CONFER-2

CONcurrency and Functions:

Evaluation and Reduction

Working Group

Project Number: 21836

Periodic Progress Report
December, 1998

Contents

Chapter 1

Overview

This report contains the Second Periodic Progress Report for ESPRIT WG Nr. 21836 (CONFER-2).

The report contains 3 main parts: management at consortium level and at each site, deliverables (programs of each CONFER-2 workshops 3 and 4, software deliverables), and appendices listing CONFER-2 publications.

Further information may be requested from the coordinator:

Jean-Jacques Lévy
INRIA, Rocquencourt
bat.8, Domaine de Voluceau
78153–Le Chesnay, Cedex
France
tel: +33-1-39-63-56-89
fax: +33-1-39-63-53-30
e-mail: `Jean-Jacques.Levy@inria.fr`

This document has been compiled from input from all of the partners in the CONFER-2 Working Group: Cosimo Laneve, Peter Sewell, Jan Willem Klop, Samson Abramsky, Pierre-Louis Curien, Jean-Francois Monin, Bent Thomsen and Lone Leth, Davide Sangiorgi, Bjorn Victor, Gianluigi Ferrari, Matthew Hennessy, and David Walker. Lone Leth and Bent Thomsen from ICL greatly helped in the writing and the assembly of the document.

Chapter 2

Executive summary

The CONFER-2 Working Group follows Basic Research Action CONFER. It is a framework to keep together the CONFER community, and to provide 2 annual workshops.

The overall objective of the CONFER-2 action is similar to the one of CONFER and to create both the theoretical foundations and the technology for combining the expressive power of the functional and the concurrent computational models. The action is organised around four main areas [1]:

- Foundational Models and Abstract Machines
- Calculi
- Logics for Concurrency and the λ-calculus
- Programming Languages

In the second year of CONFER-2, work has progressed well in many areas. Below is a short summary of some of the results, grouped according to main areas. For further details see the site reports in chapter 3.

In the area of Foundational Models and Abstract Machines progress has been made in three main directions.

Firstly the categorical understanding of models of process calculi has been furthered by recent work by Gian Luca Cattani on presheaf categories as denotational models of concurrent process calculi. Also work by Gardner and Wischik on the graphical understanding of the π-calculus takes advantage of categorical models. A symmetric version of action calculi is giving a simpler and more natural way of representing the π-calculus graphically.

Secondly the work on game semantics is maturing. Recently Abramsky, Honda and McCusker used game semantics to give the first fully abstract model for a language with higher-order functions and general reference types—a substantial fragment of “real” languages such as SML and SCHEME. Also a first bridge between dynamic semantics and static semantics of functional languages has been given using game semantics.

[1] see Periodic Progress Reports of CONFER for more explanations

Thirdly research has continued on developing the mathematical foundations of mobile systems based on the tile logic. The natural targets of the tile approach are models of mobile systems which are compositional in both space and time.

In the area of Calculi several studies of objects have been undertaken. A class-based calculus of concurrent objects has been obtained by Fournet, Maranget, Laneve and Rémy from the join-calculus by introducing primitive record structure. Also Kleist and Sangiorgi have given an interpretation of Abadi and Cardelli's first-order Imperative Object Calculus into a typed π-calculus, and Silvano Dal-Zilio has used the modelisation of Abadi and Cardelli objects in the blue calculus extended with a very simple system of localities.

Also in the area of Calculi there has recently been a growing interest in asynchronous languages and process calculi, where message emission is non-blocking. An important reference for this class of languages is the asynchronous pi-calculus. Castellani and Hennessy have studied testing semantics for asynchronous process calculi and their descriptive power has now been largely demonstrated. Massimo Merro has shown that the asynchronous π-calculus is powerful enough to encode asynchronous variants of the Chi, Update, and Fusion calculi which are modifications of the π-calculus in which communications have a more global scope. Victor has developed a broad foundational theory of the fusion calculus, both labelled and unlabelled operational semantics, and treated strong and weak bisimulation equivalences for both semantics in some detail.

Milner and Sawle are defining and investigating the polynomial pi-calculus, which may be described as the pi-calculus extended by multiway communication. Although not so easy to implement, this notion is quite attractive because it is extremely powerful, formally speaking. For example, guarded summation is derivable, up to strong (not weak) equivalence; the calculus also appears to have some advantage in expressing security protocols.

Considerable progress has been made towards an understanding of operational congruences for arbitrary calculi defined by reduction semantics by Leifer and Sewell. Leifer gave bisimulation congruence results for two sub-classes of action calculi, resting on algebraic results about action calculus context decompositions. Three simple classes were considered by Sewell, firstly ground term rewriting, then left-linear term rewriting, and then a class which is essentially the action calculi lacking substantive name binding.

In the area of Logics for Concurrency and the λ-calculus Klop and Bethke have worked on "origin tracking", both in the framework of first-order term rewriting and λ-calculus. For first-order rewriting using this technique a proof was given of the theorem of Huet and Lévy on normalization of needed reductions. This technique is in the literature also called "dynamic dependence tracking" and is used for applications such as program slicing and error recovery. For λ-calculus a corresponding notion of origin tracking using Lévy's labeled λ- calculus was formulated and used to give a new proof of the classical Genericity Lemma in λ-calculus.

In the area of Programming Languages Bologna has proceeded in the development of their implementation of Lamping's graph reduction algorithm for optimal reduction of functional languages and they have completed a new "light" version, essentially inspired by Girard's Light Linear Logic. Gardner and Norrish have experimented with the design and construction of specification tools for action graphs, which allow the user to naturally move between the syntactic and graphical presentations. Work on extending the implementation to reflexive action calculi is underway (in a student project). This is a non-trivial extension, which we hope will lead to a simpler implementation. Sewell, Unyapoth and Wojciechowski, in collaboration with Pierce, are studying the design, semantic definition and implementation of location-independent communication primitives for mobile agents. They have proposed two simple calculi of agents that can migrate between sites and can interact by both location dependent and location independent message passing. Work on an implementation, and on correctness proofs of the distributed algorithms involved, is ongoing.

CNET has investigated some aspects of low-level security in distributed programming languages, such as protecting hosts against potentially hostile mobile code in open environments. ICL has worked on transferring ideas from the Facile system to the Java world via the mlj (ML to Java Virtual Machine code) compiler from Persimmon Research. Interfaces for the Facile concurrency primitives have been implemented. ICL is currently looking into how various packages for Mobile Agents in Java can be used to implement (some of) the constructs for distributed computing from Facile. There are now two implementations of the join-calculus. Version 1.03 has been released in 1997 by Fournet and Maranget. This prototype includes a distributed implementation of the join-calculus. In June 1998, `jocaml` 1.07 has been released by le Fessant. It is an extension of the virtual machine of Objective Caml (Ocaml), which allows full compatibility with Ocaml. `jocaml` is much faster than the first implementation of the join-calculus, and more expressible, since it allows to freely use communication channels inside Caml or to send any Caml data-structure on a channel. Pisa has introduced the language KLAIM (*Kernel Language for Agents Interaction and Mobility*). KLAIM is an experimental programming language, inspired by the Linda paradigm, where mobile agents and their coordination strategies can be naturally programmed. KLAIM provides direct support for expressing and enforcing security policies that control access to resources and data. In particular, the language uses types to protect resources and data and to establish policies for access control.

As seen from the site reports in chapter 3, crossfertilisation of ideas among fellow researchers in the working group continues to be very lively. As in previous years it is evident that related work done at other sites in the consortium is referenced often and also very often used as the basis for development of new results. It is no surprise that a number of these results have been achieved in collaboration between sites, enabled through site visits for shorter or longer periods. Leading researchers outside the consortium have also been involved in fruitful collaborations.

During Year 2 of CONFER-2 two workshops have been held.

The working group has produced a large number (about 70) of reports and several of these have been published at conferences, international workshops and journals. Many of theses works are accessible at URL

`http://www.cs.auc.dk/mobility/`

where a Web page on Mobility is maintained by Uwe Nestmann, who moved in September 1997 from INRIA Rocquencourt to Aalborg.

Several Ph.D.'s are in preparation in the action, and several have been defended.

This report can be accessed on the World Wide Web at:

`http://para.inria.fr/confer/`

Chapter 3

Management

3.1 Consortium level

The main management of the project is done at INRIA, Rocquencourt, and at ICL.

The following activities at the consortium level have taken place during the second year of the CONFER-2 working group:

Two workshops have been held. The first took place in April 1998 in Brighton near University of Sussex with 20 presentations and 30 participants, and was organised by Alan Jeffrey and Matthew Hennessy. The second workshop took place in September 1998 near INRIA Sophia-Antipolis in Nice with 14 presentations and 30 participants. It was organised by Gérard Boudol.

In 1998, the site of Warwick left our working group; a new site, Oxford University, is now participating to our group. This is due to David Walker's move from Warwick to Oxford.

As specified in §2.6 of the Working Group Programme, the vast majority of the activity of CONFER-2 is devoted to the organisation of 2 workshops per year. The cost statement forms will strengthen this distribution of the activity, although some of the funds are still reserved for travel expenses between sites.

3.2 Università di Bologna

3.2.1 Research

The research activity in Bologna, organized along the main task of the Working Group, has been the following:

Calculi In this area, we have been involved both in the analysis of bisimulation in the framework of the join-calculus and the study of a novel class-based concurrent calculus – the *objective join-calculus.*

Analysis of Bisimulation We have mainly investigated the theory of bisimulation in the join-calculus [BFL97]. For this purpose, we introduce a refined operational model that makes interactions with the environment explicit. In this framework we show that the so-called "*locality property*" strongly affects the treatment of bisimulation, and provides a semantics simpler than other proposals in the literature. We also propose several formulations of bisimulation and we establish that all formulations yield the same equivalence, which is finer than barbed congruence. Bisimulation and barbed congruence coincide in the presence of name-testing.

