CONFER-2

CONcurrency and Functions:

Evaluation and Reduction

Working Group

Project Number: 21836

Periodic Progress Report
December, 1999

Contents

Chapter 1

Overview

This report contains the Third 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 5 and 6, 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, Pierre-Louis Curien, Jean-Francois Monin, Bent Thomsen, 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 is a framework to keep together the community of the previous Basic Research Action CONFER, and to provide two annual workshops. The objective of CONFER-2 is 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 organized around four main areas :

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

In the area of Foundational Models and Abstract Machines, the research activity has progressed in three main areas.

Firstly a book by Asperti and Guerrini has been published at Cambride University Press. It summarizes the research on Optimal Reductions in the lambda calculus. This area has been very active in the last decade, especially inside our Working Group (see previous Progress Reports of CONFER and CONFER-2). In the same spirit, the study of origin tracking in the lambda calculus has been pursued by Klop at CWI.

Secondly, a result on the finiteness of the normalization cone in the lambda-sigma calculus has been published by Melliès on leave from University of Edinburgh to ENS. In the same spirit, Bruni, Gadduci, and Montanari have studied the connection between three reduction systems: rewriting logic (Meseguer), action calculi (Milner), and tile logic (Montanari). Both approaches are based on category theory.

Thirdly, the work on Action Calculi is still strongly developed at Cambridge. Action Calculi form an abstract model for calculi with bound variables and guards (preventing from computing below them). Leifer investigates general methods for deriving labeled transition systems from reduction rules in a way initiated by Milner and Sewell. Indeed labeled transition systems provide an intensional semantics describing the interactions of processes on contexts, and it would be useful to automatically derive bisimulations as congruences from the extensional semantics given in terms of reduction rules and observational equivalences. A correspondence between action calculus and the synchronous pi-calculus has been also studied by Gardner.

In the area of Calculi, progress has been made mainly in four directions.

Security, which is a major issue for distributed languages with mobile primitives, has been studied in various forms at different sites inside CONFER-2. Secure encapsulation is proposed by Sewell and Vitek as a way of assembling concurrent, software systems with untrusted or partially untrusted parts. Principals have been added to the join calculus by Abadi, Fournet and Gonthier. It allows formal statements about authentication and secrecy, which may be proved with standard proof techniques of process calculi. Another proof technique has been developed by Amadio and Prasad for the verification of secrecy and authentication with symmetric shared keys. Hennessy and Yoshida consider types as a way of expressing locality in the pi-calculus. Finally, a dependency calculus is studied by Conchon and Pottier for descriptions of static interference properties using data flow techniques.

The theory of asynchronous bisimulations equivalences has been intensively considered. A tutorial on asynchronous calculi has been published by Sangiorgi. The work on the connection between the synchronous and asynchronous pi-calculi is pursued by Quaglia and Walker through typed barbed equivalences. Merro gives a labeled characterization of barbed-congruences in the asynchronous pi-calculus without matching. Castellani and Hennessy have investigated an asynchronous version of CCS with both may and must testing preorders, as a basis for future work on the theory of testing for the asynchronous pi-calculus.

Various localized asynchronous calculi have also been introduced or examined during last year. The receptive asynchronous distributed pi-calculus of Amadio, Boudol and Loussaine is based on a typing discipline insuring uniform receptiveness. Its expressivity is shown through the coding of standard examples. Safe Ambients is a new theory of Levi and Sangiorgi which provides a nice equational theory for Ambients. The feasibility of a distributed implementation for Ambients has been explored by Fournet, Lévy and Schmitt.

The pi-calculus may also have a symmetric expression through the Parrow and Victor's fusion calculus. Its expressivity is studied by Laneve and Victor. Gardner also shows the connection with action calculi.

The theory of concurrent objects is another important direction of the Calculi area. The aliasing model of Obliq, Ojeblik, demonstrates that several different interpretations exist for the initial Cardelli's Obliq calculus. The objective join calculus is still under development by Fournet, Laneve, Maranget and Rémy. The extension of the pi-calculus with concurrent objects has been published in a journal by Liu and Walker. Finally, Dal Zilio achieved his PhD on the theory of types for objects by using the Boudol's Blue Calculus.

Several more ponctual results may also be mentioned under the Calculi area: Sangiorgi studies CIA (Concurrent Idealised Algol), History-Dependent Automata is a framework for model checking for the finite pi-calculus.

In the area of Logics for Concurrency and the λ-calculus, the work has been concentrated this year mainly at ENS and Edinburgh.

Concurrent Games semantics is a new setting introduced by Abramsky and Melliès. Its completeness with respect to the multiplicative additive linear logic has been proved. Danos et al. have connected concurrent games theory with the Ehrhard hypercoherent space model. The duality of call-by-name and call-by-value on one hand, and of input and output on the other hand, is explored by Curien and Herbelin via the Curry-Howard correspondence with the fragments LKT/LKQ of the linear logic. The fragment LKTQ is also considered by Danos. This work surprisingly provides a symmetric and dual lambda-calculus.

Finally, in the area of Programming Languages, progress has been produced in the following directions.

Nomadic Pict is a distributed version of Pierce/Turner's Pict language. Sewell, Unyapoth and Wojciechowski continues their work on the design and semantics. The implementation is now completed. Jocaml (Caml augmented with the join-calculus primitives) has been equipped of a distributed garbage collector (as published in previous Progress Report). At Pisa, Klaim (Kernel Language for Agents Interaction and Mobility) is a prototype language for coordinating processes which run on a net.

ICL explores the downstream use of agent technology with Fujitsu Laboratories, and ICL Businesses. The usage of Facile has been pushed to the Java community by the new availability of MLj (Standard ML compiled to Java) of Microsoft Research. As Facile is written in SML, one easily get the translation of Facile programs to JVM code, and to strengthen the compatibility with other Java programs.

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

The working group has produced about 65 reports and several of these have been published at conferences, international workshops and journals. Many of these works were produced by researchers at different sites within the Working Group and can be stated as state-of-the-art research. They are mainly 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 third year of the CONFER-2 working group:

Two workshops have been held. The first took place in April 1999 in Pisa with 20 presentations and 30 participants, and was organized by Gianluigi Ferrari and Ugo Montanari. The second workshop took place in November 1999 at University of Paris 7 (in the new buildings of rue du Chevaleret) near ENS in Paris with 14 presentations and 30 participants. It was organized by Pierre-Louis Curien.

As specified in §2.6 of the Working Group Programme, the vast majority of the activity of CONFER-2 is devoted to the organization 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.

Finally, we obtained in 1999 the permission for a one-year prolongation of Confer-2 without total budget change.

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 in the analysis of bisimulation in the framework of join-calculus, in the expressivity of the fusion-calculus and the study of a novel class-based concurrent calculus – the *objective join-calculus.*

Analysis of Bisimulation We have finished the full paper about the theory of bisimulation in the join-calculus [FL99] (the extended abstract was published in the Procomet conference in 1998). To this purpose, we have proved that several formulations of bisimulation actually yield the same equivalence and that bisimulation and barbed congruence coincide in the presence of name-testing.

Expressivity of the Fusion Calculus In [LV99] we present a calculus of mobile processes without prefix and summation, and using two different encodings we show that it can express both action prefix and guarded summation. One encoding gives a strong correspondence but uses a match operator; the other yields a slightly weaker correspondence but uses no additional operators.

The composition of concurrent classes In [FLMR99] 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.

Programming Languages We have published the book "Optimal Implementation of Functional Programming Languages" [AG98], where most of our previous results about optimal implementations have been collected.

3.2.2 Workshops, Travels and Visits

No travel and visit.

3.2.3 Publications

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

[FL99] C. Fournet, C. Laneve: *Bisimulations in the Join-Calculus.* Submitted to a Journal. June 1999.

[FLMR99] C. Fournet, C. Laneve, L. Maranget, D. Rèmy: *Assembling Objects with Internal Concurrency.* Draft, October 1999.

[LV99] C. Laneve, B. Victor: *Solos in Concert.* ICALP'99, LNCS 1644.

3.3 University of Cambridge

3.3.1 University of Cambridge

3.3.1.1 Research directions

Categorical models of process calculi Cattani received his PhD from Aarhus University (DK) in May [CaPhD99]. Progressing from his thesis work, together with Fiore from Sussex and Winskel from Aarhus, he has recently introduced a systematic treatment of weak bisimulation and observational congruence on presheaf models [FCW99]. This work was also presented at the CONFER-2 Workshop in Pisa. The paper [PCW99], submitted last year and mentioned in the Periodic Progress Report of 1998, has now been accepted for publication.

