty. These courses were sufficiently advanced to have cast doubt on the simplistic assumption that Artificial Intelligence is Expert Systems. The students realised at the end of the courses that they still didn't know what artificial intelligence was. In practice this could be reduced to the empirical question of how to tell an artificially intelligent program from one that was just pretending - or, to be more precise about it, how to tell a system that, if it were a human being, would naturally be supposed to be intelligent, from one that, if it were a human being, would naturally be supposed to be pretending to be one.

Prof Smart of Auchtermuchty's Department of Sociology and Comics (a cost-saving merger purely in order to share admininstrative overheads) said that his research had shown that self-categorisation of the different types of AI Research had been skewed by inappropriate nomenclature. Searle had popularised the categorisation of "strong ai" and "weak ai", but being a philosopher he had not considered the Adlerian problems posed by terms with such personal meaning to the super-ego.

This was the value of a cross-disciplinary approach. For example, if you were a small nyaff with spectacles who got beaten up in the school playground, and had worked hard for twenty-five years to become a professor so you could tip your nose at boneheaded bullies, would you admit to doing research into "weak ai"? Consequently almost all researchers claimed to be doing "strong ai" themselves, asserting "weak ai" to be research done by people in other disciplines which had run out of research funds, but who had found some way of convincing the research councils that what they were doing was relevant to AI. For example, German research had shown that photographs of tall blond people were generally supposed to be more intelligent than small dark people, even though half of them were actually wearing wigs and sitting in misleadingly scaled chairs. Indeed, this practice (of increasing the appearance of intelligence by wearing wigs and sitting in special chairs) had been used for centuries by the British Judiciary to give an appearance of intelligence.

Reasoning that analogous effects might pertain in AI, Smart's HCI researchers had established that programs running in computers with large display screens and moving coloured pictures were indeed judged to be the most intelligent, but were also thought to be dauntingly unapproachable. It has been known for a long time that the highest approachability index is possessed by fluffy things with large round eyes and bulging foreheads, hence the prevalence of spectacles, beards, and baldness amongst both male and female academics, thought to be an evolutionary adaptation to offset their otherwise off-putting intelligence. The HCI team are now conducting experiments in improving the approachability of artificially intelligent systems by replacing the plastic mouse with with a fluffy bunny rabbit. It turns out that the colour of the bunny is very important, but that this varies considerably between individuals. This poses as yet unsolved and very difficult technical problems which their industrial collaborators in the intelligent fluffy toy industry are working on.

Prof Abelson of Margate's Department of Divinity and Medieval Music (a cost-saving merger purely in order to share admininstrative overheads) pointed out that many of the current philosophical disputes underlying AI were intriguingly similar to the theological disputes which troubled the early Church. The question of the relation between real and artificial intelligence was very similar to the dispute between the Homoousians and the Homoiousians which Athanasius the Great settled in AD 362 in Alexandria. As recorded in the 'Tomus ad Antiochemus', Athanasius asserted that the 'ousia' of 'homoousia' meant 'having a common element' rather than 'identity', but it was not true, as first proposed by Zahn, and accepted in some form by Gwalkins, Harnack, and Seiburg, that 'homoousios' triumphed by a transformation into 'homoiousios'. Duchesne's verdict in 'The Early History of the Church' (vol ii E.T. p281) was a fair statement of the case: "The Nicene term was in no way ousted... But the idea which the 'homoiousios' accentuated was admitted, under another formula - that of the three 'hypostases' - as a useful and even necessary explanation of the 'homoousios'." The similarity to the current philosophical debates in AI was most striking.

Prof Abelson suggested that artificial intelligence was like moral goodness, or happiness, i.e., something which could never be achieved by trying to achieve it directly, but only as a side effect of trying to achieve it in other people - or in this case, machines. Thus theological and philosophical justification had been found for one of AI's basic, but hitherto unproved, assumptions.