Inheritance in the join-calculus In [FLMR98] we present an elementary class-based calculus of concurrent objects obtained from the join-calculus by introducing primitive record structure. We provide inheritance as the combination of a few operators to assemble objects from partial join-calculus definitions. As usual, method definitions and private fields of a parent class can be reused, overridden, or extended. As expected with concurrent objects, inheritance admits some but not all refinements of the synchronization patterns of the parent class.

Logics for Concurrency and the λ-calculus Our research in this field has been mainly focused on the theory of optimal sharing in β-reduction. In particular, we proved that the notion of optimal "parallel" reduction, as formalized by Lèvy, is not *elementary recursive* [AM98]. The result should not be understood in a negative sense: it just says that the notion of optimal sharing *cannot have* a simple implementation since, essentially, *all the computational work* can be done via sharing. Actually, the result is a straightforward consequence of the fact that every term of the simply typed lambda calculus can be essentially reduced in a linear number of "parallel" (i.e. sharable) β-reductions.

Programming Languages We proceeded in the development of our implementation of Lamping's graph reduction algorithm for optimal reduction of functional languages (the Bologna Optimal Higher-order Machine [AG98]). We have completed a new "light" version, essentially inspired by Girard's Light Linear Logic.

3.2.2 Workshops, Travels and Visits

Travels Andrea Asperti and Luca Padovani participated at the Third CONFER-2 Workshop held in Brighton, April 1998.

Cosimo Laneve participated at the Fourth CONFER-2 Workshop held in Nice, September 1998.

Visits Vincent Danos visited University of Bologna for two weeks in July 1998. He gave a course on game semantics and he worked with Andrea Asperti on this issue.

Cosimo Laneve visited INRIA Rocquencourt in July 1998, where he worked with Cédric Fournet, Luc Maranget and Didier Remy on "Concurrent Class-Based Languages".

3.2.3 Publications

[AM98] A. Asperti, H.Mairson *Optimal β-reduction is not elementary recursive.* Proc. of the twenty-fifth Annual ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL'98).

[AG98] A. Asperti, S.Guerrini *The Optimal Implementation of Functional Programming Languages.* in *"Cambridge Tracts in Theoretical Computer Science"* Series, Cambridge University Press, October 1998.

[BFL97] M. Boreale, C. Fournet, C. Laneve: *Bisimulations in the Join-Calculus.* In Proc. Programming Concepts and Methods, Procomet '98, New York, U.S.A., 1998.

[FLMR98] C. Fournet, C. Laneve, L. Maranget, D. Rèmy: *Inheritance in the Join-Calculus.* Draft, October 1998.

3.3 University of Cambridge

3.3.1 Research

Work at Cambridge has proceeded on several fronts, mainly in the areas *Foundational Models*, *Calculi*, and *Programming Languages*. Most items lie in two or three areas.

- **Bisimulation** Considerable progress has been made towards an understanding of operational congruences for arbitrary calculi defined by reduction semantics. In [Lei98] Leifer gave bisimulation congruence results for two sub-classes of action calculi, resting on algebraic results about action calculus context decompositions; in [Sew98b, Sew98c] calculi with reduction semantics of three simple classes were considered by Sewell, firstly ground term rewriting, then left-linear term rewriting, and then a class which is essentially the action calculi lacking substantive name binding. General definitions of labelled transitions are given in each case, uniformly in the set of rewrite rules, and without requiring the prescription of additional notions of observation. They give rise to bisimulation congruences. Both works were presented at CONFER-2 workshops.

- **Presheaf Models** Gian Luca Cattani has recently joined the Cambridge site, bringing with him a new thread of work on categorical models of process calculi. He has studied presheaf categories as denotational models of concurrent process calculi, with a built-in notion of bisimulation, following the approach of Joyal, Nielsen and Winskel on bisimulation from open maps. His most recent published work includes a joint paper with Fiore, at Sussex and Winskel, at Aarhus University, on a general theory of domains which theoretically grounds the previous work on presheaf models [CFW98] (published before arrival at Cambridge), and a joint paper with Power, at Edinburgh, and Winskel [CPW98]. The latter paper axiomatically introduces categories of models which essentially generalise presheaf categories, while keeping the size of the categories small. A special emphasis in this and in the work on presheaf models was given to general proofs that bisimilarity is a congruence. While working on the paper [CPW98], categorical questions concerning free cocompletions arose naturally and have been addressed in [PCW98].

- **Type-theoretic and categorical models** Together with researchers at Edinburgh, Gardner has studied the type theoretic and categorical presentations of action calculi and their extensions. They give a propositions-as-types interpretation of action calculi [BGHP98], using the general binding operators of Aczel, and have shown that the type theory has a sound and complete interpretation in Power's categorical models of action calculi. They also give a conservative translation of the type theory in Benton's models for linear logic, which corresponds to the fact that Benton's models of linear logic are related to models of action calculi. Gardner and Hasegawa [GH98] give the type-theory and categorical models of the higher-order extension, and Hasegawa, in his Edinburgh PhD thesis, extends the results to the reflexive case. His type theory has similarities with the cyclic λ-graphs of Ariola and Klop.

- **Symmetric Action Calculi** Gardner and Wischik are investigating a symmetric version of action calculi. The motivation for this is threefold: it gives a simple, natural way of representing the π-calculus graphically; it is an obvious question to ask when considering the combinatoric version and corresponding category theory; and more broadly it explores the role of names and name-abstraction in general. We have results which link an algebraic presentation, the corresponding (weak) normal forms, the graphical presentation and a combinatoric version. We also believe that the symmetric action calculus is a conservative extension of the reflexive action calculus. We are in the process of discussing the categorical models with Martin Hyland. In particular, it seems likely that our basic wiring graphs have the structure of a Frobenius algebra with an additional axiom. In order to fully justify our framework, we must also look at examples and bisimulation. Our hope is that Leifer's results for the reflexive action calculi will transfer fairly easily to our setting. Our normal forms are easier to work with than the reflexive normal forms, so we might find that the proofs fall out quite naturally.

- **Specification Tool** Gardner and Norrish have experimented with the design and construction of specification tools for action graphs, which allow the user to naturally move between the syntactic and graphical presentations. Work on extending the implementation to reflexive action calculi is underway (in a student project). This is a non-trivial extension, which we hope will lead to a simpler implementation.

- **Polynomial Pi Calculus** Milner and Sawle are defining and investigating the polynomial pi calculus, which may be described as the pi calculus extended by multiway communication. Although not so easy to implement, this notion is quite attractive because it is extremely powerful, formally speaking. For example, guarded summation is derivable, up to strong (not weak) equivalence; the calculus also appears to have some advantage in expressing security protocols.

- **Pi Calculus Book** Milner has completed a Pi calculus textbook [Mil99].

- **Type systems** Earlier work on type systems for distributed programming has been developed by Sewell: in [Sew98a] a distributed π-calculus is equipped with a subtyping system that allows positive and negative capabilities to be restricted to particular locations. It is shown that the most local possible capabilities can be inferred automatically.

- **Mobile Agent Infrastructure** Sewell, Unyapoth and Wojciechowski, in collaboration with Pierce, are studying the design, semantic definition and implementation of location-independent communication primitives for mobile agents. In [SWP98a, SWP98b] we proposed two simple calculi of agents that can migrate between sites and can interact by both location dependent and location independent message passing. Work on an implementation, and on correctness proofs of the distributed algorithms involved, is ongoing; preliminary work was presented at the Nice CONFER-2 workshop.

3.3.2 Personnel and exchanges

Researchers and PhD students associated with CONFER-2 at Cambridge include Gian Luca Cattani, Philippa Gardner, James Leifer, Robin Milner, Michael Norrish, Mark Sawle, Peter Sewell, Alistair Turnbull, Asis Unyapoth, Lucian Wischik and Pawel Wojciechowski.

During 97-98 Leifer visited INRIA-Sophia Antipolis, Gardner visited INRIA-Rocquencourt and Sewell visited Edinburgh. Leifer and Sewell attended CONFER-2 workshops in Sussex and Nice.

3.3.3 Publications

[BGHP98] A. Barber, P.A. Gardner, M. Hasegawa and G. Plotkin, From Action Calculi to Linear Logic. *European Association of Computer Science Logic*, Springer-Verlag, 1998; conference article appears in BRICS Notes Series NS-97-1, 1997.

[CFW98] G. L. Cattani, M. Fiore and G. Winskel. A Theory of Recursive Domains with Applications to Concurrency. Proceedings of the Thirteenth Annual IEEE Symposium on Logic in Computer Science, LICS '98, 214–225.

[CPW98] G. L. Cattani, A. J. Power and G. Winskel. A Categorical Axiomatics for Bisimulation In *Proceedings of CONCUR '98: Concurrency Theory (Nice). LNCS 1466*, pages 581–596, September 1998.

[GH98] P.A. Gardner and M. Hasegawa. Higher-order and Reflexive Action Calculi: their type-theory and models submitted to Information and Computation, 1998.

[Lei98] James Leifer. Bisimulation Congruences for Action Calculi. Thesis submitted to Trinity College, Cambridge, 1998.

[Mil99] Robin Milner. Communicating and Mobile Systems: the Pi Calculus. Forthcoming (CUP Jan 1999)

[PCW98] A. J. Power, G. L. Cattani and G. Winskel. A Representation Result for Free Cocompletions. BRICS Technical report RS-98-21, Department of Computer Science, University of Aarhus, September 1998. Submitted for publication.

[SWP98a] Peter Sewell, Paweł T. Wojciechowski, and Benjamin C. Pierce. Location independence for mobile agents. In *Proceedings of the Workshop on Internet Programming Languages (Chicago)*, May 1998.

[SWP98b] Peter Sewell, Paweł T. Wojciechowski, and Benjamin C. Pierce. Location-independent communication for mobile agents: a two-level architecture. Submitted for publication.

[Sew98a] Peter Sewell. Global/local subtyping and capability inference for a distributed π-calculus. In *Proceedings of ICALP '98: International Colloquium on Automata, Languages and Programming (Aarhus). LNCS 1443*, pages 695–706. Springer-Verlag, July 1998.