Together with Sewell he is presently developing models for mobile-process calculi, *e.g.*, the π-calculus, which extend traditional ones such as (asynchronous) transition systems and synchronization trees, by having indexed sets of states to account for the dynamic generation and communication of channel names.

Bisimulation and Action Calculi Leifer is investigating general methods for deriving labeled transition systems from reduction rules in such a way that bisimulation over the derived labeled transitions is a congruence. The setting for this work is Milner's action calculi. Over this past year, Leifer has strengthened his results to operate on action calculi rich enough to include the synchronous pi-calculus; in this case the derived bisimulation corresponds with early bisimulation, a well-known equivalence for pi. His work relies on properties of context factorization in action calculi. He is constructing a category with terms as objects and contexts as morphisms to investigate the universal properties of the factorizations.

The work of Sewell on the problem in settings without name binding has been accepted for publication [Sew99a].

Gardner has recently shown the precise link between the synchronous π-calculus and the action calculus for the π-calculus, strengthening previous results.

Gardner is supervising the design and construction of a specification tool for action graphs, together with Michael Norrish, which allows the user to naturally move between the syntactic and graphical presentations. They have solved a difficult matching problem to identify instances of redexes in a graph, and with Andrei Serjantov have extended the tool to the reflexive case.

Fusions, Naming and Graphs Wischik and Gardner have developed a π-calculus with fusions, called the π_F-calculus, which arose from their work on process frameworks described below. It turns out that the π_F-calculus is related to the fusion calculus of Parrow and Victor, although the naming primitives are different, and to work by Honda and by Merro on equators in the π-calculus. They have identified bisimulation congruences for the π_F-calculus. There are simple embeddings of the the π-calculus, the π_I-calculus and the fusion calculus in the π_F-calculus. They are currently showing that bisimulation congruences for these calculi are related to bisimulation congruences for their embeddings in the π_F-calculus.

Wischik and Gardner are further exploring the role of naming constructs in graphical frameworks. In particular, they have introduced a process framework based on the same naming primitives as the π_F-calculus. This framework conservatively extends the reflexive action calculus framework. It seems to give a more direct presentation of many examples, and has a simple general notion of barbed congruence. Their aim is to show that the barbed congruence for the framework corresponds to natural notions of congruence in specific examples.

Nomadic Pict Sewell, Unyapoth and Wojciechowski have continued work on the *Nomadic Pict* distributed programming language, studying the design, semantic definition and implementation of location-independent communication primitives for mobile agents. The language implementation (by Wojciechowski) has now been completed; it is being used to experiment with infrastructure algorithms. The language and example infrastructures have been described in [SWP99b] (with Pierce) and in [WS99]. On the semantic side, Unyapoth has developed the theory of the *Nomadic π-calculus* to support infrastructure correctness proofs. The calculus has been equipped with operational semantics (both labeled transition and reduction), a type system and notions of bisimulation; work on partial confluence proof techniques is ongoing.

Secure Encapsulation Sewell and Vitek (of Purdue University) are considering the problem of assembling concurrent software systems from untrusted or partially trusted off-the-shelf components, using *wrapper programs* to encapsulate components and enforce security policies. In [SV99a, SV99b] they introduced the *box-π* process calculus with constrained interaction to express wrappers and discussed the rigorous formulation of their security properties. In [SV99c] they address the verification of wrapper information flow properties, presenting a novel *causal type system* that statically captures the allowed flows between wrapped possibly-badly-typed components; it is used to prove that a unidirectional-flow wrapper enforces a causal flow property.

Polynomial Pi Sawle and Milner have developed the *Polynomial Pi Calculus*, an extension of the π-calculus with multiple synchronization. An animation tool has been implemented; work on applying it to verifying properties of security protocols is underway.

3.3.1.2 Persons and Exchanges

Leifer visited INRIA Rocquencourt for a week in April and a month in May/June 1999. Gardner visited Amsterdam for two weeks, INRIA Rocquencourt for a week and Sussex. Cattani, Gardner, Leifer, Sewell and Wischik attended the CONFER workshop in Pisa.

3.3.2 Publications

[CaPhD99] G. L. Cattani. Presheaf Models for Concurrency. PhD Thesis, University of Aarhus, 1999.

[FCW99] M. Fiore, G. L. Cattani and G. Winskel. Weak Bisimulation and Open Maps. In *Proceedings of the 14th Symposium on Logic in Computer Science,* *LICS '99*, pages 67–76, IEEE Press, 1999.

[PCW99] A. J. Power, G. L. Cattani and G. Winskel. A Representation Result for Free Cocompletions. To appear in *Journal of Pure and Applied Algebra*, 1999.

[Ga99a] Philippa Gardner. Closed Action Calculi. Journal of Theoretical Computer Science, 1999. To appear.

[GW99] Philippa Gardner and Lucian Wischik. Symmetric Action Calculi. Extended abstract, Express 99.

[Ga99b] Philippa Gardner. Interaction Nets, Sharing graphs and Action Calculi. Submission in association with the Mathfit Workshop on Process Algebra, January 99.

[SV99a] Peter Sewell and Jan Vitek. Secure composition of insecure components. Technical Report 463, Computer Laboratory, University of Cambridge, April 1999.

[SV99b] Peter Sewell and Jan Vitek. Secure composition of insecure components. In *Proceedings of the 12th IEEE Computer Security Foundations Workshop. Mordano, Italy*, pages 136–150. IEEE Computer Society, June 1999.

[SV99c] Peter Sewell and Jan Vitek. Secure Composition of Untrusted Code: Wrappers and Causality Types. Submitted for publication.

[Sew99a] Peter Sewell. From rewrite rules to bisimulation congruences. *Theoretical Computer Science*, 1999. Invited submission, to appear in a special issue for CONCUR 98.

[SWP99b] Peter Sewell, Paweł T. Wojciechowski, and Benjamin C. Pierce. Location-independent communication for mobile agents: a two-level architecture. In *Internet Programming Languages*, LNCS 1686. Springer-Verlag, 1999.

[WS99] Paweł T. Wojciechowski and Peter Sewell. Nomadic Pict: Language and infrastructure design for mobile agents. In *Proceedings of ASA/MA 99: Agent Systems and Applications/Mobile Agents*, IEEE, October 1999.

[Sew99pi] Peter Sewell. A brief introduction to applied π, January 1999. Lecture notes for the Mathfit Instructional Meeting on Recent Advances in Semantics and Types for Concurrency: Theory and Practice, July 1998.

3.4 CWI

3.4.1 Research Directions

J.W. Klop has completed work on origin tracking in lambda calculus and term rewriting. This work is reported in the paper "Descendants and Origins in term Rewriting", accepted for publication in Information and Computation, also to be found at http://www.cs.vu.nl/ rdv/dotr.ps.gz (Joint work with I. Bethke and R.C. de Vrijer)

J.W. Klop (together with I. Bethke and R.C. de Vrijer) has completed a study on extensions of partial combinatory algebras, published in Math. Struct. in Comp.Science (1999) vol.9, pp.483-505. (Also at ftp://ftp.cs.vu.nl/pub/papers/theory/IR-448.ps.Z)

J.W. Klop, V. van Oostrom and R.C. de Vrijer have completed a study of infinite reduction diagrams, yielding an alternative proof of the theorem of De Bruijn and van Oostrom concerning confluence by decreasing diagrams: 'A geometric proof of confluence by decreasing diagrams.' To be published in special issue of JLC. (Also at ftp://ftp.cs.vu.nl/pub/papers/theory/IR-448.ps.Z)

3.4.2 Workshops, Travels and Visits

J.W. Klop attended the CONFER-2 Workshop in Pisa, March, 1999.
J.W. Klop, F. van Raamsdonk, V. van Oostrom attended the Workshop Terms and Types, April 10-13, Heriot-Watt, Edinburgh.
J.W. Klop gave an invited tutorial 'Ten topics in Term Rewriting' at the Logic Colloquium '99, Utrecht, The Netherlands, August 1-6, 1999.

3.5 University of Edinburgh

3.6 ENS

3.6.1 Research Directions

Work at the ENS site has gone into four directions.