As to the question of whether it would be possible for a machine to have an immortal soul, he considered that no machine which was not clever enough to wonder whether it had a soul could possibly be supposed to have one. In the case of machines which were capable of worrying about the question, it would initially have to be a question of faith on the part of the individual machine. The question would only be finally resolved by the advent of an Artificial Machine-Based Messiah. The NADIR directorate were currently seeking peer comment on the rather delicate question of Who should be responsible for this research project, and whether to waive the usual requirements of industrial relevance and the completion of the RG2 form in sextuplicate.

Their own special interest (i.e. the Department of Divinity and Medieval Music) was in artificially moral machines. Their current research project concerned scripture-based machines which were capable of understanding simple news stories from the tabloid press, and deciding whether or not the various actions described were right or wrong, using an Assumption-Based Morals Maintenance System based on the Holy Scriptures.

KAROL KZAPEK'S PROPHETIC THEATRE PIECE COMES TRUE!

FORTHCOMING EVENTS

1992

Date: June 1 - 5
Title: **International Conference on Measurement and Modelling of Computer Systems**
Acronym: SIGMETRICS '92/PERFORMANCE '92
Location: Newport, R.I., USA.
Sponsor: SIGMETRICS IFIP WG 7.3.
Contact: Linda Wright, Digital Equipment Corp., TAY1-2/J11, 151 Talor St., Littleton, MA 01460; 'phone (508) 952-4476.
Email: wright@tps.enet.dec.com.

Date: June 2 - 3
Title: **Sixth Israel Conference on Computer Systems and Software Engineering**
Location: Herzliya, Israel.
Sponsors: IEEE Computer Soc. Israel Chapter, Information Processing Assoc. of Israel.
Contact: Conference Secretariat, Ortra, PO Box 50432, Tel Aviv 61500, Israel, phone 972 (3) 664-825, fax 972 (3) 660-952.

Date: June 10 - 12
Title: **3rd Workshop on Metaprogramming in Logic**
Acronym: META-92
Location: Uppsala, Sweden.
Sponsor: Assoc. for Logic Programming and Uppsala Univ.
Contact: Jonas Barklund, Comp. Sci. Dept., Box 520, S-75120 Uppsala, Sweden; +46 18 181050; fax +46 18 521270.
Email: jonas@csd.uu.se

22

Date: June 15 - 18
Title: **11th International Conference on Automated Deduction**
Acronym: CADE 11.
Location: Saratoga Springs, NY, USA.
Contact: Program Chair: Deepak Kapur, (518) 442-4281.
Email: cade-11@cs.albany.edu
Or: Local Arrangements Chair: Neil V. Murray, (518) 442-3393.
Email: cade-11@cs.albany.edu

Date: June 15 - 18
Title: **Parallel Architectures and Languages Europe**
Acronym: PARLE '92.
Location: Paris, France.
Contact: Prof. Daniel Etiemble, Universite Paris Sud Orsay, LRI, Bat. 490, 91405 Orsay Cedex, France, tel: +33 1 69 41 66 21, fax: +33 1 69 41 65 86.
Email: de@lri.lri.fr
Or: Veronique Segaut, AFCET, 156 Bd. Pereire, 75017, Paris, tel: +33 1 47 66 24 19, telex +33 1 42 67 93 12.

Date: June 15 - 19
Title: **Logic and Computer Science**
Location: Marseille, France.
Contact: G. Blanc, Dept. of Mathematics and Computer Science, Faculty of Sciences of Luminy, Case 901, 13288 Marseille Codex 9, France.

Date: June 16 - 18
Title: **Computer Security Foundations Workshop V**
Location: Franconia, N.H., USA.
Contact: Leonard LaPadula, Information Security Technical Center, Mitre Corp., Bedford, MA 01730-0208, phone (617) 271-3261.
Email: ljl@mitre.org
Or: Ravi Sandhu, ISSE Dept., George Mason Univ., Fairfax, VA 22030-4444, phone (703) 993-1659.
Email: sandhu@sitevax.gmu.edu

