Synchronous Langauges

Willem-Paul de Roever (Universität Kiel, Germany)
Nicolas Halbwachs (Verimag Grenoble, France)
Gérard Berry (École des Mines, Sophia Antipolis, France)
Klaus Winkelmann (Siemens München, Germany)

December 02–07, 2001

The workshops on Synchronous Languages started in 1993 at Schloss Dagstuhl. Since then seven such workshops have been organized, in total: 2 in Germany, 1 in Spain, and 4 in France, with an attendance varying between 40 and 60 persons.

In 1993 the synchronous languages approach was promising: numerous design tools had been, or were in train of being, constructed based on the languages ESTEREL (Berry), Lustre (Halbwachs, Caspi), Signal (LeGuernic, Benveniste), and Statecharts (Harel, Pnueli), and were applied to the design of control of (real-time) embedded systems.

Now, in 2000/2001, this approach is established, and possibly even more essential to computer science than the well-known model-checking paradigm to automatic verification. The software for the Airbus 340 has been entirely designed using the SCADE tool, based on a graphical representation of Lustre; the software for the Airbus 385 is being designed, based on SCADE; the software for the Rafale fighter bomber by Dassault — the successor of the Mirage — has been designed using ESTEREL-based tools. Due to the application of tools based on synchronous languages, errors residing in translated code have been decreased by a factor of 20–25! That is, by more than an order of magnitude.

Similarly, the interest of VLSI manufacturers such as Texas Instruments, Intel, and Motorola in these tools is rising steeply. Until recently, Texas instruments could only test approximately 30% of the functionality of its chips; by using hardware design tools based on ESTEREL this percentage has again risen to full coverage, i.e., 100%. This may be a possible indication of this approach for hardware design and verification.

As a consequence of the increased amount of application, the limits up to which the synchronous paradigm can be pushed are being investigated. A number of researchers, e.g., at the Signal Laboratory of IRISIA at Rennes, and in several European consortia suggested by the Common Market, are investigating *how to combine the synchronous approach with the asynchronous ones.* Furthermore, the increased number of applications of this approach calls for deeper investigations into mathematical ramifications for this

approach, on the basis of statistics, probability theory and static abstraction techniques.

Contents

1 ESUIF: An Open-Source Esterel Compiler

Stephen A. Edwards
Columbia University, USA

Efficiently compiling large Esterel programs remains an open problem, and existing Esterel compilers are not open-source, hindering research on the topic.

This is a a work-in-progress description of a new open-source Esterel compiler I am currently developing. I describe its internal database and a new intermediate representation, as well as proposals for new code generation techniques.

My hope is that this framework will further research on compiling this interesting language.

2 Verification of Controllers for Hybrid Systems

Ben Lukoschus
University of Kiel, Germany

The analysis of real-world controlled systems is largely restricted by the complexity of modeling and computation. This talk presents a project in which decomposition as well as algorithmic and deductive verification are combined: a system is first decomposed into smaller physical and/or functional units and modeled by automata in various frameworks. These are specified by Assumption/Commitment pairs and can easily be model-checked. Finally, all A/C pairs can be combined deductively to obtain an overall property of the complete system.

This technique is illustrated by a laboratory chemical batch plant. The Symbolic Analysis Laboratory (SAL) tool (developed at SRI International) seems to be a good framework in which our method can be used, since it is designed to combine the use of several verification tools.

3 Phase Synchronous Equations

Mike Kishinevsky
Strategic CAD Laboratory, Intel

Synchronous languages have good potential for hardware design. However they are currently not capable of describing phase accurate behavior needed in high-speed design.

We present a formal model of phase-synchronous equations that provides formal semantics for circuits with transparent latches and two-phase clocks. Equations are generated from a netlist based on a linear complexity preliminary step of phase labeling for all wires in the design.

We next show how Esterel can be extended for modeling phase accurate behaviors and how the circuits are compiled from this extended version of the language.

4   Introduction to asynchronous design

Mike Kishinevsky
Strategic CAD Laboratory, Intel

This was a tutorial on synthesis of asynchronous circuits to satisfy curiosity of synchronous people in asynchrony.