- Probabilistic games: Vincent Danos and Russel Harmer, a former student of Pasquale Malacaria, are working on an extension of games semantics to model functional or imperative computations extended with a random instruction. So far, they have proved that in their setting of probabilistic strategies, the coin strategy that returns true or false with equal chances, is universal, that is any probabilistic strategy can be factorized as a deterministic one composed with the coin.

- Games semantics and denotational semantics: In collaboration with Samson Abramsky, Paul-André Melliès has introduced in [AbMe99] a concurrent game model of linear logic and proved a full completeness result for multiplicative additive linear logic. Then, by polarizing the concurrent game model, Paul-André Melliès has unveiled the dynamical content of the hypercoherence space model introduced by Thomas Ehrhard in 1993. The idea is to construct a *full* functor U from the game model $\mathcal{G}$ to the hypercoherence model $\mathcal{H}$, and to show that U respects the constructions of linear logic; the "fullness" of U asserts that every *static* function (or clique) of $\mathcal{H}$ may be implemented as a *dynamical* and concurrent strategy [Me99a].

  Another important question, already attacked by Thomas Ehrhard in his work on the extensional collapse of sequential algorithms, and by Patrick Baillot, Vincent Danos, Thomas Ehrhard and Laurent Regnier in their work on timeless games [Da99], is the connection between a static model like, say hypercoherence spaces, and *sequential* game models. After [BDER98], one is tempted to see the problem as purely game-theoretical, of understanding the connection between the sequential and the concurrent version of game semantics. This approach is currently under investigation.

- Call-by-name, call-by-value, continuations and sequent calculus: About four years ago, Vincent Danos, Jean-Baptiste Joinet and Harold Schellinx, have packed a comprehensive study of sequent calculi for classical logic, most notably LKT, LKQ and LKTQ, that one could embed in linear logic. In [Da99] a sound and intuitive continuation calculus is attached to LKQ, yielding a 'Curry-Howard' configuration. Plugging in a standard embedding of natural deduction in sequent calculus, one retrieves a Plotkin-like continuation passing style compilation of call-by-value lambda-calculus with control. In collaboration with a student, Charles Harris, Vincent Danos is now trying to set up a mixed call-by-value/call-by-name CPS compilation starting from LKTQ. On the other hand, Pierre-Louis Curien and Hugo Herbelin have just started to work independently along similar lines: they have proposed a syntax for call-by-value calculi that is directly inspired by the sequent calculus LKQ. This syntax offers an attractive logical explanation of the atomic steps of the call-by-value abstract machines, and enhances symmetries and dualities (call-by-name / call-by-value, input / output)

- Axiomatic rewriting: Paul-André Melliès has entirely rewritten the second part of his four articles on axiomatic rewriting theory. The paper [Me99b] contains a short and elegant proof of needed normalization for lambda-sigma-terms. The argument is based on the following observation: that a *standard* rewriting path $M \to N$ in the $\lambda\sigma$-calculus projects as a *standard* rewriting path $\sigma(M) \to N$ when N is a λ-term.

3.6.2 Publications

[Da99] V. Danos.(1999) Sequent Calculus and Continuation Passing Style Compilation. To appear in the Proceedings of the 11th Congress of Logic, Methodology and Philosophy of Science, held in Cracow, Kluwer.

[AbMe99] S. Abramsky, P.-A. Melliès.(1999) Concurrent games and full completeness. Proceedings of LiCS'99, Trento.

[Me99a] P.-A. Melliès.(1999) Hypercoherence spaces as concurrent games, in preparation.

[Me99b] P.-A. Melliès.(1999) Axiomatic rewriting theory II: The lambda-sigma calculus enjoys finite normalization cones. Submitted.

[BDER98] P. Baillot, V. Danos, T. Ehrhard, L. Regnier.(1998) Timeless Games. In proceedings of CSL'97, held in Aarhus. Springer LNCS 1414, Mogens Nielsen and Wolfgang Thomas eds., pp. 56–77.

3.7 France Telecom – CNET

3.7.1 Research Directions

In 1999, P.!Chavin worked on an enrichment of CORBA-IDL with operational concepts. C. Bergerot, P. Brisset and J.F. Monin begun to investigate problems raised by dynamic configurations of interacting software processes with complex communication and control patterns. As a typical case study, they consider a planet-wide chat services application. P. Brisset wrote a first version using CML constructs implemented in Objective Caml. The next step, undertaken by C. Bergerot and J.F. Monin, is to take profit of a mobile process approach in order to dynamically reconfigure the routing topology.

3.7.2 Conferences

J.F. Monin and F. presented a paper at FM'99, about a completely formalized proof of a crucial component for the ATM CAC called ABR. They also presented a tutorial on formal methods in telecommunications at the same conference.

3.7.3 Publications

[MKl99] Jean-François Monin and Francis Klay, "Correctness Proof of the Standardized Algorithm for ABR Conformance", FM'99 – Formal Methods, LNCS 1708, Springer Verlag", ed. Jeannette Wing and Jim Woodcock and Jim Davies, pp 662-681, 1999

3.8 ICL

3.8.1 Research Directions

The ICL framework in Agents and Mobility described in PPR1 and PPR2 is running well with a number of different activities, still encouraging 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.

The ICL participation in CONFER-2 is one of the framework activities.

The virtual laboratory in collaboration with Fujitsu Laboratories, run as part of the framework, is very much a working reality now. Ideas are exchanged on the mailing list together with updates on relevant issues. Other parts of the Fujitsu Family have now joined the virtual laboratory.

Recently the focus has been on the Kafka-Pathwalker technology from Fujitsu Laboratories. Kafka-Pathwalker is a distributed multi-process and agent/action Java based programming library based on a distributed message queuing system. Kafka-Pathwalker draws upon ideas from higher order process calculi and concurrent functional programming from languages such as April and Facile.

A major project finished this year, demonstrating how agent technology could be used to enhance web-computing products from Fujitsu's TeamWARE group by integration of agent technology into their core architecture, called Phoenix. The resulting system is called Agent Enhanced Phoenix. This technology is now scheduled for productization.

At ICL we have used this technology for building demonstrators of how agent technology can be used in real world scenarios. A very successful demonstrator built around the scenario of "Intelligent Knowledge Management for Engineering in ICL" has helped ICL Businesses understand the potential of the technology. There are now plans for making the technology part of the ICL solution kit and applying it in major projects.

Close interaction with the developers of the Kafka-Pathwalker technology in Fujitsu Laboratories and the developers of Phoenix in TeamWARE Group ensures transfer of ideas, also from the CONFER-2 context, and feedback from prototype development will hopefully make systems more flexible and better targeted towards future applications.

Last year we reported on work on transferring ideas from the Facile system to the Java world via the MLj (ML to Java Virtual Machine code) compiler (which has now moved home from Persimmon to Microsoft Research in Cambridge). We have succeeded in building the Facile concurrency primitives in MLj. However, work on the Facile distribution primitives has not progressed as quickly as anticipated due to other commitments. In the coming year we hope to use the Kafka-Pathwalker library as a transport layer in developing the Facile distribution primitives in MLj. Kafka-Pathwalker already implements many of the necessary facilities for sending and receiving arbitrary values including higher order objects.

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

3.8.2 Exchange of Personnel

Lone Leth Thomsen and Bent Thomsen participated in the 5th CONFER-2 workshop in Pisa (University of Pisa) 29.03.99 - 31.03.99.

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:

[LTL99] J.-J. Lévy, B. Thomsen, L. Leth: "Second Year Report for Esprit Working Group 21836 CONFER-2 CONcurrency and Funstions: Evaluation and Reduction", EATCS Bulletin Number 67 February 1998 pp. 35-51.

[DPLB99] P. Degano, C. Priami, L. Leth, B. Thomsen: "Causality for Debugging Mobile Agents", accepted for publication in ACTA INFORMATICA, 1999.

3.9 INRIA-Rocquencourt

3.9.1 Research directions

INRIA Rocquencourt has worked in the following directions.

- Distributed Join-Calculus.

  The implementation of the Join calculus is being pursued, under Jocaml. (see `join.inria.fr`)

- Concurrency and Security

  In collaboration with M. Abadi (Compaq SRC – Lucent Technology), C. Fournet (on leave for Microsoft Research, Cambridge) and G. Gonthier give a semantics for abstractions of secure cryptographic protocols, and prove their correctness. Usually, security is described by informal and global statements, and needs a formal definition consistent with the semantics of programming languages. By providing a set of high-level primitives inside a programming language, one allows automatic properties such as secrecy and authentication. The proof of security is reduced to the proof of a full abstraction theorem between the high-level programming language with security primitives and the implementation language. Moreover, this proof demonstrates how a secure implementation of such a programming language can be built on top of the join calculus.