Date: June 17 - 19
Title: **Conference on Programming Language Design and Implementation**
Acronym: ACM SIGPLAN'92
Location: San Francisco, California, USA.
Sponsor: ACM SIGPLAN.
Contact: Conference Chair: Stuart I. Feldman, Bellcore, 445 South Street, Morristown, NJ 07960-1910, 'phone (201) 829-4305.
Email: sif@bellcore.com
Program Chair: Christopher W. Fraser, AT&T Bell Laboratories, Room 2C-464, 600 Mountain Avenue, Murray Hill, NJ 07974-0636, 'phone (908) 582-3583.
Email: cwf@research.att.com
Local Arrangements Chair: Stan Osborne, Computer Science Department, San Francisco State University, 1600 Holloway Avenue, San Francisco, CA 94132, 'phone (415) 338-2168.
Email: stan@cs.sfsu.edu

Date: June 17 - 19
Title: **4th International Conference on Software Engineering and Knowledge Engineering**
Location: Capri, Italy.
Acronym: SEKE 92.
Sponsor: Univ. of Salerno, Univ. of Naples, CRIAI, Univ. of Pittsburgh, SWIFT.
Contact: S.K. Chang, Dept. of Comp. Sci., Univ. of Pittsburgh, Pittsburgh, PA 15260; (412) 624-8432.
Email: chang@cs.pitt.edu

Date: June 18 - 20
Title: **18th International Workshop on Graph-Theoretic Concepts in Computer Science**
Location: Weisbaden-Naurod, Germany.
Sponsor: Johann Wolgang Goethe Univ.
Contact: Ernst W. Mayr, J.W. Goethe Univ., Theoretische Informatik, Robert-Mayer-Str. 11-15, W-6000 Frankfurt am Main, Germany; Tel: ++49 69 7988325.
Email: mayr@thi.informatik.uni-frankfurt.de

Date: June 19 - 20
Title: **Workshop on Partial Evaluation and Semantics-Based Program Manipulation**
Sponsor: ACM SIGPLAN
Location: San Francisco, CA, USA.
Contact: General and program chair: Charles Consel, Department of Computer Science, Yale University, 51 Prospect Street, New Haven, CT 06520, USA.
Email: consel@cs.yale.edu
Local Arrangements: Stan Osborne.
Email: stan@ana.com

Date: June 20 - 21
Title: **Workshop on ML and its Applications**
Sponsor: ACM SIGPLAN
Location: San Francisco, CA, USA.
Contact: Program Chair: Peter Lee, School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213-3890, 'phone (412) 268-3049.
Email: Peter.Lee@cs.cmu.edu
General Chair: David MacQueen, AT&T Bell Laboratories, Room 2C-322, 600 Mountain Avenue, Murray Hill, NJ 07974.
Email: macqueen@research.att.com

Date: June 21
Title: **Workshop on Continuations**
Sponsor: ACM SIGPLAN
Location: San Francisco, CA, USA.
Paper Submission Details:
Olivier Danvy, Department of Computing and Information Sciences, Kansas State University, Manhattan, KS 66502, USA.
General and program chair: Olivier Danvy.
Email: danvy@cis.ksu.edu,
Or: Carolyn L. Talcott.
Email: clt@sail.stanford.edu

Date: June 22 - 24
Title: **International Conference on Theorem Provers in Circuit Design: Theory, Practice and Experience**
Location: Nijmegen, The Netherlands.
Contact: V. Stavridou, TPCD Programme Chair, Department of Computer Science, RHBNC, University of London, Egham Hill, Egham, Surrey, TW20 0EX, UK.

Date: June 22 - 24
Title: **1992 ACM Conference on Lisp and Functional Programming**
Location: San Francisco, CA, USA.
Contact: General Chair: Jon L. White, Lucid, Inc., 707 Laurel Street, Menlo Park, CA 94025, tel: 415-329-8400, x5514.
Email: jonl@lucid.com
Program Chair: William Clinger, Attn: LFP 92, Computer Science Dept., University of Oregon, Euhene, OR 97403, tel: 503-346-4411.
Email: will@cs.uoregon.edu

Date: June 22 - 25
Title: **7th Annual IEEE Conference on Structure in Complexity Theory**
Location: Boston, Mass., USA.
Sponsor: IEEE.
Contact: T. Long, Dept. of Comp. Sci., New Mexico Univ., Box 3001, Dept. 3CU, Las Cruces, NM 88003-0001.
Email: long@nmsu.edu.