5   Quantitative Verification of Reachability Properties on Probabilistic Systems by Means of Abstraction and Successive Refinements

Bertrand Jeannet
IRISA Rennes, France

Joint work with Kim Larsen, Henrik Jensen (Aalborg University), and Pedro d'Argenio (Twente Universtity)

Verification of probabilistic systems is faced to the state-explosion problem, like finite state verification. Its effect in probabilistic verification is however even worse, because numerical computations are involved instead of graph algorithms.

We propose a method that uses abstraction and performs approximate analysis in order to overcome this state explosion problem in case of reachability analysis.

We have first defined a probabilistic simulation relation that preserves such properties. We have designed efficient algorithms based on BDDs and MTBDDs data-structure to efficiently compute abstractions of a concrete system. Lastly, refinement algorithms allow to refine abstractions when these are too coarse to enable the proof of the property. Experiments on the BRP protocol were conducted and show that interesting properties can be shown on quite coarse abstractions in a fully automatic way.

6 Efficient Compilation of Array Iterators for Lustre

Lionel Torel
Verimag, France

Arrays were introduced in Lustre for hardware purposes (developement of regular circuits): the operators provided were particularely adapted to that kind of applications (slices extraction, static recursion, etc.). The compilation techniques used (based on expression of array into independent variables) were adapted both to hardware description and verification by model-checking but not to software development. The code generated is not as efficient as one would expect and writing programs with these operators can be very tedious.

We propose new operators (called "iterators") inspired from functional programming, along with a compilation scheme and optimizations for "cascades of iterators".

We show, that from these iterators we can generate much more efficient code.

7 A decidable clock language for synchronous specifications

Mirabelle Nebut
IRISA Rennes, France

Presence and absence of signals inside a reaction are inherent to the synchronous paradigm, as well as clocks which are sets of instants that indicate when a given condition is fulfilled over a sequence of reactions (e.g. when a signal is present). Clocks are essential to capture the control in data-flow specifications; more generally relations between clocks should be analyzed to detect e.g. inconsistencies and verify some properties. These relations express particular safety properties many of which can be verified without considering the dynamic of systems, by means of a static abstraction. We propose a language CL to describe such properties and prove it decidable. Finally, model-checking is derived for e.g. Signal programs, on the basis of a translation from the static abstraction of Signal into CL.

8 Slicing Synchronous Reactive Programs

S. Ramesh
Department of Computer Science and Engineering, Indian Institute of Technology Bombay.

Slicing is a well known in the domain of sequential transformational programs and has been found to be useful in analysis, debugging and verification. The classical definition of slicing is inadequate for reactive programs. In this talk, a new definition of slicing for reactive programs is proposed. An algorithm for computing slices based upon this definition is outlined. The Argos language is used for concrete description of our ideas; they are of general applicability and can be applied easily to other synchronous languages.

9 Design Decisions for a Synchronous Language

Reinhard Budde
Frauenhofer Inst, Germany

Based on the concrete syntax of Java, design decisions for synchronous languages are discussed

- to support re-usable self contained components

- to be in-line with the synchronous paradigm

- to get compact and efficient code

in the domain of controller design.

Extending causality checking from the control part to data by data flow analysis and abstract interpretation and using the same techniques to remove unnecessary dynamic bindings was discussed, too.

10 Verification of Embedded Systems: SafeAir Achievements

Klaus Winkelmann
Siemens AG, Germany

The IST project "SafeAir" aims at an Avionics System Design Environment (ASDE) based on formal specification, synchronous languages and formal verification. User partners EADS Israel Aircraft Industries and Snecma Moteurs co-operate with providers TNI-Valiosys, Telelogic, iLogix, INRIA, Siemens, OFFIS and Weizmann Institute. The project

started in January 2000 and has by November 2001 achieved a prototype integrating the ModelBuild, Scade, Simulink and Statemate front-ends with state-of-the-art verification tools BOOST (Siemens) for complex control-dominated discrete designs and HYBRID (OFFIS) for loosely coupled hybrid systems (discrete + continuous). Verification performance has been increased by a factor of 20 over previous results of the predecessor SACRES project. [See http://www.safeair.org]

11 Control Algorithms for Resource Allocation in 3G networks

Simin Nadjm-Tehrani
Linköping University, Sweden

I present an example synchronous computation embedded in a highly asynchronous environment: a future generation telecom network. The talk is based on recent work with collaborators[1] where we study strategies to deal with inherent load uncertainties in 3G mobile networks. We address QoS policies and resource allocation techniques at communication system layers; and specifically the problem of CPU load control in a radio network controller (RNC) at Ericsson Radio Systems.