[Sew98b] Peter Sewell. From rewrite rules to bisimulation congruences. Technical Report 444, University of Cambridge, June 1998.

[Sew98c] Peter Sewell. From rewrite rules to bisimulation congruences. In *Proceedings of CONCUR '98: Concurrency Theory (Nice). LNCS 1466*, pages 269–284, September 1998.

3.4 CWI

3.4.1 Research Directions

J.W. Klop and I. Bethke have worked on "origin tracking", both in the framework of first-order term rewriting and lambda calculus. For first-order rewriting using this technique a proof was given of the theorem of Huet and Levy on normalization of needed reductions. This technique is in the literature also called "dynamic dependence tracking" and is used for applications such as program slicing and error recovery. For lambda calculus a corresponding notion of origin tracking using Levy's labeled lambda calculus was formulated and used to give a new proof of the classical Genericity Lemma in lambda calculus.

3.4.2 Workshops, Travels and Visits

J.W. Klop attended the conference RTA 99 in Tsukuba, Japan, March 30 - April 1, 1998, where he gave an invited talk 'Descendants and Origins in Term Rewriting'.
J.W. Klop and I. Bethke attended the CONFER-2 Workshop in Brighton, April 13-16, 1998.
F. van Raamsdonk attended the CONFER-2 Workshop in Nice, September 14-15, 1998.

3.5 University of Edinburgh

3.5.1 Research directions

Abramsky, Honda and McCusker used game semantics to give the first fully abstract model for a language with higher-order functions and general reference types—a substantial fragment of "real" languages such as SML and SCHEME. Their paper appeared in LiCS '98.

Abramsky and McCusker wrote a survey paper based on Abramsky's lectures at Marktoberdorf '97 on the use of game semantics to give a precise account of the "space" of programming languages between purely functional languages, and languages extended with features such as references and non-local control constructs. This account is highly structured: the various language extensions are accounted for by relaxing some of the constraints on strategies imposed to capture the purely functional discipline, and the extended setting is related to the more restricted one by means of a factorization theorem. This paper will appear in the Proceedings of Marktoberdorf '97.

Abramsky and Melliès have developed a concurrent version of game semantics where player and opponent may play many tokens at each move, in a truly concurrent fashion. Contrary to traditional "sequential" games where interaction between strategies proceeds along *one* active thread, interaction in concurrent games involves *many* threads evolving concurrently, with the possibility of delays and synchronisations. This modelling ability of concurrent games allows to describe the whole of Linear Logic, and not just fragments as previously in sequential game semantics. Finally, a Full Completeness result for the Multiplicative-Additive fragment of Linear Logic was obtained for the model. The work was presented on various occasions, at the Linear Logic Workshop in Luminy in April 1998, at the MFPS conference at Queen Mary and Westfield College, London in May 1998, in September 1998 at the Linear workshop in Tende and at the CONFER-2 workshop in Nice. A paper containing the Full Completeness result was submitted to LiCS'99.

Melliès presented his result of stability in Axiomatic Rewriting Systems [AHM98] at the LiCS'98 conference held in Indianapolis, US.

3.5.2 Workshops, Travels and Visits

Travels Paul-André Melliès participated at the CONFER-2 Workshops held in Brighton and Nice.

3.5.3 Publications

[Mel98] P.-A. Melliès, Axiomatic Rewriting Theory IV A stability theorem in Rewriting Theory, *Proceedings of the 14th Annual Symposium on Logic in Computer Science,* Indianapolis, pp. 287-298, 1998. LICS'98

[AHM98] S. Abramsky, K. Honda and G. McCusker, A fully abstract game semantics for general references, *Proceedings of the 14th Annual Symposium on Logic in Computer Science,* Indianapolis, pp. 334–344, 1998. LICS '98

3.6 ENS

3.6.1 Research Directions

Game semantics has been the main topic of investigation at ENS.

In [BDER98], a first bridge between dynamic semantics and static semantics of functional languages is given. Specifically, it is shown that there is a lax functor from a variant of the AJM games model where morphisms are strategies [BDER97] to a variant of plain relational semantics which the authors call "bipolarized pointed relations" and where morphisms are relations. Both models are models for classical linear logic without weakening. Intuitively, that functor works in two steps: it first goes from the modeling of a strategy in terms of moves to a modeling in terms of positions, and then forgets the time ordering of positions.

This paper together with [DHR96] sets up a neat connection between operational semantics and denotational semantics.

One direction of further investigation might be the construction of that intermediate games model based on positions, the existence of which the functor above suggests. But additional structure seems to be needed, such as a coherence relation over positions.

Di Cosmo, Loddo, and Nicolet have introduced in [DCLN98] a semantics of Logic Programming based on classical Game Theory, which is proven to be sound and complete w.r.t. the traditional operational semantics and Negation as Failure. This approach opens the way to a better understanding of the mechanisms at work in parallel implementations of logic programs and in the operational semantics of logic programs with negative goals.

3.6.2 Publications

[BDER97] Patrick Baillot, Vincent Danos, Thomas Ehrhard, and Laurent Regnier. Believe it or not, AJM's games model is a model of linear logic. In *Proceedings of Logic in Computer Science '97*. IEEE, 1997.

[BDER98] Patrick Baillot, Vincent Danos, Thomas Ehrhard, and Laurent Regnier. Timeless games. In *Proceedings of Computer Science Logic '98*. Springer LNCS 1414, 1998.

[DCLN98] Roberto Di Cosmo, Jean-Vincent Loddo, and Stéphane Nicolet. A game semantics foundation for logic programming. In Catuscia Palamidessi, Hugh Glaser, and Karl Meinke, editors, *PLILP'98*, volume 1490, pages 355–373, 1998.

[DHR96] Vincent Danos, Hugo Herbelin, and Laurent Regnier. Games semantics and abstract machines. In *Proceeding of Logic in Computer Science '96*, 1996.

3.7 France Telecom – CNET

3.7.1 Research

In 1998, CNET investigated some aspects of low-level security in distributed programming languages, such as protecting hosts against potentially hostile mobile code in open environments. A popular solution involves performing static analysis on executable code.

Using the Coq proof-assistant, P. Brisset has studied the feasibility of a certified byte-code verifier for a small subset of the Java architecture while C. Charras investigated Carnegie Mellon University's Proof-Carrying Code technique. Security properties of implementations of higher-order concurrent process calculi could be formally verified using similar approaches.

3.7.2 Workshops

CNET participated to the two CONFER-2 workshops in Brighton and Nice. Pascal Brisset presented his results at the Nice Workshop, September 98.

3.8 ICL

3.8.1 Research Directions

The role of ICL in CONFER-2 is, through our work on designing programming languages and advanced end-user applications, to be serving as a “laboratory” in which theoretical results are transferred into industrial usage.

In the First Year Report of CONFER-2 we reported on a new research programme, called the ICL GTD[1] framework in Agents and Mobility. This framework has now been chosen as the example of how ICL sets up frameworks in general, hence it has been renamed to the ICL framework in Agents and Mobility. The programme itself still encourages ICL businesses to take part in the research process from an early stage to ensure they are prepared for knowledge and technology transfer as results emerge. Part of the framework is running a virtual laboratory in collaboration with Fujitsu Laboratories (recently Fujitsu have become the sole owner of ICL). The participation in CONFER-2 is an integral part of this framework.

Of particular interest to CONFER-2 is recent work on transferring ideas from the Facile system to the Java world via the mlj (ML to Java Virtual Machine code) compiler from Persimmon Research (http://research.persimmon.co.uk/mlj/). We have developed interfaces for the Facile concurrency primitives and are currently looking into how various packages for Mobile Agents in Java can be used to implement (some of) the constructs for distributed computing from Facile. Recently discussions of this work with Nick Benton from Microsoft Research and Andrew Kennedy from Persimmon Research have taken place in the context of the EU working group APPSEM 26142.

Work on understanding the foundation for mobile agent programming and mobile/distributed computing support in general is continuing to influence the development of ICL’s technical strategy in this area.

In addition to the technical work relevant to CONFER-2 Lone Leth and Bent Thomsen have assisted Jean-Jacques Levy in the management of the working group.

3.8.2 Exchange of Personnel

Lone Leth and Bent Thomsen participated in the 3rd CONFER-2 workshop in Brighton (University of Sussex) 14.04.98-16.04.98.

Lone Leth participated in the 4th CONFER-2 workshop in Nice (INRIA) 14.09.98-15.09.98. Raja Nagarajan from Imperial College also attended this workshop on behalf of ICL.

3.8.3 Publications

In the second year of CONFER-2 the group at ICL involved in the working group has produced two relevant publications:

[1] Group Technical Directorate

- J.-J. Levy, B. Thomsen, L. Leth: “First Year Report for Esprit Working Group 21836 CONFER-2 CONcurrency and Functions: Evaluation and Reduction”, EATCS Bulletin Number 64, February 1998, pp. 42-62.

- B. Thomsen and L. Leth Thomsen: “Towards global computations guided by concurrency theory”, EATCS Bulletin Number 66, October 1998, pp. 92-98.

3.9 INRIA-Rocquencourt

3.9.1 Research directions

INRIA Rocquencourt has worked in the following directions.

- Distributed Join-Calculus.

  We have 2 implementations of the join-calculus. Version 1.03 has been released in 1997 by Fournet and Maranget. This prototype includes a distributed implementation of the join-calculus, which is about 20000 lines of Caml-light code. The system comprises a byte-code compiler, a run-time interpreter, an on-line documentation, a tutorial and many examples. In June 1998, `jocaml` 1.07 has been released by le Fessant. It is an extension of the virtual machine of Objective Caml (Ocaml), which allows full compatibility with Ocaml. `jocaml` is much faster than our first implementation of the join-calculus, and more expressible, since it allows to freely use communication channels inside Caml or to send any Caml data-structure on a channel. Moreover, `jocaml` has a distributed garbage collector, able of collecting unused loops. As demo programs, we have a distributed calculation of the Mandelbrot set and several games. Several non-trivial Caml programs, such as a small editor or Hevea (a Latex to HTML translator) has been easily transformed into mobile programs.