  As a first example, the case of the private communication channels of the join calculus is studied. The security of the implementation may be transparently achieved by a set of interface modules between each principal and the network, instead of providing a special compiled version of each principal.

  For authentication, a new notion of principal is added to the high-level language. It gives two new interesting features: firstly, the expressivity of the high-level language is augmented and for instance may contain statements about access control policies (with mere channels, one is usually restricted to capabilities); secondly it allows again efficient implementations. This latter point is rather surprising and comes from the small number of cryptographic operations which now are simple multiplexers for all operations between principals.

  The two versions of this work are being submitted (long version) or presented at POPL'2000 (conference version).

- Distributed Ambients

  C. Fournet, J.-J. Lévy, and A. Schmitt studied the distributed implementations for the Cardelli-Gordon's Ambient calculus. This framework is a very simple and esthetic model of concurrent processes with mobility. However it has no distributed implementation until now. Using Jocaml, an asynchronous, distributed implementation may be designed. It relies on small protocols with several steps for implementing each of the atomic step of the Ambient calculus. A translation of

Ambients into the join calculus provides a Jocaml implementation for Ambients. Although this implementation is a simple prototype, its efficiency is rather good, since it is bases on pure asynchrony. The corresponding implementation has been presented at MOS'99 by C. Fournet and A. Schmitt, and a paper describing its theory is being submitted.

- Objects

  This year, Fournet, Maranget, Laneve (Bologna) and Rémy pursued their work on distributed objects.

- Security and Flow analysis

  S. Conchon and F. Pottier consider data dependency and flow analysis in the lambda calculus to express interference between lambda terms. This study can be extended to concurrent processes, such as the Join calculus.

3.9.2 Persons and exchanges

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

From Cambridge, we had regular visits by Cédric Fournet, a one month visit by James Leifer, and from Bologna several visits by Cosimo Laneve.

3.9.3 Publications

[AFG99a] Martín Abadi, Cédric Fournet, and Georges Gonthier. Secure implementation of channel abstractions. à paraître, 1999.

[AFG99b] Martín Abadi, Cédric Fournet, and Georges Gonthier. A top-down look at a secure message. In *Proc. of the 19th Foundations of Software Technology and Theoretical Computer Science, Chennai*, December 1999.

[AFG00] Martín Abadi, Cédric Fournet, and Georges Gonthier. Authentication primitives and their compilation. In *27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. ACM Press, January 2000.

[FLS99] Cédric Fournet, Jean-Jacques Lévy, and Alan Schmitt. A distributed implementation of ambients. 1999. à paraître.

[FS99] Cédric Fournet and Alan Schmitt. An implementation of Ambients in JoCAML. In *5th Mobile Object System Workshop: Programming Languages for Wide Area Networks*, June 1999.

[lF99] Fabrice le Fessant. Jocaml. available at `http://join.inria.fr`, 1999.

[LM99] Jean-Jacques Lévy and Luc Maranget. Explicit substitutions and programming languages. In *Proc. of the 19th Foundations of Software Technology and Theoretical Computer Science, Chennai*, December 1999.

3.10 INRIA-Sophia

3.10.1 Participation to CONFER Workshops and expenses

M. Merro, R. Amadio, G. Boudol, C. Lhoussaine, S. Dal Zilio attended the CONFER II workshop in Pisa.

Some CONFER money has been used for visits of I. Castellani to Hennessy's group in Sussex (1 week), and D. Sangiorgi to Walker's group in Oxford (2 weeks). D. Sangiorgi has also visited Pisa, to continue a collaboration with F. Levi.

3.10.2 Joint works

Boudol has collaborated with Curien and Lavatelli (ENS Paris), producing the joint work [BCL99].

F. Levi (Pisa) has spent about 10 months as a postdoc at Sophia. A collaboration with D. Sangiorgi has produced [LS00].

3.10.3 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 distribution and types.

Semantics and expressiveness Merro [Mer99], developing work carried out last year in Sophia and Sussex, has given a labeled characterization of barbed congruence in asynchronous π-calculus, which, unlike previous characterizations, does not use the matching construct. In absence of matching the observer cannot directly distinguish two names. In asynchronous π-calculus the fact that two names are indistinguishable can be modeled by means of Honda and Yoshida's notion of *equator*. The labeled characterization is based on such a notion. As an application of the theory a fully-abstract encoding w.r.t. barbed congruence of *external mobility* (communication of free names) in terms of *internal mobility* (communication of private names) is provided.

Merro, together with Hüttel, Kleist, and Nestman (Aalborg, Denmark) [NHKM99] has also continued his work on distributed objects based on Cardelli's Obliq, a lexically scoped, distributed, object-oriented programming language. In Obliq, object migration was suggested as the creation of a copy of an object's state at the target site, followed by turning the object itself into an alias, also called *surrogate*, for the remote copy. The creation of object surrogates as an abstraction of the above-mentioned style of migration is considered. *Ojeblik*, a distribution-free subset of Obliq, is introduced and three different configuration-style semantics, which only differ in the respective *aliasing model*, are provided. It is shown that two of the semantics, one of which matches Obliq's implementation, render migration unsafe, while a new proposal for a third semantics is provably safe. This work suggests a straightforward repair of Obliq's aliasing model such that it allows programs to safely migrate objects.

Sangiorgi, in a collaboration with Röckl (Technische Universität München) [RS99], has studied the use of the π-calculus for semantical descriptions of languages such as *Concurrent Idealised Algol* (CIA), combining imperative, functional and concurrent features. First an operational semantics for CIA has been produced, given by SOS rules and a contextual form of behavioral equivalence; then a π-calculus semantics. As behavioral equivalence on π-calculus processes the standard (weak early) bisimilarity has been chosen. The two semantics have been compared, demonstrating that there is a close operational correspondence between them and that the π-calculus semantics is sound. This allows for applying the π-calculus theory in proving behavioral properties of CIA phrases. For this, laws and examples which have served as benchmarks to various semantics have been used, as well as a more complex example involving procedures of higher order. By means of counterexamples it is shown that the π-calculus semantics is not complete, and discussed how these counterexamples can be dealt with by adopting refined forms of π-calculus types and, correspondingly, typed versions of bisimilarity.

Sangiorgi has written a tutorial paper [San99a] on the comparison between the *first-order* and the *higher-order* paradigms for the representation of mobility in process calculi. The prototypical calculus in the first-order paradigm is the π-calculus. In this work, an asynchronous π-calculus (Lπ) that may be regarded as the basis of some experimental programming languages (or proposal of programming languages) like Pict, Join, Blue has been taken. The calculus has then been extended so to allow the communication of higher-order values, that is values that may contain processes, and it has been shown that the extension does not add expressiveness: the resulting higher-order calculus can be compiled down into the original calculus Lπ. Although the paper is mostly a tutorial, it contains some original contributions. The main one is the full abstraction proof, which, with respect to previous proofs, is simpler and does not rely on certain non-finitary features of the languages such as infinite summation. Another contribution is the study of optimizations of the compilation, with which: recursive types can be handled; full abstraction can be proved also for *strong* behavioral equivalences.

The Blue calculus is a direct extension of both the lambda and the pi calculi. Dal Zilio [DZ99a] has defined an equivalence for this calculus based on barbed congruence, and proved the validity of the replication laws. For example, he proves that a replicated resource, shared by many processes, can be safely copied and distributed.

Papers written by Boudol in collaboration with Curien and Lavatelli (ENS Paris) [BCL99], and Sangiorgi [San99b] in previous years have been revised and will appear in a special issue of Mathematical Structures in Computer Science in Honour of Roger Hindley.

Distribution and types R. Amadio, G. Boudol, C. Lhoussaine [ABL99] have studied an asynchronous distributed π-calculus, with constructs for localities and migration. They have shown that a simple static analysis ensures the receptiveness of channel names, which, together with a simple type system, guarantees a local deadlock-freedom property, that they call message deliverability. This property states that any migrating message will find an appropriate receiver at its destination locality. The authors argue that this distributed, receptive calculus is still expressive enough, by giving a series of examples illustrating the "receptive style" of programming we have. Finally the authors show that the calculus contains the π_1-calculus, up to weak asynchronous bisimulation.