The algorithms we present distinguish between two types of uncertainty: the resource needs for arriving requests and the variation over time with respect to user service policies. We use variants of adaptive control for the first type, and policy-dependent deterministic algorithms for the second type. One approach combines feedback control with a deterministic Pool allocation mechanism, while the other is based on an optimal control algorithm.

Simulation results based on real traffic models generated at Ericsson, which test our algorithms in comparison with their existing one are in progress.

12 A Calculus for communicating synchronous processes

Jean-Pierre Talpin
INRIA Rennes, project ESPRESSO

A broad range of applications requires reasoning on a combination of synchronous and asynchronous interaction, from co-designed hardware/software architectures, multi-threaded real-time systems to distributed and reactive telecommunication services. To

---

[1]Kayvan Najarian, Calin Curescu, Tomas Lingvall, Teresa Dahlberg

capture this whole spectrum of applications, we consider a calculus of communicating synchronous processes and address the issue of developing a synchronous specification on a distributed architecture: the issue of desynchronization. We introduced a criterion for checking desynchronization correct upon a hierarchical representation of processes that makes control explicit.

URLS:

http://www.inria.fr/prive/talpin

http://www.inria.fr/espresso

13 Synchronous Reasoning about Asynchronous Protocols

Bengt Jonsson
Uppsala University, Sweden

Joint work with Werner Damm, OFFIS. We address the problem of verifying/analyzing

collections of asynchronously communicating state machines. Such occur, e.g., in SDL. UML specifications or asynchronous concurrent languages, such as Erlang. In order to handle unbounded capacity in a safe way, there are several symbolic verification techniques that employ regular expressions (for queues), linear constraints on message occurrences, etc. We propose to represent possible channel contents by its potential effect on the receiver process. For each message transmission, a representation of its potential reception is generated. For communicating finite state machines, this yields a natural widening operation and guaranteed finiteness of the analysis. We show how the techniques can be extended to protocols with data parameters.

14 Software Esterel Execution

Dumitru Potop-Butucaru
Ecole des Mines de Paris

The translation of Esterel into sequential code has proved to be a difficult problem. Currently, there are two techniques for handling large programs: the circuit translation, followed by circuit simulation and the one based on a concurrent control-flow graph intermediate representation, which is then executed. The former is used in the Esterel Technologies compiler, while the latter is used in the compiler developed by Stephen Edwards and the CHEF group.

The circuit-based approach generates relatively slow code. On the other hand, the CCFG-based methods reject lots of correct programs.

In my presentation I proposed an intermediate representation and some simulation methods that try to combine the qualities of the two approaches. That is:

- The use of a CCFG-like formalism, in order to support fast sequential code generation;

- simplifications of this intermediate formalism;

- a scheduling method combining 3-valued simulation with acyclic code scheduling.

15 Hot and Cold Topics in Esterel Technologies

Gerard Berry
Esterel Technologies, France

My talk consisted of a number of parts:

- A sales pitch for my new company ESTEREL Technologies (about 100 employees)

- A presentation of its product development plan regarding new products, in particular as regards new compilers for the industrial Esterel VII, and the consequences of the recent acquisition of the SCADE tool and its corresponding development team.

- Technical issues regarding a version of a version of Esterel suitable for hardware design, done in combination with Michael Kishinewski (Intel), were treated in more detail.

- Main issues related to technology scaling:

    - Enhancements in verification technology
    - Test generation according to different criteria
    - Static analysis of synchronous programs
    - The geometry of transition systems, which needs to be better understood.

16 Verification of Embedded Control Software

Ralf Huuck
University of Kiel, Germany

Programmable Logic Controllers (PLC) are used to drive automative systems like transportation belts, lifts or chemical plants. The standard IEC 61131-3 provides a number of programming languages for PLCs. One of them is Sequential Function Charts (SFC), which is a graphical high-level description language comprising a number of interesting concepts like parallelism, hierarchy and activity manipulation.

Although defined in the standard, the semantics of SFC is far from obvious. In this talk we presented a formal syntax and semantics including all the aforementioned aspects. Moreover a translation to the input of CaSMV is defined. This is implemented in the tool SFCheck. Finally, an example application is shown and briefly discussed.