  An article on the pattern-matching algorithm used in the join-calculus to synchronise communications has been presented at HLCL'98 in Nice [lFM98]. A paper by le Fessant on his distributed garbage collector has been presented at PLDI'98 [lF98a]. Another technique for garbage collection has been submitted to PODC'99.

  Both join-calculus systems are available at http://join.inria.fr/.

- Equivalences in the Join-Calculus

  The work started in 1997 on a new theory of bisimulation by Boreale (Roma), Fournet (INRIA) and Laneve (Bologna) has been presented at PROCOMET'98 [BFL98]. They consider open terms of the join calculus, and define two bisimulation, weak and asynchronous, which are both refinements of the observational equivalence. This gives a compositional sense to the interaction of the terms of the join-calculus with contexts, and is a start point to logical systems for reasoning on join-terms.

  Another work started in 1997 has been presented at ICALP'98 by Fournet and Gonthier [FG98]. One compared equivalences with fairness used both in the pi and join calculus, mostly based on reduction semantics, observation of barbs and preservation by contexts. Many variants appear according to the way these elementary constructs are combined. But fortunately many lead to a same result. One shows that fair-testing is connected to coupled bisimulation, or that omega-limit bisimulation is also related to single-barb gfp bisimulation. This provides new proof techniques and a better understanding of process calculi.

- Concurrency and Security

  Last years work on security has been presented at LICS'98 [AFG98a]. This work in collaboration with M. Abadi (Compaq SRC) gives inside the join-calculus semantics the correctness assertion for abstractions of secure cryptographic protocols. It is interesting in the new technique for proving equivalences of secure abstractions. A long version of the LICS'98 article is in prepartion [AFG98b].

- Objects

  This year, Fournet, Maranget, Laneve (Bologna) and Rémy started some work on distributed objects. The join-calculus allows straightforward implementation of objects with possibly exclusive or concurrent methods. But some work has to be done in order to add inheritance. Several solutions are feasible. A preliminary presentation of this work has been made at the CONFER-2 Workshop in Nice.

3.9.2 Persons and exchanges

INRIA Rocquencourt participants are Jean-Jacques Lévy, Fabrice le Fessant, Cédric Fournet, Georges Gonthier, Luc Maranget, and Didier Rémy. A new PhD student, Sylvain Conchon, joined our group this year.

Carolina Lavatelli left INRIA in December. Cédric Fournet passed his PhD in November, and works now at Microsoft Research laboratory in Cambridge.

From Cambridge, we had one-week visits by Luca Cardelli in April and by Philippa Gardner in September, and from Bologna a one-week visit by Cosimo Laneve in July. We also had a visit by Silvano Dal-Zilio.

3.9.3 Publications

[AFG98a] Martín Abadi, Cédric Fournet, and Georges Gonthier. Secure implementation of channel abstractions. In *Thirteenth Symposium on Logic in Computer Science (LICS '98, Indianapolis)*, June 1998, pages 105–116.

[AFG98b] Martín Abadi, Cédric Fournet, and Georges Gonthier. Secure implementation of channel abstractions. Manuscript, full version of this paper, on the Web at `http://pauillac.inria.fr/join`, 1998.

[BFL98] Michele Boreale, Cédric Fournet, and Cosimo Laneve. Bisimulations in the join-calculus. In *IFIP Working Conference on Programming Concepts, Methods and Calculi (PROCOMET' 98).* IFIP, June 1998.

[FG98] Cédric Fournet and Georges Gonthier. A hierarchy of equivalences for asynchronous calculi. In *Proc. of ICALP'98*, July 1998.

[lF98a] Fabrice Le Fessant, Ian Piumarta, and Marc Shapiro. An implementation for complete asynchronous distributed garbage collection. In *Proceedings of SIGPLAN'98 Conference on Programming Languages Design and Implementation*, ACM SIGPLAN Notices, Montreal, June 1998. ACM Press.

[Lav98] Carolina Lavatelli. Deadlock sensitive types for lambda calculus with resources. In *8th Conference FST & TCS 1998, Chennai, India*, 1998.

[lFM98] Fabrice Le Fessant and Luc Maranget. Compiling join-patterns. In Uwe Nestmann and Benjamin C. Pierce, editors, *HLCL '98: High-Level Concurrent Languages (Nice, France, September 12, 1998)*, volume 16.3 of *entcs*. Elsevier Science Publishers, 1998.

3.10 INRIA-Sophia

3.10.1 Research directions

The work conducted this year by the Sophia group can be divided into a part on semantics and expressiveness, and a part on objects and mobility.

3.10.2 Semantics and expressiveness

Milner has shown that the π-calculus can encode functions. Work by Boudol then showed that Milner's encodings can be seen as CPS (continuation passing style) transformations. In [STR98] Sangiorgi examines this issue in details, analysing different translations of the λ-calculus into π-calculus and their relationship with CPS transformations. This work is also a tutorial paper on encodings of λ-calculus into π-calculus and on the equality among functions that is induced by their representation as π-calculus processes. This question is also analysed by Boudol [Bou98], who shows that, in the case of call-by-name λ-calculus, the same equality among functions can also be obtained from a CPS transformation.

Massimo Merro [ME98] has shown that the asynchronous π-calculus is powerful enough to encode asynchronous variants of Chi, Update, and Fusion calculi, which are modifications of the π-calculus (Update and Fusion were developed within CONFER by the Swedish group) in which communications have a more global scope.

Recently there has been a growing interest in asynchronous languages and process calculi, where message emission is non-blocking. An important reference for this class of languages is the asynchronous pi-calculus, introduced by Honda and Tokoro and independently by Boudol, whose descriptive power has now been largely demonstrated. Castellani and Hennessy [CH98] study testing semantics for asynchronous process calculi. They examine an asynchronous version of CCS with internal and external choice operators. On this language asynchronous versions of the may and must testing preorders originally introduced by De Nicola and Hennessy are defined. Alternative characterisations for these preorders are given, which are contextual in nature. An equational characterisation for both preorders on the finite fragment of the language is also presented.

Boreale and Sangiorgi [BS97] study barbed equivalence in name-passing languages where there is no matching construct for testing equality between names. They concentrate on the π-calculus with capability types and subtypes, of which the untyped π-calculus without matching is a special case. The techniques introduced are used to prove some process equalities that fail in the ordinary π-calculus.

The asynchronous π-calculus is considered the basis of experimental programming languages (or proposal of programming languages) like Pict, Join, and Blue calculus. However, at a closer inspection, these languages are based on an even simpler calculus, called Local π, where: (a) only the output capability of names may be transmitted; (b) there is no matching or similar constructs for testing equality between names. In Local π, as reported by Sangiorgi and Merro [MeSa98], a number of useful equalities that are not sound in the standard π-calculus, do become sound. Proof techniques for proving such equalities are studied in [MeSa98].

3.10.3 Objects and mobility

In [KlSa98] Kleist and Sangiorgi give an interpretation of Abadi and Cardelli's first-order Imperative Object Calculus into a typed π-calculus. The interpretation validates the subtyping relation and the typing judgements of the Object Calculus, and is computationally adequate. The proof of computational adequacy makes use of (a π-calculus version) of ready simulation, and of a factorisation of the interpretation into a functional part and a very simple imperative part. The interpretation can be used to compare and contrast the Imperative and the Functional Object Calculi, and to prove properties about them, within a unified framework.

This work is being continued by Hüttel, Kleist, Merro, and Nestman [HKMN98] where a modelisations of the core of Cardelli's Obliq language is studied. The additional important problem here is to model the primitives that allow one to program the migration of objects, and to show their correctness.

Silvano Dal-Zilio [S98b] uses the modelisation of Abadi and Cardelli objects in the blue calculus extended with a very simple system of localities. It is shown that two migration behaviours can be defined, namely those of bouncing and quiet objects. These migration control abstractions are defined separately from the other aspects of the object definition and can easily be reused, thus providing more flexibility in the definition of "migration constraints". The interpretation of objects in the blue calculus is described in detail in [S98a].

3.10.4 Participation to Workshops

R. Amadio, I. Castellani, M. Merro and S. Dal-Zilio attended the CONFER-2 workshop in Brighton.

R. Amadio, G. Boudol, I. Castellani, M. Merro, D. Sangiorgi and S. Dal-Zilio attended the CONFER-2 workshop in Nice.

3.10.5 Publications

[AC98] R. Amadio, S. Coupet-Grimal, Analysis of a guard condition in type theory (extended abstract) , *in: Proc. FOSSACS, ETAPS 98, Springer Lect. Notes in Comp. Sci. 1378*, p. 48–62, 1998.

[AP98] R. Amadio, S. Prasad, Modelling IP mobility , *in: Proc. CONCUR98, Springer Lect. Notes in Comp. Sci. 1466*, p. 301–316, 1998.

[BS97] M. Boreale, D. Sangiorgi, Bisimulation in name-passing calculi without matching , *in: thirteen LICS Conf.*, IEEE Computer Society Press, 1998.

[Bou98] G. Boudol, On the semantics of the call-by-name CPS-transform , available at the author's web page, 1998, submitted.

[CH98] I. Castellani, M. Hennessy, Testing theories for asynchronous languages , *in: Proceedings FST-TCS'98*, 1998. To appear

[HKMN98] H. Hüttel, J. Kleist, M. Merro, and U. Nestmann. Migration = cloning ; aliasing. To be presented at Sixth Workshop on Foundations of Object-Oriented Languages (FOOL 6), 1999.

[KlSa98] J. Kleist, D. Sangiorgi, Imperative Objects and Mobile Processes , *in: Proc. IFIP Working Conference on Programming Concepts and Methods (PROCOMET'98)*, North-Holland, 1998.

[MeSa98] M. Merro, D. Sangiorgi, On asynchrony in name-passing calculi , *in: 25th ICALP*, *LNCS*, *1443*, Springer Verlag, 1998.

[ME98] M. Merro, On the Expressiveness of Chi, Update and Fusion Calculi , Electronic Notes in Theoretical Computer Science Volume 16 , Issue 2 1998.