Gerard Boudol and Silvano Dal-Zilio [BDZ99] have provided a translation of Fisher-Honsell-Mitchell's delegation-based object calculus with subtyping into a lambda-calculus with extensible records. The target type system is an extension of the system F^ω of dependent types with recursion, extensible records and a form of bounded universal quantification. They have shown that the translation is computationally adequate, that the typing rules of Fisher-Honsell-Mitchell's calculus can be derived in a rather simple and natural way, and that the new system enjoys the standard subject reduction property.

During this year, Silvano Dal Zilio has completed his PhD thesis [DZ99b]. The thesis focuses on the blue calculus, is a variant of the polyadic pi-calculus that directly embeds the notion of function. In the thesis, Dal Zilio considers a version of the blue calculus extended with records, and studies whether it provides a good basis for a typed concurrent programming language with imperative, higher-order, and object features. He notably studies the modeling of the functional and object oriented programming idioms, and the addition of polymorphic typing and inheritance. The thesis is divided into four parts. The first part consists of a detailed analysis of the blue calculus and its expressiveness. In the second part, which includes [DZ99a] discussed above, a behavioral equivalence based on the classical notion of barbed congruence, and a labeled bisimulation that is finer than this congruence are defined. Dal Zilio then uses up-to proof techniques to prove the validity of several algebraic laws like, for example, an analogous of Milner's replication theorem for the pi-calculus.

In the third part, Dal Zilio studies type systems for the blue calculus. Starting from a simple implicit type system that encompasses both Curry's type system for the lambda-calculus, and Milner's sorting for the pi-calculus, he successively proposes three extensions of increasing complexity. He studies the addition of subtyping, then parametric polymorphism. In this last case, he also studies the decidability of the type inference problem. Finally, he studies an higher-order type system with recursion and a particular form of bounded universal quantification. This system, that is suitable for the typing of objects, can be intuitively viewed as a Curry style presentation of F-sub. He proves the soundness of this system.

The last part of the thesis is devoted to the study of objects in the blue calculus, and is based on [BDZ99] discussed above. Dal Zilio gives a typed interpretation of two popular object calculi, namely Abadi-Cardelli's object calculus – in its functional version, and in its concurrent version –, and the calculus of extensible objects defined by Fisher and Mitchell. The main contribution is a typed interpretation of the calculus of extensible objects that preserves subtyping.

Sangiorgi, in a collaboration with Francesca Levi (University of Pisa) [LS00] has studied interference in in Cardelli and Gordon's *Mobile Ambients* (MA). Two forms of interferences have been individuated: *plain interferences*, which are similar to the interferences one finds in CCS and π-calculus; and *grave interferences*, which are more dangerous and may be regarded as programming errors. To control interferences, the MA movement primitives are modified. On the new calculus, the *Safe Ambients* (SA), a type system is defined that: controls the mobility of ambients; removes all grave interferences. Other advantages of SA are: a useful algebraic theory; programs sometimes more robust (they require milder conditions for correctness) and/or simpler. All these points are illustrated on several examples.

Amadio, in a collaboration with S. Prasad (Indian Institute of Technology, Delhi, India) has worked on security aspects of distributed mobile systems [AP99]. A name-passing calculus is presented that can be regarded as a simplified π-calculus equipped with a *cryptographic table*. The latter is a data structure representing the relationships among names. The calculus is applied to the modeling and verification of secrecy and authenticity properties in cryptographic protocols relying on symmetric shared keys. Following classical approaches, the verification task is formulated as a reachability problem and its decidability proved assuming finite principals and bounds on the sorts of the messages synthesized by the attacker.

3.10.4 Publications

[ABL99] R. Amadio, G. Boudol, and C. Lhoussaine. The receptive distributed π-calculus. In *Proc. Found. Software Tech. and Theoret. Comp. Sci.*, LNCS. Springer-Verlag, 1999. To appear.

[AP99] R. Amadio and S. Prasad. The game of the name in cryptographic tables. In *Proc. ASIAN'99*, LNCS. Springer-Verlag, 1999. To appear. Also appeared as INRIA Research Report 3733.

[BCL99] G. Boudol, P.-L. Curien, and C. Lavatelli. A semantics for lambda calculi with resources. *Journal of Mathematical Structures in Computer Science*, 9(4), 1999. Special Issue on "Lambda-Calculus and Logic" in Honour of Roger Hindley.

[BDZ99] Gérard Boudol and Silvano Dal-Zilio. An interpretation of extensible objects. In *Proc. of FCT '99 – 12th International Symposium on Fundamentals of Computation Theory*, aug 1999.

[DZ99a] Silvano Dal-Zilio. A bisimulation for the blue calculus. Technical Report 3664, INRIA, apr 1999.

[DZ99b] Silvano Dal-Zilio. *Le calcul bleu: types et objets.* PhD thesis, Université de Nice – Sophia-Antipolis, 1999.

[LS00] F. Levi and D. Sangiorgi. Controlling interference in ambient. In *Proc. POPL'00.* ACM Press, 2000. to appear.

[Mer99] M. Merro. On Equators in Asynchronous Name-passing calculi without Matching. In I. Castellani and B. Victor, editors, *Proc. Express'99*, volume 27 of *Electronic Notes in Theoretical Computer Science.* Elsevier, 1999.

[NHKM99] U. Nestmann, H. Hüttel, J. Kleist, and M. Merro. Aliasing models for object migration. In P. Amestoy, P. Berger, M. Daydé, I. Duff, V. Frayssé,

L. Giraud, and D. Ruiz, editors, *EuroPar'99 Parallel-Processing*, volume 1685 of *LNCS*, pages 1353–1368. Springer-Verlag, 1999.

[RS99] C. Röckl and D. Sangiorgi. A pi-calculus semantics of Concurrent Idealised ALGOL. In *Proc. Fossacs'99*, volume 1578 of *LNCS*, pages 306–322. Springer-Verlag, 1999.

[San99a] D. Sangiorgi. Asynchronous process calculi: the first-order and higher-order paradigms (tutorial). To appear in TCS, 1999.

[San99b] D. Sangiorgi. From λ to π, or: Rediscovering continuations. *Journal of Mathematical Structures in Computer Science*, 9(4), 1999. Special Issue on "Lambda-Calculus and Logic" in Honour of Roger Hindley.

3.11 KTH

3.11.1 Research directions

We have explored the expressiveness of the fusion calculus, by showing that continuations may be removed both from inputs and outputs while keeping enough expressiveness to encode prefixes and guarded summation.

In this subcalculus – the fusion calculus of solos – we have presented two different encodings of the full calculus. The causal dependency between the prefix and the continuation is encoded first by the dependency between a match condition and its guarded agent, and secondly by the dependency between a scoped subject and a corresponding scope extrusion. In both encodings the causal dependencies are controlled by fusion effects – the same machinery cannot be used in the pi-calculus. Another important factor is that the calculus is polyadic: an input or output can carry arbitrarily many objects. In the monadic calculus we strongly conjecture that the expressiveness of prefixes is strictly greater than that of solos.

3.11.2 Persons and exchanges

During 1999, the following persons have been active within CONFER-2: Joachim Parrow and Björn Victor (Uppsala University). Victor visited University of Pisa in December 1999.

3.11.3 Publications

[LV99] Cosimo Laneve and Björn Victor. Solos in concert. In J. Wiederman, P. van Emde Boas and M. Nielsen, eds, *Proceedings of ICALP '99*, volume 1644 of *LNCS*, pages 513–523. Springer, July 1999.

3.12 Oxford University

3.12.1 Research activity

Work has been done on theory and applications of π-calculus.

The final version of a paper on semantic definition of concurrent object-oriented programming languages was prepared for publication [LW99]. The paper explains how a calculus that extends the π-calculus with first-order data and higher-order abstractions, but has only first-order interaction, can be used to give semantic definitions that are clear, precise, and tractable.

The paper [QW99a] studies the relationship between synchronous and asynchronous mobile processes, in the setting of the π-calculus. A primitive of the π-calculus, inherited from its ancestor CCS, is a form of handshake communication. This primitive is central to the tractability of CCS, π-calculus, and many other theories of concurrent systems that are based on similar operators.

On the other hand, many concurrent systems, especially distributed systems, use forms of asynchronous communication, in which the act of sending a datum and the act of receiving it are separate. Relatedly, many languages for programming concurrent or distributed systems have asynchronous primitives, an important reason for this being that they are amenable to efficient implementation. Language features for synchronized communication are often implemented using asynchronous primitives.