17 Embedded Synchronous Hardware Description Languages and Verification

Gordon Pace
INRIA Rhone-Alpes

work done with Koen Claessen - Chalmers University, Gothenburg, Sweden

Various languages have been proposed to describe synchronous hardware at an abstract, yet synthesizeable level. Quite frequently, for large problems, it is beneficial to be able to combine together different languages within the same framework for description, simulation, synthesis, and verification.

Our approach to this problem is by using embedded languages - allowing us to easily experiment with new formal languages and language features, and also providing easy access to formal verification tools aiding program verification. However, compilation of hybrid code proved to be more challenging than it appeared to be at the outset. Various hand proofs were necessary to prove that different languages and compilation approaches were compatible.

In this talk different ways of automatically reasoning about such embedded languages will be presented and will conclude with techniques to relegate many of the mundane proofs arising from language combinations to automatic model checkers - still from within the embedded-language framework.

18   Space and Time exploration of Lustre programs: the Ludic debugger

Fabien Gaucher
Verimag, France

Since synchronous languages are often used for safety critical systems, a lot of effort has been put on the definition of formal semantics and formal verification tools, to ensure a safe execution. But, as far as the development phase is concerned, programmers have to find bug causes by hand.

Our goal is to formalize what people are used to do by hand for exploring large programs. This approach is based on the algorithmic debugging principle, introduced by Shapiro for logic programs. It involves dynamic slicing of programs and provides a method for exploring both time and space dimensions of programs, where time is the discrete time of the basic clock and space is the network of operators that defines one reaction.

19   Rlucid Synchronous Language

John Plaice
The University of New South Wales, Australia

Rlucid is a timed extension to Wadge and Ashcroft's Lucid dataflow language. An RLucid stream is a timestamped stream, i.e.

$$X = ([x_0, s_0], [x_1, s_1], \ldots, [x_n, s_n], \ldots)$$

, where the x_i are the values of X and the $s_i \in \mathbf{N}^*$ are non-standard timestamps. Data operations are applied in a pointwise manner over their arguments, and take no time to execute. With this approach, all Lucid programs are valid RLucid programs. An additional RLucid operator, before, allows decisions to be made according to the arrival of inputs. RLucid gives a much simpler timed dataflow semantics than either Lustre or Signal. Nevertheless, compilation is more complex than for Lustre, and static analysis will be required to select those programs that might be executable with bounded memory.

20 Translating Statemate statecharts to Scade / Lustre

Marc Segelken
OFFIS Oldenburg, Germany
For the purpose of integrating statecharts as implemented

in the Statemate tools with Scade/Lustre models a translation from Statemate to Scade has been developed allowing for composition of models crossing language boundaries as well as for processing Statemate models through their Scade representation.

From SMI as an intermediate language within the translation a Scade model is produced after various transformations such as elimination of non-determinism and single-assignment code transformation, have been applied.

Thanks to the link from Scade to SMI based on the generated C code being built for verification purposes it would be possible to automatically apply an equivalence check on the generated Scade model thus partially validating the translation result.

21 Simulation of non-deterministic Synchronous Systems

Yvan Roux
Simulation of non-deterministic synchronous systems can be useful for validation purposes:

- generation of relevant test sequences

- debugging programs

We introduce a language (called Lutin) for describing non-deterministic systems. This language merge two styles:

- a declarative one to describe a set of possible event in instant

- a sequential one to describe temporal behavior (regular expressions)

We have implemented a prototype which can simulate the behavior of a non-deterministic system described in Lutin.

22 The recent work on the reactive package in the PING Project

Christian Brunette

Ecole des Mines de Paris, France

The goal of PING (Platform of Interactive Networked Game) is to develop a massively multi-player platform which support heavy virtual world. What we called massively multi-player is at least one thousand. The work of the reactive language in this project is to allow the user to create behaviors for autonomous objects or user-controlled one. To do this, we use Junior (a reactive language on Java) and Rejo (a reactive language on Junior). The recent works are about the consistency of the different simulation. Because we are on a distributed architecture and because it will be too heavy to synchronize all reactive machines, the reactive machines don't share the same instant and run at different speed. So we use some anticipation mechanisms to improve the consistency of the behaviors at the applicative level, the Dead-Reckoning. An other work was to give the possibility to the users to create their behaviors in a graphical way and for this, we use Icobjs (Iconic Objects). Icobjs permit to combine some elementary or more complex behaviors with a graphical constructor at run-time.