[S98a] S. Dal-Zilio, Objets concurrents dans un π-calcul applicatif , *in: accepté par les Journées Francophones des Langages Applicatifs*, 1998.

[S98b] S. Dal-Zilio, Quiet and Bouncing Objects: Two Migration Abstractions in a Simple Distributed Blue Calculus , *in: 1st International Workshop on Semantics of Objects as Processes*, BRICS Notes Series, 1998.

[STR98] D. Sangiorgi, Interpreting functions as pi-calculus processes: a tutorial , *rapport de recherche NoRR-3470*, INRIA-Sophia Antipolis, 1998.

3.11 KTH

3.11.1 Research directions

We have developed the fusion calculus which simplifies and extends the pi-calculus.

The fusion calculus contains the polyadic pi-calculus as a proper subcalculus and thus inherits all its expressive power. In addition fusion contains actions akin to updating a shared state, and a scoping construct for bounding their effects. Therefore it is easier to represent computational models with shared state, including concurrent constraint formalisms. It is also easy to represent the so called strong reduction strategies in the lambda-calculus, involving reduction under abstraction. In the pi-calculus these tasks require elaborate encodings.

The fusion calculus simplifies the pi-calculus by reducing the number of binding operators and the number of bisimulation equivalences, and by making input and output symmetric like in pure CCS. We attain a calculus where concepts from other models of computation are more easily expressed than in the pi-calculus, thereby taking a step towards a unified yet simple model of computation.

We have developed a broad foundational theory of the fusion calculus. We defined its labelled and unlabelled operational semantics, and treated strong and weak bisimulation equivalences for both semantics in some detail, including complete axiom systems for finite terms. The equivalences were given symbolic characterizations, leading to algorithms and an automatic tool for equivalence checking. We demonstrated the expressive power of the fusion calculus to give simple encodings of foundational calculi for functional and concurrent constraint programming.

3.11.2 Persons and exchanges

During 1998, the following persons have been active within CONFER-2: Joachim Parrow and Björn Victor (Uppsala University). Parrow attended the CONFER workshop in Brighton, and Victor attended the workshop in Nice.

3.11.3 Publications

[PV98a] J. Parrow and B. Victor. The fusion calculus: Expressiveness and symmetry in mobile processes. In *Proceedings of LICS '98*, 1998.

[PV98b] J. Parrow and B. Victor. The tau-laws of fusion. In D. Sangiorgi and R. de Simone, eds, *Proceedings of CONCUR '98*, volume 1466 of *LNCS*, pages 99–114. Springer, Sept. 1998.

[Vic98a] B. Victor. *The Fusion Calculus: Expressiveness and Symmetry in Mobile Processes.* PhD thesis, Department of Computer Systems, Uppsala University, June 1998. Available as report DoCS 98/98.

[VP98] B. Victor and J. Parrow. Concurrent constraints in the fusion calculus. In K. G. Larsen, S. Skyum and G. Winskel, eds, *Proceedings of ICALP '98*, volume 1443 of *LNCS*, pages 455–469. Springer, July 1998.

3.12 Università di Pisa

3.12.1 Outline

The major line of research of the group at the Dipartimento di Informatica, Università di Pisa, has been in the areas of *Calculi*, *Foundational Models and Abstract Machines* and *Programming Languages*.

- *Calculi*

  The History-Dependent automata (HD-Automata) model has been further developed. HD-automata have shown to be well suited to represent some key elements (name generations and name handling) of the operational semantics of name passing process calculi. HD-automata come equipped with a bisimulation-based observational semantics that captures ordinary bisimulation semantics for name passing process calculi.

  Research has continued on developing efficient finite-state techniques for verifying properties of calculi for mobile processes (e.g. synchronous and asynchronous π-calculus) whose behaviours are modelled in terms of HD-automata. The prototype version of the semantic-based verification environment, the HAL environment (*History-dependent Automata Laboratory*), has been developed. The HAL environment supports verification of both behavioural properties (bisimulation checkers) and logical formulae (model checkers) of π-calculus specifications.

  A higher order presentation of the π-calculus based on typed λ-calculus has been introduced. Such a presentation highlights the nature of the binding operators and elucidates the role of free and bound channels. As a result, the final semantics of the π-calculus is defined making use of this higher order presentation within a category of *hypersets*.

  The description of a software architecture style must include the structural model of the components and their interactions, the laws governing the dynamic changes in the architecture, and the communication pattern. An approach where a system is as a graph where hyperedges are components and nodes are ports of communication have been introduced. The construction and dynamic evolution of the style will be represented as context-free productions and graph rewriting. To model the evolution of the system we adopt techniques of constraint solving. From this approach we obtain an intuitive way to model systems with nice characteristics for the description of dynamic architectures and reconfiguration and, a unique language to describe the style, model the evolution of the system and prove properties.

- *Foundational Models and Abstract Machines*

  Research has continued on developing the mathematical foundations of mobile systems based on the tile logic. The natural targets of the tile approach are models of mobile systems which are compositional in both space and time. The tile logic provides a logic of concurrent systems with conditional state changes and synchronization. Tile logic has been shown to be particularly useful for defining operational formats for calculi of mobile processes and causal and located concurrent processes. A tile-based model for a calculus which integrates actors and π-calculus has been introduced. There, coordination primitives are defined on to the union of the two calculi and allow actors and π-agents to cooperate. The tile model has been used for modelling (Petri-Net based) coordination models equipped with flexible and powerful composition mechanisms (e.g. synchronization primitives).

Two categorical notions, namely, *symmetric strict monoidal double category* and *cartesian double category with consistently chosen products* have been introduced. They provide an adequate semantic setting for the two variants of the tile logic (process and term tile logic). The new model theory of *2EVH-categories*, required to relate the categorical models of tile logic and rewriting logic, has also been introduced. As a result of this comparison, a correct rewriting implementation of tile logic has been obtained. This implementation uses a meta-layer to control the rewritings, so that only tile proofs are accepted. Making use of the reflective capabilities of the Maude language, some (general) internal strategies are then defined to implement the mapping from tile systems into rewriting systems, and some interesting applications related to the implementation of processes calculi have been developed.

- *Programming Languages*

  We introduced the language KLAIM (*Kernel Language for Agents Interaction and Mobility*). KLAIM is an experimental programming language, inspired by the Linda paradigm, where mobile agents and their coordination strategies can be naturally programmed. KLAIM provides direct support for expressing and enforcing security policies that control access to resources and data. In particular, the language uses types to protect resources and data and to establish policies for access control.

  The main guidelines for the design of KLAIM and of its type system are:

  – KLAIM processes and types are *network* aware;

  – Networks and processes are different entities;

  – Access control policies are network coordination policies.

  A prototype implementation of the language has also been developed. Paradigmatic examples of mobile code applications (e.g. Electronic Commerce and Distributed Retrieval) have been implemented in KLAIM.

3.12.2 Perspective

We are interested to further develop theoretical foundations of HD-automata and the corresponding verification techniques. One aspect that we intend to study is the development of verification methods for checking observational (bisimulation) semantics for classes of mobile calculi.

The format of the tile rules is essential in proving important properties of the system. For instance, a main proof method is given by the ability to vertically or horizontally decompose a derivable tile into more basic derivable tiles. This decomposition is driven by a decomposition of the effect or of the initial configuration, which in turn is possible if every rule (in a certain class) has an atomic initial configuration or an atomic effect. Among the properties provable in this way there are also important results such as "bisimulation is a congruence", as shown in for the tile system for CCS. It would be interesting to develop similar results for general tile systems, thus exploiting the powerful rule induction intrinsic in the tile model.

On the application side, we would like to mention the graphical orientation of tiles. While no graphical support for tiles is currently available, we think that the (natural) wire-and-box representation of tiles is (after some training) expressive and suggestive. It could be useful for users with a limited theoretical background, or for applications where simplicity and flexibility is essential, such as software architectures.

We intend to further investigate notions of types for the KLAIM language. We aim at demonstrating that network coordination policies based on typing information can be systematically used to guarantee that well–typed KLAIM processes always enjoy classes of security properties.

3.12.3 Persons

The group at Pisa involved in the project consists of Ugo Montanari, Gianluigi Ferrari, Marco Pistore and Roberto Bruni.

Two PhD-students, Marco Pistore and Roberto Bruni, both under the supervision of Ugo Montanari, are going to present their PhD thesis on topics which are closely related to the aims of the CONFER-2 Working Group.

3.12.4 Networking

Dates	Place	Participant
28/3/98	Lisboa (Portugal) ETAPS'98 Conf.	Montanari U.
13/4/98	Brighton (UK) CONFER-2 Workshop	Ferrari G.
13/4/98	Brighton (UK) CONFER-2 Workshop	Montanari U.
13/4/98	Brighton (UK) CONFER-2 Workshop	Pistore M.
6/9/98	Nice (France) CONCUR'98 & CONFER-2	Pistore M.
6/9/98	Nice (France) CONCUR'98	Montanari U.
13/9/98	Nice (France) CONFER-2	Montanari U.

3.12.5 Publications

1. Bettini, L., De Nicola, R., Ferrari, G., Pugliese, R. Interactive Mobile Agents in XKLAIM. *WETICE'98*, *Proceedings*, IEEE, 1998, to appear.

2. Bruni, R., A Logic for Modular Descriptions of Asynchronous and Synchronized Concurrent Systems (abstract) in Proceedings 2nd International Workshop on Rewriting Logic and its Applications, (Kirchner C., Kirchner H., Editors), ENTCS 15 (1998).

3. Bruni R., Gadducci F., and Montanari U. Normal Forms for Partitions and Relations in Proceedings 13th Workshop on Algebraic Development Techniques, LNCS, to appear.

4. Bruni, R. and Montanari, U. Zero-Safe Nets: The Individual Token Approach in Proceedings 12th Workshop on Algebraic Development Techniques, (Parisi-Presicce F., Editor), LNCS 1376, pp. 122–140 (1998).