The π-calculus has a subcalculus (the asynchronous π-calculus) in which communication can be understood as asynchronous. The theory of the asynchronous subcalculus is much less tractable than that of the π-calculus. It is the subcalculus, however, that is the basis for concurrent programming language Pict; and the join language is based on the join calculus, an asynchronous calculus that is closely related to the asynchronous π-calculus.

The paper is concerned with the effect on behavioral equivalence of the standard translation from the π-calculus to the asynchronous π-calculus. It introduces a type system for terms of the target language (the monadic asynchronous π-calculus), and uses it to prove a full-abstraction theorem. The theorem states that two processes of the polyadic π-calculus are typed barbed congruent iff their translations into the subcalculus are asynchronous monadic typed barbed congruent.

The paper develops techniques introduced in [QW99b], and builds on earlier work on types for mobile processes, a topic that is important both for programming mobile systems and for analysing their behaviors.

3.12.2 Workshops and site visits

Working-Group funds were used to support the participation of David Walker in the Pisa workshop in March 1999. It is intended that they will also have supported the cost of accommodation for a visit to Oxford by Davide Sangiorgi from INRIA Sophia Antipolis where the main focus of interaction was joint work on π-calculus. (The bill for this accommodation has not yet been received.)

3.12.3 Publications

[LW99] X. Liu and D. Walker, *Concurrent objects as mobile processes*, in **Proof, Language, and Interaction**, editor G. Plotkin, MIT Press (to appear).

[QW99a] P. Quaglia and D. Walker, *On synchronous and asynchronous mobile processes*, submitted for publication.

[QW99b] 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, vol. 1530, 42–53.

3.13 Università di Pisa

3.13.1 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. In particular, it has been addressed the problem of verifying behavior of asynchronous name passing calculi. An algorithm to verify asynchronous bisimilarity for finitary π-calculus processes has been developed.

  We propose a specification language for distributed mobile systems based on asynchronous communication. The language is designed to support the composition of specifications: it allows to express the global properties of a system in terms of the local properties of its components and of coordination patterns. A preliminary work attemps at providing a refinement framework to deal with security issues.

- *Foundational Models and Abstract Machines*

  Some foundational models, namely *rewriting logic*, *action calculi* and *tile logic*, have been compared. For each of these logics we first try to understand their foundations, then we briefly sketch some applications. The overall goal of is to find out a common layout where these logics can be recasted, thus allowing for a comparison and an evaluation of their specific features.

  Research has continued on developing mathematical foundations of the tile model. The natural targets of the tile approach are distributed mobile systems which are compositional in both space and time. An executable specification of tile systems has been developed. This has been obtained by mapping tile logic back into rewriting logic. In particular, this implementation requires the development of a metalayer to control rewritings, i.e., to discard computations that do not correspond to any deduction in tile logic. Our methodology is applied to *term tile systems* that cover and extend a wide-class of SOS formats for the specification of process calculi. The case study of full CCS, where the term tile format is needed to deal with recursion (in the form of the replicator operator), is discussed in detail.

  We introduced the notion of *cartesian closed double category* to provide mobile calculi for communicating systems with specific semantic models: One dimension is dedicated to compose systems and the other to compose their computations and their observations. Also, inspired by the connection between simply typed λ-calculus and cartesian closed categories, we define a new typed framework, called *double λ-notation*, which is able to express the abstraction/application and pairing/projection operations in all dimensions. In this development, we take the categorical presentation as a guidance in the interpretation of the formalism. A case study of the π-calculus, where the double λ-notation straightforwardly handles name passing and creation, concludes the presentation.

Zero-safe nets have been introduced to extend classical Petri nets with a primitive notion of *transition coordination*. To this aim, besides ordinary places, called *stable*, zero-safe nets are equipped with *zero places*, which cannot contain any token in a stable marking . An evolution between two stable markings is called *transaction* and can be a complex computation that involves zero places, with the restriction that no stable token generated in a transaction can be reused in the same transaction. The abstract counterpart of a generic zero-safe net B consists of an ordinary PT net whose places are the stable places of B, and whose transitions are the transactions of B. The two nets offer the refined and the abstract model of the same system, where the former can be much smaller than the latter, because of the transition coordination mechanism. Depending on the chosen approach — *collective* vs *individual token philosophy* — two notions of transaction may be defined, each leading to different operational and abstract models.

- *Programming Languages*

  Research has been continued on developing both the implementation and the theoretical foundations of the experimental programming language Klaim (Kernel Language for Agents Interaction and Mobility). The Klaim language, by making use of notions of code mobility, gives the possibility of coordinating the activity of processes running on a net. Klaim security architecture exploits a capability–based type system to provide mechanisms for specifying and enforcing policies that control uses of resources and authorize migration and execution of processes. Several programming examples illustrate the expressive power and the flexibility of Klaim type system to support the specification of control policies and their enforcement. Finally, a prototype implementation of Klaim in `Ada95`, has been also developed.

3.13.2 Perspective

A strong interaction is expected between the work on tiles and the more applied development on languages for mobile computing. We intend to provide the mathematical foundations of existing calculi by exploiting tile logic and its higher order version. Moreover, research is under way to develop a process calculus representation of the KLAIM language in order to define notions of observations and of behavioral equivalences as a basis for verification. We also intend to check the theoretical developments by designing and implementing suggestive case studies, taking advantage of the KLAIM prototype.

About verification, we plan to extend the HD-automata techniques to the analysis of security protocols specified in the secure π-calculus. Moreover, we intend to address the problem of developing a specification logic for open and untrusted dynamic networks and its model checking techniques.

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

The foundational work on tiles will continue studying the double λ-notation associated to their cartesian closed version. Preliminary results about the early version of the π-calculus show the adequacy of this notation for modeling name passing and creation in a straightforward way. Also the obvious connections between coalgebras and tiles (which naturally define compositional transition systems and are already equipped with bisimulation equivalences and congruences) will be studied.

3.13.3 Persons

The group at Pisa involved in the project consists of Ugo Montanari, Gianluigi Ferrari, Roberto Bruni and Emilio Tuosto. A PhD-student, Emilio Tuosto, under the supervision of Ugo Montanari and Gianluigi Ferrari, is working on topics which are closely related to the aims of the CONFER 2 Working Group.

3.13.4 Publications

[BM99a] Bruni, R. and Montanari, U., Zero-Safe Nets: Composing Nets via Transition Synchronization, Int. Colloquium on Petri Net Technologies for Modelling Communication Based Systems, 1999, Fraunhofer Gesellschaft ISST.

[BMM99] Bruni, R. and Meseguer, J. and Montanari, U., Executable Tile Specifications for Process Calculi, FASE'99, Fundamental Approaches to Software Engineering, LNCS 1577, 1999.

[BMMS99] Bruni, R. and Meseguer, J. and Montanari, U. and Sassone, V., Functorial Semantics for Petri Nets under the Individual Token Philosophy, Category Theory and Computer Science, Elsevier ENTCS 29, 1999.

[BM99b] Bruni, R. and Montanari, U., Cartesian Closed Double Categories, their Lambda-Notation, and the Pi-Calculus, LICS'99, 14th Annual IEEE Symposium on Logic in Computer Science, IEEE Computer Society, 1999.

[CFP99] F. Corradini, G. Ferrari and M. Pistore. On the Semantics of Durational Actions. *Theoretical Computer Science*, to appear 1999.

[NFP99] R. De Nicola, G. Ferrari, and R. Pugliese. Types as Specification of Access Policies In *Secure Internet Programming: Security Issues for Distributed and Mobile Objects*, LNCS State-Of-The-Art-Survey (J. Vitek and C. Jensen Eds.), LNCS 1603, 1999.

[NFPV99] R. De Nicola, G. Ferrari, R. Pugliese and B. Venneri. Types for Access Control. In *Theoretical Computer Science*, to appear.

[FMSS99] Ferrari, G., Montangero, C., Semini, L. and Semprini, S. A Refinement Calculus for Mobility: Expressing Security Policies, In Proc. OOPSLA Distributed Object Security Workshop, 1999.

[GM99] F. Gadducci, U Montanari. Comparing Logics for Rewriting: Rewriting Logic, Action Calculi and Tile Logic. Submitted to a special issue of *Theoretical Computer Science*, 1999.

[MP99] Montanari , U. and Pistore, M. Finate State Verification for the Asynchronous π-calculus. In Proc. TACAS'99, LNCS 1579, 1999.