23   Symbolic simulation of reactive systems

David Garriou
IRCCyN, Nantes, France

Symbolic simulation of reactive systems is an intermediate method between simulation and formal verification of reactive systems. We handle programs involving boolean variables and unbounded numerical variables.

Symbolic simulation consists in simulating a reactive program by giving it at each instant the set of its possible inputs. A key aspect is the ability to deal with values for numerical variables and keep them in account while computing properties of control.

Symbolic simulation allows to verify safety properties of reactive systems. We classify the properties we can handle according to the way they may be proved (Taking into account dynamic behaviors of boolean or integer variables), and according to their nature (boolean or integer).

A symbolic simulator has been developed. It is based on DDCs (Diagram Decision with Constraints), which are BDDs whose nodes are either boolean variables or linear constraints. Operators of DDC are those of BDDs. Moreover DCCs come with a set of specific operators dealing with numerical parts of DDCs so that we can take into account semantics of constraints. At last, we describe a widening operator for DDCs to allow convergence of reachability analysis of programs.

24 A Tool for Validating Time Constraints

R. K. Shyamasundar
Tata Institute of Fundamental Research, India

Synchronous languages specify reactions of the reactive/real-time systems to its environment within the language. This makes it easier to validate and verify the systems relative to the assumptions and notions of simultaneity. However, concrete embedding of the code makes it necessary to predict time for reactions of the underlying run-time systems and the sensors so that the strict notion of simultaneity can be relaxed keeping the logical correctness intact. Further, for hard real-time systems, there is a need for referring to clock times. In this talk, we describe a method that uses timed annotations for ESTEREL programs that makes it possible to predict the timing constraints on run-time system, the sensors and clock times and validate the concrete realization relative to time-annotated ESTEREL specifications. Using such time annotations, it is possible to establish that the concrete implementation is indeed a correct realization of the abstract system specified in annotated ESTERELrelative to the given model or architecture. Further, the method establishes time constraints to be satisfied by the concrete architectures for realizing the logical specification of the system. We shall illustrate the technique with the example of producer-consumer example and show the design of a tool using some of the existing code generation tools of the ESTEREL environment. Furthermore, the method can also be used for arriving at scheduling constraints on the asynchronous tasks with which the ESTEREL specification is interfaced with.

25 Formal Reasoning About Esterel-Like Synchronous Languages

Klaus Schneider
Universität Karlsruhe Synchronous languages like Esterel have a clean semantics that

seems at a first glance to be fairly simple, but with a second look turns out to be quite subtle in some cases. While this is not a too severe problem for programmers (since they can simulate their programs), it makes it hard to prove the correctness of program transformations. For that purpose, we have embedded our Esterel-variant called Quartz into the interactive theorem prover HOL (higher order logic). The embedding was greatly simplified by using a new way for defining the semantics: The control flow is defined by predicates that describe entering conditions, conditions for internal moves, and termination conditions. The data flow is defined by guarded commands that consist of pairs of signal emissions/assignments and their weakest preconditions.

The embedding is now used for different applications: First, we defined a simplified circuit synthesis (as already available for Esterel) and proved its correctness for all programs. Hence, the theorem prover can now be used as a formally verified compiler that can compile programs into equivalent circuit netlists. Second, we defined a program transformation that splits statements into sequences of their surface and depths, which is a very important transformation, e.g. to eliminate schizophrenia problems. Third, we can verify properties of given programs either by interactive application of HOL proof rules or by translating into an input format for our external symbolic model checker. The latter is even able to compute the distance between two sets of program locations in terms of macro steps, or to count the minimal/maximal number of iterations a data-dependent loop may perform.

List of Participants - Dagstuhl Seminar 01491

webpage: http://www.dagstuhl.de/DATA/Participants/01491.html

Raul **Acosta-Bermejo**
Ecole des Mines de Paris
Centre de Mathématique Appliquées (CMA)
2004 Routes de Lucioles
B.P. 93
F-06902 Sophia Antipolis
France
fax: +33-4-92 38 79 98
e-mail: racosta@sophia.inria.fr

Gérard **Berry**
Esterel Technologies
885 avenue du Docteur Julie Lefèbvre
F-06270 Villeneuve
France
e-mail: Gerard.BERRY@esterel-technologies.com
url: http://www-sop.inria.fr/meije/personnel/Gerard.Berry.html