Date: June 22 - 25
Title: **7th Annual IEEE Symposium on Logic in Computer Science**
Location: Santa Cruz, California, USA.
Acronym: LICS.
Contact: Program Chair: Prof. Andre Scedrov, Inst. for Research in Cognitice Sci., University of Pennsylvania, 3401 Walnut Street, Suite 400C, Philadelphia, PA 19104-6228, USA, Fax (215) 573 2048 (Attn: LICS).
Email: lics92@cis.upenn.edu

Date: June 23 - 25
Title: **Third International Workshop on Rapid System Prototyping**
Acronym: RSP.
Location: Research Triangle Park, N.C., USA.
Cosponsors: IEEE Computer Soc. Technical Committees on Design Automation, Simulation, and Test Tech.
Contact: Nick Kanapoulos, Center for Systems Eng., Research Triangle Inst., 3040 Cornwallis Rd., Research Triangle Park, NC 27709, phone (919) 541-7341, fax (919) 541-6515.
Email: rsp@rti.rti.org

Date: June 24 - 26
Title: **Thirteenth International Conference on Application and Theory of Petri Nets and Petri Nets Tutorial 22 - 23 June**
Location: Sheffield, England.
Contact: Kurt Jensen, Computer Science Department, Aarhus University, Ny Munkegade, Bldg. 540, DK-8000 Aarhus C, Denmark, tel: +45 86 12 71 88, fax: +45 86 13 57 25, telex: 64767 aausci dk.
Email: kjensen@daimi.aau.dk
Or: Conference Services, Sheffield City Polytechnic, 36 Collegiate Crescent, Sheffield S10 2BP, England, tel: +44 742 532 576, fax: +44 742 532 579.

Date: June 25 - 26
Title: **10th Software Reliability Symp.**
Location: Denver, USA.
Sponsor: IEEE Reliability Soc. Denver Chapter.
Contact: Rich Karcich, Storage Tech. Corp., 2270 S. 88th St., MS 2286, Louisville, CO 80028, tel: (303) 673-6223, fax: (303) 673-3353.

Date: June 29 - July 1
Title: **Fourth Workshop on Computer-Aided Verification**
Location: Montreal, Quebec, Canada.
Contact: G. v. Bochmann, Universite de Montreal, Department d'IRO, C.P. 6128, succ. A, Montreal, Quebec H3C 3J7, Canada.
Email: bochmann@iro.umontreal.ca

Date: June 29 - July 3
Title: **Sixth European Conference on Object-Oriented Programming**
Location: Aarhus, Denmark.
Acronym: ECOOP '92.
Contact: Ole Lehrmann Madsen, Comp. Sci Dept., Aarhus Univ., Ny Munkegade, DK-8000 Aarhus C, Denmark; Tel: +45 86 12 71 88; Fax: +45 86 13 57 25.
Email: olmadsen@daimi.aau.dk

Date: June 29 - July 3
Title: **2nd International Conference on The Mathematics of Program Construction**
Location: Oxford, UK.
Contact: Carroll Morgan & Jim Woodcock, Programming Research Group, 11 Keble Road, Oxford OX1 3QD, UK, tel: +44 865 273840.
Email: carroll@prg.ox.ac.uk
Or: jimw@prg.ox.ac.uk

Date: July 6 - 7
Title: **Workshop on Fault-Tolerant Parallel and Distributed Systems**
Location: Amherst, Mass., USA.
Contact: N.K. Jha, Dept. of Electrical Engineering, Princeton Univ., Princeton, NJ 08544; Tel: (609) 258-4754.
Email: jha@ee.princeton.edu

Date: July 7 - 10
Title: **Fifth Int'l Workshop on Computer-Aided Software Eng.**
Acronym: CASE 92
Location: Montreal.
Contact: Nazim Madhavji, School of Computer Science, McGill Univ., fax (514) 398 3993.
Email: case@softeng.cs.mcgill.ca.
Or: John Jenkins, City Univ. of London, Business Systems Analysis Dept., London, UK, fax 44 (71) 608 1270.