5. Bruni R., Meseguer J. and Montanari U. Process and Term Tile Logic Technical Report SRI-CSL-98-06, SRI International (July 1998). Also Technical Report TR-98-09, Department of Computer Science, University of Pisa (1998).

6. Bruni R., Meseguer J. and Montanari U. Internal Strategies in a Rewriting Implementation of Tile Systems in Proceedings 2nd International Workshop on Rewriting Logic and its Applications, (Kirchner C., Kirchner H., Editors), ENTCS 15 (1998).

7. Bruni R., Meseguer J. and Montanari U. Implementing Tile Systems: Some Examples from Process Calculi in Proceedings 6th Italian Conference on Theoretical Computer Science, World Scientific, 1998.

8. Bruni R., Meseguer J., Montanari U. and Sassone V., A Comparison of Petri Net Semantics under the Collective Token Philosophy in Proceedings 4th Asian Computing Science Conference, (Hsiang J., Ohori A., Editors), LNCS, to appear.

9. Bruni, R., Montanari, U., Zero-Safe Nets: Comparing the Collective and Individual Token Information and Computation, to appear.

10. De Nicola, R., Ferrari, G., Pugliese, R., Klaim: a Kernel Language for Agents Interaction and Mobility. *IEEE Transactions on Software Engineering*, Vol.24(5):315-330, May, 1998.

11. Ferrari,G., Gnesi, S., Montanari, U., Pistore, M. and Ristori, G. Verifying Mobile Processes in the HAL Environment, In Proc. CAV'98, LNCS 1427. Springer Verlag, 1998.

12. Ferrari, G. and Montanari, U., Tile Formats for Located and Mobile Systems, To appear in Information and Computation, 1998.

13. Honsell, F., Lenisa, M., Montanari, U., Pistore, M. Final Semantics for the pi-Calculus, In Proc. PROCOMET'98, 1998.

14. Hirsch, D., Inverardi, P., and Montanari, U., Modeling Software Architectures and Styles with Graph Grammars and Constraint Solving, to appear in Proceedings WICSA 98, 1998.

15. Montanari, U. and Meseguer, J., Mapping Tile Logic into Rewriting Logic, in: Francesco Parisi-Presicce, Ed., Recent Trends in Algebraic Development Techniques, Spinger LNCS 1376, 1998, pp. 62-91. i

16. Montanari , U. and Pistore, M. An Introduction to History Dependent Automata In Proc. HOOTS'98, ENTCS 10. Elsevier Science, 1998.

17. Montanari , U. and Talcott, C., Can Actors and pi-Agents Live Together?, in: Andrew Gordon, Andrew Pitts and Carolyn Talcott, Eds, Second Workshop on Higher-Order Operational Techniques in Semantics (HOOTS II), ENTCS, Vol. 10, 1998.

3.13 University of Sussex

3.13.1 Research directions

Asynchronous Mobile Processes In this area we have worked on two lines of research. The first concerns the expressivity of the various programming constructs which comprise typical mobile calculi. Specifically we have investigated a particular finite set of ”combinators” which are sufficiently powerful to describe all (asynchronous) pi-calculus terms. Moreover we have shown that this set is minimal, in the sense that if any one combinator is omitted then there are pi-calculus terms which are no longer expressible. The research is reported in [Yos98]

In the second line we have proposed a new operational model for asynchronous processes, based on certain collapsings of traditional transition systems. This leads to a new class of processes, which we call “asynchronously regular processes”. Using the new models we provide two appealing abstract characterisations of asynchronous bisimulation equivalence, namely, as spans of open maps and as a winning strategies for a bisimulation game. Also, by exploiting the coincidence of finite graphs with regular processes we see that bisimulation is polynomial time decidable over our class of asynchronously regular processes. The details are given in [Rat98].

Types for Distributed Processes Here we have continued our line of research started in the previous report period of CONFER-2. In [RH98] we propose a distributed version of the pi-calculus, in which mobile agents may migrate from site to site in a distributed environment. The novelty here is the use of a system of types, which may be used to manage access to resources, such as memory or communication channels. We prove that if all agents are well-typed then, in some intuitive sense, no agent can have access to resources unless they have been specifically granted permission. This work is continued in [HR98], where we now assume that there may be malicious agents, agents which do not respect the type system. By introducing a system of dynamic type-checking, where incoming agents are checked below allowed access to a site, we show that good site, which respect the type system, can not be harmed by malicious sites.

3.13.2 Personnel and Exchanges

The personnel at the University of Sussex who were actively involved in CONFER-2 related research during the period covered by the progress report are Matthew Hennessy, Julian Rathke, James Riely and Nobuko Yoshida.

3.13.3 Publications

[HR98] Matthew Hennessy and James Riely. Type-safe execution of mobile agents in anonymous networks (extended abstract). In *Proceedings of 4th*

Workshop on Mobile Onbject Systems, Brussels, Belgium, July 1998. Full version available as Computer Science Technical Report 03/98. Available from http://www.cogs.susx.ac.uk/.

[Rat98] J. Rathke. Resource based models for asynchrony. In *Proc. FOSSACS'98, International Conference on Foundations of Software Science and Computation Structures, Lisbon.* Springer-Verlag, 1998.

[RH98] James Riely and Matthew Hennessy. Resource access control in systems of mobile agents (extended abstract). In *Proceedings of 3rd International Workshop on High-Level Concurrent Languages*, Nice, France, September 1998. Full version available as Computer Science Technical Report 02/98. Available from http://www.cogs.susx.ac.uk/.

[Yos98] Nobuko Yoshida. Minimality and separation results on asynchronous mobile processes: representability theorem by concurrent combinators (extended abstract). In *Proceedings of CONCUR98*, Nice, France, September 1998. Full version available as Computer Science Technical Report 05/98. Available from http://www.cogs.susx.ac.uk/.

3.14 University of Warwick, Oxford University

3.14.1 Research activity

Work has been done on theory of π-calculus, and the final versions of papers on semantic definition of concurrent object-oriented programming languages and correctness of program-transformation rules have been published in journals.

In the monadic π-calculus an interaction between processes involves the transmission of a single name, while in the polyadic π-calculus a tuple of names is passed in an interaction. Communication of tuples of names is expressible in the monadic calculus. Moreover, although passing of an n-tuple requires $n + 1$ monadic interactions, only the first is semantically significant: in the first action a private link is established between the communicating processes; the subsequent semantically-inert communications transfer the names of the tuple from sender to receiver via that link.

In the polyadic calculus there is a pressing need for some kind of typing discipline, as among the processes are terms where the components disagree on the length of the tuple to be passed. Even on well-typed polyadic processes, however, the translation to monadic processes is not fully abstract when barbed congruence is adopted as process equivalence. Equivalence of the monadic translations of well-typed polyadic processes implies equivalence of the polyadic processes themselves, but the converse does not hold, the reason being, briefly, that there are monadic contexts that do not conform to the interaction regime that underlies the translation.

In [QW98] a typing system for monadic processes that captures the interaction regime underlying the translation is introduced and used to obtain a full-abstraction result: two well-typed polyadic processes are barbed congruent if and only if their monadic translations are barbed equivalent in all well-typed monadic contexts. The monadic typing system combines and extends ideas present in other typing systems for π-calculus processes.

The papers [LW98] and [PW98] give a detailed account of a theory of partial confluence of mobile processes, together with applications of the theory to correctness of transformation rules for concurrent object-oriented programs.

The main theorems of [PW98] concern systems which can be thought of as composed from two components: a questioner, and an answerer which may process questions concurrently. One result gives conditions under which a system of this kind is behaviourally indistinguishable from the corresponding system in which the answerer may handle only one question at a time. This result is used in conjunction with a semantic definition for a concurrent object-oriented programming language by translation to a mobile-process calculus to show the interchangeability in an arbitrary program context of two class definitions which prescribe binary-tree symbol-table data structures, one supporting concurrent operations, the other sequential.

In [LW98], this work is extended to show the soundness of transformation rules for concurrent object-oriented programs which increase the scope for concurrent activity within the system prescribed by a program without altering its observable behaviour. The rules allow one object to release another from a rendezvous before it has finished all the associated activity, and allow one object to delegate to a second object the responsibility for returning a result to a third object.

3.14.2 Workshops and site visits

Working-Group funds were used to support the participation in the Brighton workshop in April 1998 of Guy McCusker, Corin Pitcher, and David Walker. They also supported David Walker's participation in the Sophia Antipolis workshop in September 1998 which was combined with a short site visit to INRIA where the main focus of interaction was joint work on π-calculus with Davide Sangiorgi.

3.14.3 Publications

[QW98] P. Quaglia and D. Walker, *On encoding $p\pi$ in $m\pi$*, in Proceedings of Foundations of Software Technology and Theoretical Computer Science, December 1998, Springer-Verlag Lecture Notes in Computer Science, to appear.

[LW98] X. Liu and D. Walker, *Partial confluence of processes and systems of objects*, Theoretical Computer Science 206 (1998) 127–162.

[PW98] A. Philippou and D. Walker, *On transformations of concurrent object programs*, Theoretical Computer Science 195 (1998) 259–289.

Chapter 4

Deliverables

4.1 Workshop 3

The Third workshop was held in Brighton, organised by University of Sussex, with more than 30 participants. Luca Cardelli and Andrew Gordon, Microsoft Research, were invited.