Christian **Brunette**
Ecole des Mines de Paris
Centre de Mathématique Appliquées (CMA)
2004 Routes de Lucioles
B.P. 93
F-06902 Sophia Antipolis
France

Reinhard **Budde**
Fraunhofer Institut
AIS - Autonome Intelligente Systeme
Schloss Birlinghoven
D-53754 St. Augustin
Germany
phone: +49 2241-14 2417
fax: +49 2241-14 2324
e-mail: Reinhard.Budde@gmd.de
url: http://ais.gmd.de/∼budde/

Stephen A. **Edwards**
Columbia University
Computer Science Dept.
MC 0401
1214 Amsterdam Avenue
NY 10027-7003 New York
USA
fax: +1 212 666 0140
e-mail: sedwards@cs.columbia.edu
url: http://www.cs.columbia.edu/∼sedwards/

David **Garriou**
Ecole Centrale de Nantes
IRCCyn
1 Rue de la Noe
B.P. 92101
F-44321 Nantes
France
fax: +33-2-40 37 69 30
e-mail: David.Garriou@irccyn.ec-nantes.fr

Fabien **Gaucher**
Verimag
INPG / ENSIMAG
2 Avenue Vignate
Centre Equation
F-38610 Gières
France
e-mail: Fabien.Gaucher@imag.fr

Nicolas **Halbwachs**
Verimag
INPG / ENSIMAG
2 Avenue Vignate
Centre Equation
F-38610 Gières
France
fax: +33-4-76 63 48 50
e-mail: halbwachs@imag.fr

Leszek **Holenderski**
Philips Research Laboratories
WDC-2-010
Prof. Holstlaan 4
P.O. Box 80.000
NL-5656 AA Eindhoven
The Netherlands
fax: +31 40 2745033
e-mail: Leszek.Holenderski@philips.com

Frank **Huch**
Universität Kiel
Institut für Informatik und Prakt. Mathematik
LST Programmiersprachen & Compilerbau
Olshausenstr. 40
D-24098 Kiel
Germany
e-mail: fhu@informatik.uni-kiel.de
url: http://www-i2.informatik.rwth-aachen.de/hutch/

Ralf **Huuck**
Universität Kiel
Inst. für Informatik und Prakt. Mathematik
Preuerstr. 1-9
D-24105 Kiel
Germany
fax: +49-431 566143
e-mail: rhu@informatik.uni-kiel.de
url: http://www.informatik.uni-kiel.de/∼rhu/

Klaus **Höppner**
Universität Kiel
Institut für Informatik und Prakt. Mathematik
Institut III - CAU Kiel
Olshausenstr. 40
D-24098 Kiel
Germany
fax: +49-431-880-7613
e-mail: klh@informatik.uni-kiel.de

Bertrand **Jeannet**
Aalborg University
Dept. of Comp. Science
add. adress: IRISA, Rennes
Fredrik Bajers Vej 7E
DK-9220 Aalborg
Denmark
phone: +33-299 84 74 72
e-mail: bjeannet@cs.auc.dk

Bengt **Jonsson**
Uppsala University
Dept. of Computer Systems
PO Box 325
S-751 05 Uppsala
Sweden
phone: +46-18 47 13 157
fax: +46-18 55 02 25
e-mail: bengt@docs.uu.se
url: http://www.docs.uu.se/∼bengt/

Mike **Kishinevsky**
Intel Corporation
Strategic CAD Lab. / JF4-211
2111NE 25th Street
OR 97124-6497 Hillsboro
USA
fax: +1-503-264-4490
e-mail: mkishine@ichips.intel.com

Marcel **Kyas**
Universität Kiel
Inst. für Informatik und Prakt. Mathematik
Christian-Albrechts-Platz 4
D-24118 Kiel
Germany
fax: +49-431-5661-43
e-mail: mky@informatik.uni-kiel.de
url: http://www.informatik.uni-kiel.de/∼mky/

Ben **Lukoschus**
Universität Kiel
Inst. für Informatik und Prakt. Mathematik
R315
Preuerstr. 1-9
D-24105 Kiel
Germany
fax: +49-431-56 61 43
e-mail: bls@informatik.uni-kiel.de
url: http://www.informatik.uni-kiel.de/∼bls/