[Tu99] Tuosto, E. An `Ada95` Implementation of a Network Coordination Language with Code Mobility, Ada-Europe International Conference on Reliable Software Technologies Santander, Spain, June 1999

3.13.5 Networking

A CONFER2 workshop has been hold in Pisa from March 29 to March 31. The program of the workshop is reported below.

3.14 University of Sussex

3.14.1 Research directions

Asynchronous Mobile Processes Although asynchronous languages have recently become popular, at least as vehicles for implementation initiatives, there has been very little success in developing behavioral theories for such languages. As a step in this direction we investigated an asynchronous version of CCS and have found an equational inference systems which characterizes both may and must testing preorders for this language. Our hope is to extend this to the more expressive asynchronous pi-calculus. See CH98 for details.

Types for Distributed Processes During this period we investigated the use of types to ensure locality of resources in distributed systems. The research, reported in YH99 below, was carried out in a higher-order version of the pi-calculus, augmented with the primitives from the call-by-value lambda-calculus, in which there is also a primitive notion of location. We showed that it was possible to extend existing IO-type systems for the pi-calculus to ensure that all resources (in this case modeled by pi-calculus channels), even those created dynamically, can be serviced at at most one location.

Higher-order behavioral equivalences One major problem in developing behavioral theories for higher-order distributed calculi is the interaction between new name generation and functional abstraction. This interaction can be studied, in isolation from other distributed concepts, within an apparently very simple language called the nu-calculus, which consists of the lambda-calculus augmented with new name generation. In JR99 below it is shown that by adding assignment to this language bisimulation equivalence coincides with contextual equivalence. It is hoped that further work along these lines will help illuminate bisimulation equivalence for higher-order distributed languages.

3.14.2 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, Nobuko Yoshida.

Confer-2 resources were used to fund a research visit by Matthew Hennessy to INRIA-Sophia-Antipolis and to fund a research visit by Ilaria Castellani, from INRIA-Sophia-Antipolis to Sussex. In addition it supported (with permission) the attendance by Matthew Hennessy and Julian Rathke at the workshop HOOTS99.

3.14.3 Publications

[CH98] Ilaria Castellani and Matthew Hennessy. Testing theories for asynchronous languages. In V Arvind and R Ramanujam, editors, *18th Conference* *on Foundations of Software Technology and Theoretical Computer Science (Chennai, India, December 17–19, 1998)*, lncs. sv, December 1998.

[Hen99] Y. Yoshida M. Hennessy. Subtyping and locality in distributed higher order mobile processes (extended abstract). In *CONCUR '99*. Springer-Verlag, 1999. Full version available as Computer Science Technical Report 1/99, University of Sussex, 1997. Available from `http://www.cogs.susx.ac.uk/`.

[JR99] A. Jeffrey and J. Rathke. Towards a theory of bisimulation for local names. In *Proc. LICS'99*, 14^{th} *Annual Symposium on Logic in Computer Science, Trento*, pages 56–66. IEEE Computer Society Press, 1999.

Chapter 4

Deliverables

4.1 Workshop 5

The Fifth workshop was held in Pisa, organized by University of Pisa, with more than 30 participants.

Monday, March 29th

9:30 – 10:15	Klaim: An Overview R. De Nicola
10:15 – 11:00	On Behavioural Equivalence in Ambient Calculus F. Levi
11:00 – 11:30	Coffee-Break
11:30 – 12:15	An Interpretation of Extensible Objects S. Dal Zilio
12:15 – 13:00	Mobile Objects as Mobile Processes M. Merro
13:00 – 15:00	Lunch
15:00 – 15:45	Symmetric Action Calculi P. Gardner

Monday, March 29th

15:45 – 16:30	Comparing Logics for Rewritings: Rewriting Logic, Action Calculi and Tile Logic F. Gadducci
16:30 – 17:00	Coffee-Break
17:00 – 17:45	Demo X-Klaim Pogramming L. Bettini

Tuesday, March 30th

9:30 – 10:15	Static Analysis for Security of Processes P. Degano
10:15 – 11:00	Secure Composition of Insecure Components P. Sewell
11:00 – 11:30	Coffee-Break
11:30 – 12:15	Security Types for Mobile Agents M. Hennessy
12:15 – 13:00	Mobility Types for Mobile Ambients G. Ghelli
13:00 – 15:00	Lunch
15:00	Security for Mobile Agents (Panel Discussion) Moderator G. Ferrari
20:00	Social Dinner

Wednesday, March 31th

9:30 – 10:15	Cartesian Closed Double Categories R. Bruni
10:15 – 11:00	Weak Bisimulation and Open Maps G. Cattani

Monday, March 29th

11:00 – 11:30	Coffee-Break
11:30 – 12:15	Origing Tracking in the Lambda Calculus J.W. Klop
12:15 – 13:00	Hypercoherence as Polarized Games P-A. Mellies

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 6

The Fourth workshop was held in Nice, organized 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.

Thursday, November 25th

9h - 9h30	Registration
9h30 - 10h15	Structured Coalgebras, SOS Specifications and Tiles for Open Systems Ugo Montanari (Pisa) (with Andrea Corradini, Pisa, and Reiko Heckel, Paderborn)
10h15 - 10h45	Coffee Break
10h45 - 11h30	The π_F-calculus: a π-calculus with fusions Philippa Gardner (Cambridge) (with Lucian Wischik)
11h30 - 12h15	Assigning Types to Processes Matthew Hennessy (Sussex) (with Nobuko Yoshida)

Monday, March 29th

12:15 – 14h00	Lunch
2h - 2h45	On the duality of call-by-name and call-by-value Pierre-Louis Curien (Paris) (with Hugo Herbelin)
2h45 - 3h30	Observational Equivalences for a Mobile Process Calculus Asis Unyapoth (Cambridge)
3:30 - 4h00	Coffee Break
4h00 - 4h45	Secure Composition of Untrusted Code: Wrappers and Causality Types Peter Sewell (Cambridge) (with Jan Vitek, Purdue University)
4h45 - 5h30	A Logical Framework for the spi calculus Emilio Tuosto (Pisa)

Friday, November 26th

9:30 - 10:15	Double categories for calculi with name passing and name creation Roberto Bruni (Pisa)
10:15 - 10:45	Coffee Break
10:45 - 11:30	A translation of Ambients in the distributed Join calculus Alan Schmitt (INRIA)
11:30 - 12:15	Authentication primitives and their compilation Georges Gonthier (INRIA)
12:15 – 14h00	Lunch
2:00 - 2:45	Closed Reductions in the Lambda Calculus Ian Mackie (Paris)
2:45 - 3:30	The lambda-sigma calculus enjoys finite normalization cones Paul-Andre Mellies (Paris)

3:30 - 4:00	Coffee Break
4:00 - 4:45	Verifying arbitrary temporal formulas in the Temporal Logic of Actions Lucian Wischik (Cambridge)
4:45 - 5:30	Games theory and linear logic Vincent Danos (Paris)

List of Participants: Gerard Boudol, Roberto Bruni, Guiseppe Castagna, Ilaria Castellani, Sylvain Conchon, Vincent Danos, Philippa Gardner, Georges Gonthier, Matthew Hennessy, Hugo Herbelin, Jan Jürjens, Jean-Louis Krivine, Cosimo Laneve, Fabrice le Fessant, Jean-Jacques Lévy, Lone Leth, Ian Mackie, Luc Maranget, Paul-André Mellies, Massimo Merro, Ugo Montanari, Bent Thomsen, Emilio Tuosto, Peter Sewell, Asis Unyapoth, Alan Schmitt, Asis Unyapoth, Lucian Wischik

4.3 Software deliverables

The following is a listing of software constructed during the third 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

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

[FL99] C. Fournet, C. Laneve: *Bisimulations in the Join-Calculus.* Submitted to a Journal. June 1999.

[FLMR99] C. Fournet, C. Laneve, L. Maranget, D. Rèmy: *Assembling Objects with Internal Concurrency.* Draft, October 1999.

[LV99] C. Laneve, B. Victor: *Solos in Concert.* ICALP'99, LNCS 1644.

[CaPhD99] G. L. Cattani. Presheaf Models for Concurrency. PhD Thesis, University of Aarhus, 1999.

[FCW99] M. Fiore, G. L. Cattani and G. Winskel. Weak Bisimulation and Open Maps. In *Proceedings of the 14th Symposium on Logic in Computer Science, LICS '99*, pages 67–76, IEEE Press, 1999.