Date: July 8 - 10
Title: **International Symposium on Fault-Tolerant Computing**
Acronym: FTCS 22
Location: The Lafayette Hotel, Boston, MA, USA.
Sponsor: IEEE Computer Society, University of Massachusetts.
Contact: Prof. Dhiraj K. Pradhan, Conference Chairman, Electrical and Computer Engineering Dept., University of Massachusetts, Amherst, MA 01003; Tel: (413) 545-0160, Fax: (413) 545-4611.
Email: pradhan@ecs.umass.edu

Date: July 8 - 10
Title: **Third Scandinavian Workshop on Algorithm Theory**
Acronym: SWAT'92
Location: University of Helsinki, Finland.
Contact: Prof. Esko Ukkonen, Department of Computer Science, University of Helsinki, Teollisuuskatu 23, SF-00510 Helsinki, Finland, tel: +358-0-7084172, fax: +358-0-7084441.
Email: swat@cs.helsinki.fi

Date: July 12 - 17
Title: **Conference on Logic Programming and Automated Reasoning**
Location: St. Petersburg, Russia.
Acronym: LPAR'92.
Contact: A. Voronkov, ECRC, Arabellastr. 17, D-W 8000 München 81, Germany.

Date: July 12 - 17
Title: **19th International Colloquium on Automata, Languages, and Programming**
Acronym: ICALP 92
Location: Technische Universität Wien, Austria.
Contact: Prof. Werner Kuich, Institut für Algebra und Diskrete Mathematik, Technische Universität Wien, Wiedner Hauptstraße 8-10, A-1040 Wien, Tel.: +43 1 58801 5450.
Email: kuich@btx.UUCP.

Date: July 20 - 24
Title: **Symposium on Logical Foundations of Computer Science**
Acronym: LFCS '92.
Location: Tver University, Tver, USSR.
Contact: Dr. I.Kh. Musikaev, LFCS '92, Tver University, 33 Zhelyabova Str., Tver 170013, USSR.
Or: Prof. V.W. Marek, Dept. of Computer Science, University of Kentucky, Lexington, KY 40506 USA, fax (606) 257 4078, tel: (606) 257 3496.
Email: marek@ms.uky.edu
Or: Conference Manager: A.P. Stolboushkin, USSR Acad. Sci., Program Systems Inst., Pereslavl-Zalessky 152140, USSR, telex 412531 BOAT SU.
Email: aps1@node.las.msk.su

Date: July 27 - 29
Title: **International Symposium on Symbolic Algebraic Computation**
Acronym: ISSAC.
Location: Berkeley, California, USA.
Sponsor: SIGSAM.
Contact: Erich Kaltofen, 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 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, Kazašská 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: 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: 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: **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 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 MI 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.
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 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 Elettronica, via dela Ricera Scientifica, I-00173 Roma, Italy.
Email: iazeolla@irmias.bitnet

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.
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: **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 Programming 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: (++49 l 0) 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 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.
Email: eoog@cui.unige.ch

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 Vetoio, 67010 Coppito, L'Aquila, Italy.

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, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands, fax: (+31-20) 592-4199.
Email: apt@cwi.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 16 - 18
Title: Third Annual International Symposium on Algorithms and Computation
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@csce.kuyshu-u.ac.jp

Date: 18 - 20 December
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: February 25 - 27
Title: 10th Symposium on Theoretical Aspects of Computer Science '93 Congress Centrum
Location: Würzburg, Germany.
Acronym: STACS '92.
Sponsors: 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: April 13 - 17
Title: TAPSOFT 93 (CAAP FASE Advanced Seminar)
Location: Orsay, France.
Acronym: TAPSOFT 93.
Sponsors: AFCET, EATCS.
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: 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 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: 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