Florence **Maraninchi**
Verimag
INPG / ENSIMAG
2 Avenue Vignate
Centre Equation
F-38610 Gières
France
e-mail: Florence.Maraninchi@imag.fr
url: http://www-verimag.imag.fr/∼maraninx/

Christophe **Mauras**
Ecole Centrale de Nantes
IRCCyn
1 Rue de la Noe
B.P. 92101
F-44321 Nantes
France
fax: +33-2-40 37 69 30
e-mail: Christophe.Mauras@irccyn.ec-nantes.fr

Sabine **Moisan**
INRIA
Sophia Antipolis
2004 Route des Lucioles
B.P. 93
F-06902 Sophia Antipolis
France
fax: +33-4-92 38 79 98
e-mail: sabine.moisan@sophia.inria.fr

Lionel **Morel**
Verimag
INPG / ENSIMAG
2 Avenue Vignate
Centre Equation
F-38610 Gières
France
fax: +33-4-76 63 48 50
e-mail: lionel.morel@imag.fr

Simin **Nadjm-Tehrani**
Linköping University
Dept. of Computer & Information Science
SE-58183 Linköping
Sweden
fax: +46-13-284-020
e-mail: snt@ida.liu.se
url: http://www.ida.liu.se/∼rtslab/

Mirabelle **Nebut**
Université de Rennes
IRISA
B 219
Campus de Beaulieu
F-35042 Rennes
France
fax: +33-2-99 84 25 28
e-mail: mnebut@irisa.fr
url: http://www.irisa.fr/prive/Mirabelle.Nebut/

Gordon **Pace**
INRIA Rhone-Alpes
ZIRST
VASY
655 Ave. de l'Europe
F-38330 Montbonnot St. Martin
France
fax: +33-4-76 61 52 52
e-mail: Gordon.Pace@inria.fr
url: http://www.inrialpes.fr/vasy/people/Gordon.Pace/

Fabrice **Peix**
INRIA
Sophia Antipolis
2004 Route des Lucioles
B.P. 93
F-06902 Sophia Antipolis
France
e-mail: fabrice.peix@sophia.inria.fr

Marc **Perrault**
Esterel Technologies
885 avenue du Docteur Julie Lefèbvre
F-06270 Villeneuve
France
fax: +33-4-92 02 57 85
e-mail: marc.perreaut@esterel-technologies.com

Dimitri **Petoukhov**
Ecole des Mines de Paris
Centre de Mathématique Appliquées (CMA)
2004 Routes de Lucioles
B.P. 93
F-06902 Sophia Antipolis
France
fax: +33-4-92 38 79 98
e-mail: dimitri.petoukhov@sophia.inria.fr

John **Plaice**
The University of New South Wales
School of Computer Science and Engineering
NSW 2052 Sydney
AU
phone: +61-2-93 85 60 12
fax: +61-2-93 85 57 70
e-mail: plaice@cse.unsw.edu.au
url: http://www.cse.unsw.edu.au/∼plaice/

Axel **Poigné**
Fraunhofer Institut
FIT - Inst. for Applied Information Technology
Schloss Birlinghoven
D-53754 St. Augustin
Germany
phone: +49-2241-142-440
fax: +49-2241-142-324
e-mail: poigne@ais.fraunhofer.de
url: http://www.ais.fraunhofer.de/∼ap/

Dumitru **Potop-Butucaru**
Ecole des Mines de Paris
Centre de Mathématique Appliquées (CMA)
2004 Routes de Lucioles
B.P. 93
F-06902 Sophia Antipolis
France
fax: +33-4-92 38 79 98
e-mail: dpotop@sophia.inria.fr

S. **Ramesh**
Indian Institute of Technology
Dept. of Computer Science & Eng.
Powai
400 076 Bombay
India
phone: +91-22 576 7722
fax: +91-22-579 4290
e-mail: ramesh@cse.iitb.ernet.in
url: http://www.cse.iitb.ernet.in/∼ramesh/

Pascal **Raymond**
Verimag
INPG / ENSIMAG
2 Avenue Vignate
Centre Equation
F-38610 Gières
France
e-mail: Pascal.Raymond@imag.fr

Annie **Ressouche**
INRIA
Sophia Antipolis
2004 Route des Lucioles
B.P. 93
F-06902 Sophia Antipolis
France
fax: +33-4-92 38 79 88
e-mail: Annie.Ressouche@sophia.inria.fr