Tuesday, April 17th

9.30 - 10.10	Resource access control in systems of mobile agents, Matthew Hennessy
10.30 - 11.10	Origin tracking in orthogonal term rewriting systems, J.W. Klop
11.10 - 11.50	Tile Formats for Located and Mobile Systems, Gianluigi Ferrari
12.45-14.00	Lunch
14.00 - 14.40	A Concurrent Object Calculus, Andy Gordon
14.40 - 15.20	Bisimulation congruences for Action Calculi, James Leifer
15.40 - 16.20	From Rewrite Rules to Bisimulation Congruences, Peter Sewell
16.20 - 17.00	A conceptual view of weak bisimulation, Marcelo Fiore

Wednesday, April 18th

9.30 - 10.10	Type systems for Ambients, Luca Cardelli
10.30 - 11.10	Compilation of Higher-Order Languages in Graphical Form, Ralf Schweimeier
11.10 - 11.50	Concurrent objects in the blue calculus, Silvano Dal-Zilio
12.00-14.00	Lunch
14.00 - 14.40	Modelling IP Mobility, Roberto Amadio
14.40 - 15.20	Verifying Mobile Processes in the HAL Environment, Marco Pistore
15.40 - 16.20	tba, Ugo Montanari
16.20 - 17.00	Semantics of active networks Alan Jeffrey

Wednesday, April 19th

9.30 - 10.10	The fusion calculus Joachim Parrow
10.30 - 11.10	Games semantics for call-by-value computation, Kohei Honda
11.10 - 11.50	Resource based models for asynchrony, Julian Rathke
12.00-14.00	Lunch

List of Participants: Roberto Amadio, Andrea Asperti, Inge Bethke, Gerard Boudol, Pascal Brisset, Roberto Bruni, Ilaria Castellani, Silvano Dal-Zilio, Vincent Danos, Gianluigi Ferrari, Marcelo Fiore, Philippa Gardner, Matthew Hennessy, Kohei Honda, Alan Jeffrey, Ole Hoegh Jensen, Jan Willem Klop, Cosimo Laneve, Carolina Lavatelli, James Leifer, Fabrice le Fessant, Lone Leth, Jean-Jacques Lévy, Luc Maranget, Paul-Andre Melliès, Jean-Francois Monin, Uwe Nestmann, Martin Odersky, Catuscia Palamidesi, Joachim Parrow, Marco Pistore, Femke van Raamsdonk, Paola Quaglia, Julian Rathke, Didier Remy, Peter Sewell, Bent Thomsen.

4.2 Workshop 4

The Fourth workshop was held in Nice, organised by INRIA Sophia Antipolis, with 30 participants, at the end of a long week with Express, Concur, HLCL and CONFER-2.

Martin Odersky, University of South Wales, was invited.

Monday, September 14

9h - 9h45	The Join Calculus from a Functional Perspective Martin Odersky
9h45 - 10h30	From the Join-Calculus to a Class-Based, Concurrent OO Language Didier Remy
10h30 - 11h	pause
11h - 11h30	Communication Infrastructure for Mobile Agents: Calculi for Algorithms Peter Sewell
11h30 - 12h	Fusion Particles: Symmetrically Asynchronous Mobile Processes Bjorn Victor
12.00-14.00	Lunch
14h - 14h45	Static Analysis of Concurrent Programs, with Applications (application to the Ariane 5 on-board programs) Alain Deutsch
14h45 - 15h30	Final Semantics for the Pi-Calculus Ugo Montanari
15h30 - 16h	pause
16h - 16h45	Towards Certified Byte-Code Verification and Proof-Carrying Code Pascal Brisset

Tuesday, September 15

9h - 9h45	Finite State Verification for the Asynchronous Pi-Calculus Marco Pistore
9h45 - 10h30	Bisimulation and Confluence Cedric Fournet
10h30 - 11h	pause
11h - 11h30	A stability theorem with games Paul-Andre Mellies

List of Participants: Andrea Asperti, Inge Bethke, Gerard Boudol, Pascal Brisset, Ilaria Castellani, Sylvain Conchon, Pierre-Louis Curien, Silvano Dal-Zilio, Vincent Danos, Fabrice le Fessant, Gianluigi Ferrari, Cedric Fournet, Georges Gonthier, Matthew Hennessy, Alan Jeffrey, Jan Willem Klop, Cosimo Laneve, Carolina Lavatelli, James Leifer, Fabrice le Fessant, Lone Leth, Jean-Jacques Levy, Jean Loddo, Luc Maranget, Paul-André Melliès, Massimo Merro, Jean-Francois Monin, Raja Nagarajan, Uwe Nestmann, Femke van Raamsdonk, Peter Sewell, Bjorn Victor

4.3 Software deliverables

The following is a listing of software constructed during the second year of CONFER-2:

- PICT
  Cambridge — Typed higher-order programming language based on π-calculus
  University of Cambridge and Edinburgh — B. Pierce, D. Turner.
  A distributed version is under construction by P. Sewell, P. Wojciechowski.
  Available at http://www.cs.indiana.edu/hyplan/pierce/ftp/pict

- The Join-Calculus
  INRIA-Rocquencourt — C. Fournet, L. Maranget.
  Also with contributions of G. Gonthier, J.-J. Lévy, D. Rémy.
  Available at http://join.inria.fr

- `jocaml`
  INRIA-Rocquencourt — F. le Fessant.
  Available at http://join.inria.fr

Chapter 5

Appendices

[AM98] A. Asperti, H.Mairson *Optimal β-reduction is not elementary recursive.* Proc. of the twenty-fifth Annual ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL'98).

[AG98] A. Asperti, S.Guerrini *The Optimal Implementation of Functional Programming Languages.* in *"Cambridge Tracts in Theoretical Computer Science"* Series, Cambridge University Press, October 1998.

[BFL97] M. Boreale, C. Fournet, C. Laneve: *Bisimulations in the Join-Calculus.* In Proc. Programming Concepts and Methods, Procomet '98, New York, U.S.A., 1998.

[FLMR98] C. Fournet, C. Laneve, L. Maranget, D. Rèmy: *Inheritance in the Join-Calculus.* Draft, October 1998.

[BGHP98] A. Barber, P.A. Gardner, M. Hasegawa and G. Plotkin, From Action Calculi to Linear Logic. *European Association of Computer Science Logic*, Springer-Verlag, 1998; conference article appears in BRICS Notes Series NS-97-1, 1997.

[CFW98] G. L. Cattani, M. Fiore and G. Winskel. A Theory of Recursive Domains with Applications to Concurrency. Proceedings of the Thirteenth Annual IEEE Symposium on Logic in Computer Science, LICS '98, 214–225.

[CPW98] G. L. Cattani, A. J. Power and G. Winskel. A Categorical Axiomatics for Bisimulation In *Proceedings of CONCUR '98: Concurrency Theory (Nice). LNCS 1466*, pages 581–596, September 1998.

[GH98] P.A. Gardner and M. Hasegawa. *Higher-order and Reflexive Action Calculi: their type-theory and models*, submitted to Information and Computation, 1998.

[Lei98] James Leifer. Bisimulation Congruences for Action Calculi. Thesis submitted to Trinity College, Cambridge, 1998.

[Mil99] Robin Milner. Communicating and Mobile Systems: the Pi Calculus. Forthcoming (CUP Jan 1999)

[PCW98] A. J. Power, G. L. Cattani and G. Winskel. A Representation Result for Free Cocompletions. BRICS Technical report RS-98-21, Department of Computer Science, University of Aarhus, September 1998. Submitted for publication.

[SWP98a] Peter Sewell, Paweł T. Wojciechowski, and Benjamin C. Pierce. Location independence for mobile agents. In *Proceedings of the Workshop on Internet Programming Languages (Chicago)*, May 1998.

[SWP98b] Peter Sewell, Paweł T. Wojciechowski, and Benjamin C. Pierce. Location-independent communication for mobile agents: a two-level architecture. Submitted for publication.

[Sew98a] Peter Sewell. Global/local subtyping and capability inference for a distributed π-calculus. In *Proceedings of ICALP '98: International Colloquium on Automata, Languages and Programming (Aarhus). LNCS 1443*, pages 695–706. Springer-Verlag, July 1998.

[Sew98b] Peter Sewell. From rewrite rules to bisimulation congruences. Technical Report 444, University of Cambridge, June 1998.

[Sew98c] Peter Sewell. From rewrite rules to bisimulation congruences. In *Proceedings of CONCUR '98: Concurrency Theory (Nice). LNCS 1466*, pages 269–284, September 1998.

[Mel98] P.-A. Melliès, Axiomatic Rewriting Theory IV A stability theorem in Rewriting Theory, *Proceedings of the 14th Annual Symposium on Logic in Computer Science,* Indianapolis, pp. 287-298, 1998. LICS'98

[AHM98] S. Abramsky, K. Honda and G. McCusker, A fully abstract game semantics for general references, *Proceedings of the 14th Annual Symposium on Logic in Computer Science,* Indianapolis, pp. 334–344, 1998. LICS '98

[BDER97] Patrick Baillot, Vincent Danos, Thomas Ehrhard, and Laurent Regnier. Believe it or not, AJM's games model is a model of linear logic. In *Proceedings of Logic in Computer Science '97.* IEEE, 1997.

[BDER98] Patrick Baillot, Vincent Danos, Thomas Ehrhard, and Laurent Regnier. Timeless games. In *Proceedings of Computer Science Logic '98.* Springer LNCS 1414, 1998.

[DCLN98] Roberto Di Cosmo, Jean-Vincent Loddo, and Stéphane Nicolet. A game semantics foundation for logic programming. In Catuscia Palamidessi, Hugh Glaser, and Karl Meinke, editors, *PLILP'98*, volume 1490, pages 355–373, 1998.

[DHR96] Vincent Danos, Hugo Herbelin, and Laurent Regnier. Games semantics and abstract machines. In *Proceeding of Logic in Computer Science '96*, 1996.

[LTL98] J.-J. Levy, B. Thomsen, L. Leth: "First Year Report for Esprit Working Group 21836 CONFER-2 CONcurrency and Functions: Evaluation and Reduction", EATCS Bulletin Number 64 February 1998 pp. 42-62.

[TLT98] B. Thomsen and L. Leth Thomsen: "Towards global computations guided by concurrency theory", EATCS Bulletin Number 66 October 1998 pp. 92-98.

[AFG98a] Martín Abadi, Cédric Fournet, and Georges Gonthier. Secure implementation of channel abstractions. In *Thirteenth Symposium on Logic in Computer Science (LICS '98, Indianapolis)*, June 1998, pages 105–116.

[AFG98b] Martín Abadi, Cédric Fournet, and Georges Gonthier. Secure implementation of channel abstractions. Manuscript, full version of this paper, on the Web at `http://pauillac.inria.fr/join`, 1998.