Martin **Richard**
Ecole de Mines de Nantes
4 rue Alfred Kastler
B.P. 20722
F-44321 Nantes cedex 3
France
e-mail: Martin.Richard@emn.fr

Jan-Willem **Roorda**
Chalmers University of Technology
Dept. of Computing Science
Eklandagatan 86
SE-41296 Göteborg
Sweden
fax: +46-31-16 56 55
e-mail: jwr@cs.chalmers.se
url: http://www.cs.chalmers.se/∼jwr/

Yvan **Roux**
Verimag
INPG / ENSIMAG
2 Avenue Vignate
Centre Equation
F-38610 Gières
France
e-mail: Yvan.Roux@imag.fr

Klaus **Schneider**
Universität Karlsruhe
Fakultät für Informatik
Institute for Computer Design & Fault Tolerance
Adenauerring 20a
Postfach 6980
D-76128 Karlsruhe
Germany
phone: +49 (721) 608 3641
fax: +49 (721) 370 455
e-mail: Klaus.Schneider@informatik.uni-karlsruhe.de
url: http://goethe.ira.uka.de/∼schneider/

Marc **Segelken**
OFFIS
Escherweg 2
D-24121 Kiel
Germany
phone: +49-441-798-2154
fax: +49-441-798-2145
e-mail: marc.segelken@offis.de
url: http://ca.informatik.uni-oldenburg.de/staff/Marc.Segelken.html

Ellen M. **Sentovich**
Cadence Berkeley Labs
3rd Floor
2001 Addison Street
CA 94704 Berkeley
USA
phone: +1-510-647-2807
fax: +1-510-486-0205
e-mail: ellens@cadence.com
url: http://www-cad.cecs.berkeley.edu

Mary **Sheeran**
Chalmers University of Technology
Dept. of Computing Science
Eklandagatan 86
SE-41296 Göteborg
Sweden
phone: +46-31-772-1013
fax: +46-31-16 56 55
e-mail: ms@cs.chalmers.se
url: http://www.cs.chalmers.se/∼ms/

Rudrapatna K. **Shyamasundar**
Tata Institute of Fundamental
School of Technology & Computer Science
400 005 Bombay
India
phone: +91 22 215 2971 X 2288
fax: +91 22 215 2181
e-mail: shyam@tcs.tifr.res.in
url: http://www.tcs.tifr.res.in/∼shyam/

Jean-Ferdinand **Susini**
Ecole des Mines de Paris
Centre de Mathématique Appliquées (CMA)
2004 Routes de Lucioles
B.P. 93
F-06902 Sophia Antipolis
France
fax: +33-4-92 38 79 98
e-mail: Jean-Ferdinand.Susini@sophia.inria.fr

Jean-Pierre **Talpin**
Université de Rennes
INRIA
Campus de Beaulieu
F-35042 Rennes
France
phone: +33-2-99 84 74 36
e-mail: talpin@irisa.fr
url: http://www.irisa.fr/prive/talpin/

Olivier **Tardieu**
INRIA
Sophia Antipolis
2004 Route des Lucioles
B.P. 93
F-06902 Sophia Antipolis
France
fax: +33-4-92 38 79 98

Shmuel **Tyszberowicz**
Tel Aviv University
Dept. of Computer Science
Ramat Aviv
69978 Tel-Aviv
Israel
phone: +972-3-6407728
fax: +972-3-6409357
e-mail: tyshbe@math.tau.ac.il

Klaus **Winkelmann**
Siemens AG
Corporate Technology
Otto-Hahn-Ring 6
D-81739 München
Germany
phone: +49-89-636-4 11 25
fax: +49-89-636-4 22 84
e-mail: klaus.winkelmann@mchp.siemens.de

Willem-Paul **de Roever**
Universität Kiel
Inst. für Informatik und Prakt. Mathematik
Christian-Albrechts-Platz 4
D-24118 Kiel
Germany
phone: +49-431 56 04 71 / 74
fax: +49-431 56 61 43
e-mail: wpr@informatik.uni-kiel.de
url: http://www.informatik.uni-kiel.de/∼wpr/

Robert **de Simone**
INRIA
Sophia Antipolis
2004 Route des Lucioles
B.P. 93
F-06902 Sophia Antipolis
France
e-mail: Robert.de_Simone@sophia.inria.fr