[BFL98] Michele Boreale, Cédric Fournet, and Cosimo Laneve. Bisimulations in the join-calculus. In *IFIP Working Conference on Programming Concepts, Methods and Calculi (PROCOMET' 98)*. IFIP, June 1998.

[FG98] Cédric Fournet and Georges Gonthier. A hierarchy of equivalences for asynchronous calculi. In *Proc. of ICALP'98*, July 1998.

[lF98a] Fabrice Le Fessant, Ian Piumarta, and Marc Shapiro. An implementation for complete asynchronous distributed garbage collection. In *Proceedings of SIGPLAN'98 Conference on Programming Languages Design and Implementation*, ACM SIGPLAN Notices, Montreal, June 1998. ACM Press.

[Lav98] Carolina Lavatelli. Deadlock sensitive types for lambda calculus with resources. In *8th Conference FST & TCS 1998, Chennai, India*, 1998.

[lFM98] Fabrice Le Fessant and Luc Maranget. Compiling join-patterns. In Uwe Nestmann and Benjamin C. Pierce, editors, *HLCL '98: High-Level Concurrent Languages (Nice, France, September 12, 1998)*, volume 16.3 of *entcs*. Elsevier Science Publishers, 1998.

[AC98] R. Amadio, S. Coupet-Grimal, Analysis of a guard condition in type theory (extended abstract) , *in: Proc. FOSSACS, ETAPS 98, Springer Lect. Notes in Comp. Sci. 1378*, p. 48–62, 1998.

[AP98] R. Amadio, S. Prasad, Modelling IP mobility , *in: Proc. CONCUR98, Springer Lect. Notes in Comp. Sci. 1466*, p. 301–316, 1998.

[BS97] M. Boreale, D. Sangiorgi, Bisimulation in name-passing calculi without matching , *in: thirteen LICS Conf.*, IEEE Computer Society Press, 1998.

[CH98] I. Castellani, M. Hennessy, Testing theories for asynchronous languages , *in: Proceedings FST-TCS'98*, 1998. To appear

[S98a] S. Dal-Zilio, Objets concurrents dans un π-calcul applicatif , *in: accepté par les Journées Francophones des Langages Applicatifs*, 1998.

[S98b] S. Dal-Zilio, Quiet and Bouncing Objects: Two Migration Abstractions in a Simple Distributed Blue Calculus , *in: 1st International Workshop on Semantics of Objects as Processes*, BRICS Notes Series, 1998.

[HKMN98] H. Hüttel, J. Kleist, M. Merro, and U. Nestmann. Migration = cloning ; aliasing. To be presented at Sixth Workshop on Foundations of Object-Oriented Languages (FOOL 6), 1999.

[KlSa98] J. Kleist, D. Sangiorgi, Imperative Objects and Mobile Processes , *in: Proc. IFIP Working Conference on Programming Concepts and Methods (PROCOMET'98)*, North-Holland, 1998.

[MeSa98] M. Merro, D. Sangiorgi, On asynchrony in name-passing calculi , *in: 25th ICALP*, *LNCS*, *1443*, Springer Verlag, 1998.

[ME98] M. Merro, On the Expressiveness of Chi, Update and Fusion Calculi , Electronic Notes in Theoretical Computer Science Volume 16 , Issue 2

1998.

[STR98] D. Sangiorgi, Interpreting functions as pi-calculus processes: a tutorial , *rapport de recherche NoRR-3470*, INRIA-Sophia Antipolis, 1998.

[Bou98] G. Boudol, On the semantics of the call-by-name CPS-transform , available at the author's web page, 1998, submitted.

[1] PV98a J. Parrow and B. Victor. The fusion calculus: Expressiveness and symmetry in mobile processes. In *Proceedings of LICS '98*, 1998.

[2] PV98b J. Parrow and B. Victor. The tau-laws of fusion. In D. Sangiorgi and R. de Simone, eds, *Proceedings of CONCUR '98*, volume 1466 of *LNCS*, pages 99–114. Springer, Sept. 1998.

[3] Vic98a B. Victor. *The Fusion Calculus: Expressiveness and Symmetry in Mobile Processes.* PhD thesis, Department of Computer Systems, Uppsala University, June 1998. Available as report DoCS 98/98.

[4] VP98 B. Victor and J. Parrow. Concurrent constraints in the fusion calculus. In K. G. Larsen, S. Skyum and G. Winskel, eds, *Proceedings of ICALP '98*, volume 1443 of *LNCS*, pages 455–469. Springer, July 1998.

[BNFP98] Bettini, L., De Nicola, R., Ferrari, G., Pugliese, R. Interactive Mobile Agents in XKLAIM. *WETICE'98*, *Proceedings*, IEEE, 1998, to appear.

[B98] Bruni, R., A Logic for Modular Descriptions of Asynchronous and Synchronized Concurrent Systems (abstract) in Proceedings 2nd International Workshop on Rewriting Logic and its Applications, (Kirchner C., Kirchner H., Editors), ENTCS 15 (1998).

[BGM98] Bruni R., Gadducci F., and Montanari U. Normal Forms for Partitions and Relations in Proceedings 13th Workshop on Algebraic Development Techniques, LNCS, to appear.

[BM98a] Bruni, R. and Montanari, U. Zero-Safe Nets: The Individual Token Approach in Proceedings 12th Workshop on Algebraic Development Techniques, (Parisi-Presicce F., Editor), LNCS 1376, pp. 122–140 (1998).

[BMM98a] Bruni R., Meseguer J. and Montanari U. Process and Term Tile Logic Technical Report SRI-CSL-98-06, SRI International (July 1998). Also Technical Report TR-98-09, Department of Computer Science, University of Pisa (1998).

[BMM98b] Bruni R., Meseguer J. and Montanari U. Internal Strategies in a Rewriting Implementation of Tile Systems in Proceedings 2nd International Workshop on Rewriting Logic and its Applications, (Kirchner C., Kirchner H., Editors), ENTCS 15 (1998).

[BMM98c] Bruni R., Meseguer J. and Montanari U. Implementing Tile Systems: Some Examples from Process Calculi in Proceedings 6th Italian Conference on Theoretical Computer Science, World Scientific, 1998.

[BMMS98] Bruni R., Meseguer J., Montanari U. and Sassone V., A Comparison of Petri Net Semantics under the Collective Token Philosophy in Proceedings 4th Asian Computing Science Conference, (Hsiang J., Ohori A., Editors), LNCS, to appear.

[BM98b] Bruni, R., Montanari, U., Zero-Safe Nets: Comparing the Collective and Individual Token Information and Computation, to appear.

[NFP98] De Nicola, R., Ferrari, G., Pugliese, R., Klaim: a Kernel Language for Agents Interaction and Mobility. *IEEE Transactions on Software Engineering*, Vol.24(5):315-330, May, 1998.

[FGMPR98] Ferrari,G., Gnesi, S., Montanari, U., Pistore, M. and Ristori, G. Verifying Mobile Processes in the HAL Environment, In Proc. CAV'98, LNCS 1427. Springer Verlag, 1998.

[FeM98] Ferrari, G. and Montanari, U., Tile Formats for Located and Mobile Systems, To appear in Information and Computation, 1998.

[HLMP98] Honsell, F., Lenisa, M., Montanari, U., Pistore, M. Final Semantics for the pi-Calculus, In Proc. PROCOMET'98, 1998.

[HIM98] Hirsch, D., Inverardi, P., and Montanari, U., Modeling Software Architectures and Styles with Graph Grammars and Constraint Solving, to appear in Proceedings WICSA 98, 1998.

[MM98] Montanari, U. and Meseguer, J., Mapping Tile Logic into Rewriting Logic, in: Francesco Parisi-Presicce, Ed., Recent Trends in Algebraic Development Techniques, Spinger LNCS 1376, 1998, pp. 62-91. i

[MP98] Montanari , U. and Pistore, M. An Introduction to History Dependent Automata In Proc. HOOTS'98, ENTCS 10. Elsevier Science, 1998.

[MT98] Montanari , U. and Talcott, C., Can Actors and pi-Agents Live Together?, in: Andrew Gordon, Andrew Pitts and Carolyn Talcott, Eds, Second Workshop on Higher-Order Operational Techniques in Semantics (HOOTS II), ENTCS, Vol. 10, 1998.

[HR98] Matthew Hennessy and James Riely. Type-safe execution of mobile agents in anonymous networks (extended abstract). In *Proceedings of 4th Workshop on Mobile Object Systems*, Brussels, Belgium, July 1998. Full version available as Computer Science Technical Report 03/98. Available from http://www.cogs.susx.ac.uk/.

[Rat98] J. Rathke. Resource based models for asynchrony. In *Proc. FOSSACS'98, International Conference on Foundations of Software Science and Computation Structures, Lisbon*. Springer-Verlag, 1998.

[RH98] James Riely and Matthew Hennessy. Resource access control in systems of mobile agents (extended abstract). In *Proceedings of 3rd International Workshop on High-Level Concurrent Languages*, Nice, France, September 1998. Full version available as Computer Science Technical Report 02/98. Available from http://www.cogs.susx.ac.uk/.

[Yos98] Nobuko Yoshida. Minimality and separation results on asynchronous mobile processes: representability theorem by concurrent combinators (extended abstract). In *Proceedings of CONCUR98*, Nice, France, September 1998. Full version available as Computer Science Technical Report 05/98. Available from http://www.cogs.susx.ac.uk/.

[QW98] P. Quaglia and D. Walker, *On encoding $p\pi$ in $m\pi$*, in Proceedings of Foundations of Software Technology and Theoretical Computer Science, December 1998, Springer-Verlag Lecture Notes in Computer Science, to appear.

[LW98] X. Liu and D. Walker, *Partial confluence of processes and systems of objects*, Theoretical Computer Science 206 (1998) 127–162.

[PW98] A. Philippou and D. Walker, *On transformations of concurrent object programs*, Theoretical Computer Science 195 (1998) 259–289.