[PCW99] A. J. Power, G. L. Cattani and G. Winskel. A Representation Result for Free Cocompletions. To appear in *Journal of Pure and Applied Algebra*, 1999.

[Ga99a] Philippa Gardner. Closed Action Calculi. Journal of Theoretical Computer Science, 1999. To appear.

[GW99] Philippa Gardner and Lucian Wischik. Symmetric Action Calculi. Extended abstract, Express 99.

[Ga99b] Philippa Gardner. Interaction Nets, Sharing graphs and Action Calculi. Submission in association with the Mathfit Workshop on Process Algebra, January 99.

[SV99a] Peter Sewell and Jan Vitek. Secure composition of insecure components. Technical Report 463, Computer Laboratory, University of Cambridge, April 1999.

[SV99b] Peter Sewell and Jan Vitek. Secure composition of insecure components. In *Proceedings of the 12th IEEE Computer Security Foundations Workshop. Mordano, Italy*, pages 136–150. IEEE Computer Society, June 1999.

[SV99c] Peter Sewell and Jan Vitek. Secure Composition of Untrusted Code: Wrappers and Causality Types. Submitted for publication.

[Sew99a] Peter Sewell. From rewrite rules to bisimulation congruences. *Theoretical Computer Science*, 1999. Invited submission, to appear in a special issue for CONCUR 98.

[SWP99b] Peter Sewell, Paweł T. Wojciechowski, and Benjamin C. Pierce. Location-independent communication for mobile agents: a two-level architecture. In *Internet Programming Languages*, LNCS 1686. Springer-Verlag, 1999.

[WS99] Paweł T. Wojciechowski and Peter Sewell. Nomadic Pict: Language and infrastructure design for mobile agents. In *Proceedings of ASA/MA 99: Agent Systems and Applications/Mobile Agents*, IEEE, October 1999.

[Sew99pi] Peter Sewell. A brief introduction to applied π, January 1999. Lecture notes for the Mathfit Instructional Meeting on Recent Advances in Semantics and Types for Concurrency: Theory and Practice, July 1998.

[Da99] V. Danos.(1999) Sequent Calculus and Continuation Passing Style Compilation. To appear in the Proceedings of the 11th Congress of Logic, Methodology and Philosophy of Science, held in Cracow, Kluwer.

[AbMe99] S. Abramsky, P.-A. Melliès.(1999) Concurrent games and full completeness. Proceedings of LiCS'99, Trento.

[Me99a] P.-A. Melliès.(1999) Hypercoherence spaces as concurrent games, in preparation.

[Me99b] P.-A. Melliès.(1999) Axiomatic rewriting theory II: The lambda-sigma calculus enjoys finite normalization cones. Submitted.

[BDER98] P. Baillot, V. Danos, T. Ehrhard, L. Regnier.(1998) Timeless Games. In proceedings of CSL'97, held in Aarhus. Springer LNCS 1414, Mogens Nielsen and Wolfgang Thomas eds., pp. 56–77.

[MKl99] Jean-François Monin and Francis Klay, "Correctness Proof of the Standardized Algorithm for ABR Conformance", FM'99 – Formal Methods, LNCS 1708, Springer Verlag", ed. Jeannette Wing and Jim Woodcock and Jim Davies, pp 662-681, 1999

[LTL99] J.-J. Lévy, B. Thomsen, L. Leth: "Second Year Report for Esprit Working Group 21836 CONFER-2 CONcurrency and Funstions: Evaluation and Reduction", EATCS Bulletin Number 67 February 1998 pp. 35-51.

[DPLB99] P. Degano, C. Priami, L. Leth, B. Thomsen: "Causality for Debugging Mobile Agents", accepted for publication in ACTA INFORMATICA, 1999.

[AFG99a] Martín Abadi, Cédric Fournet, and Georges Gonthier. Secure implementation of channel abstractions. à paraître, 1999.

[AFG99b] Martín Abadi, Cédric Fournet, and Georges Gonthier. A top-down look at a secure message. In *Proc. of the 19th Foundations of Software Technology and Theoretical Computer Science, Chennai*, December 1999.

[AFG00] Martín Abadi, Cédric Fournet, and Georges Gonthier. Authentication primitives and their compilation. In *27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. ACM Press, January 2000.

[FLS99] Cédric Fournet, Jean-Jacques Lévy, and Alan Schmitt. A distributed implementation of ambients. 1999. à paraître.

[FS99] Cédric Fournet and Alan Schmitt. An implementation of Ambients in JoCAML. In *5th Mobile Object System Workshop: Programming Languages for Wide Area Networks*, June 1999.

[lF99] Fabrice le Fessant. Jocaml. available at `http://join.inria.fr`, 1999.

[LM99] Jean-Jacques Lévy and Luc Maranget. Explicit substitutions and programming languages. In *Proc. of the 19th Foundations of Software Technology and Theoretical Computer Science, Chennai*, December 1999.

[ABL99] R. Amadio, G. Boudol, and C. Lhoussaine. The receptive distributed π-calculus. In *Proc. Found. Software Tech. and Theoret. Comp. Sci.*, LNCS. Springer-Verlag, 1999. To appear.

[AP99] R. Amadio and S. Prasad. The game of the name in cryptographic tables. In *Proc. ASIAN'99*, LNCS. Springer-Verlag, 1999. To appear. Also appeared as INRIA Research Report 3733.

[BCL99] G. Boudol, P.-L. Curien, and C. Lavatelli. A semantics for lambda calculi with resources. *Journal of Mathematical Structures in Computer Science*, 9(4), 1999. Special Issue on "Lambda-Calculus and Logic" in Honour of Roger Hindley.

[BDZ99] Gérard Boudol and Silvano Dal-Zilio. An interpretation of extensible objects. In *Proc. of FCT '99 – 12th International Symposium on Fundamentals of Computation Theory*, aug 1999.

[DZ99a] Silvano Dal-Zilio. A bisimulation for the blue calculus. Technical Report 3664, INRIA, apr 1999.

[DZ99b] Silvano Dal-Zilio. *Le calcul bleu: types et objets.* PhD thesis, Université de Nice – Sophia-Antipolis, 1999.

[LS00] F. Levi and D. Sangiorgi. Controlling interference in ambient. In *Proc. POPL'00.* ACM Press, 2000. to appear.

[Mer99] M. Merro. On Equators in Asynchronous Name-passing calculi without Matching. In I. Castellani and B. Victor, editors, *Proc. Express'99*, volume 27 of *Electronic Notes in Theoretical Computer Science.* Elsevier, 1999.

[NHKM99] U. Nestmann, H. Hüttel, J. Kleist, and M. Merro. Aliasing models for object migration. In P. Amestoy, P. Berger, M. Daydé, I. Duff, V. Frayssé, L. Giraud, and D. Ruiz, editors, *EuroPar'99 Parallel-Processing*, volume 1685 of *LNCS*, pages 1353–1368. Springer-Verlag, 1999.

[RS99] C. Röckl and D. Sangiorgi. A pi-calculus semantics of Concurrent Idealised ALGOL. In *Proc. Fossacs'99*, volume 1578 of *LNCS*, pages 306–322. Springer-Verlag, 1999.

[San99a] D. Sangiorgi. Asynchronous process calculi: the first-order and higher-order paradigms (tutorial). To appear in TCS, 1999.

[San99b] D. Sangiorgi. From λ to π, or: Rediscovering continuations. *Journal of Mathematical Structures in Computer Science*, 9(4), 1999. Special Issue on "Lambda-Calculus and Logic" in Honour of Roger Hindley.

[LV99] Cosimo Laneve and Björn Victor. Solos in concert. In J. Wiederman, P. van Emde Boas and M. Nielsen, eds, *Proceedings of ICALP '99*, volume 1644 of *LNCS*, pages 513–523. Springer, July 1999.

[LW99] X. Liu and D. Walker, *Concurrent objects as mobile processes*, in **Proof, Language, and Interaction**, editor G. Plotkin, MIT Press (to appear).

[QW99a] P. Quaglia and D. Walker, *On synchronous and asynchronous mobile processes*, submitted for publication.

[QW99b] 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, vol. 1530, 42–53.

[BM99a] Bruni, R. and Montanari, U., Zero-Safe Nets: Composing Nets via Transition Synchronization, Int. Colloquium on Petri Net Technologies for Modelling Communication Based Systems, 1999, Fraunhofer Gesellschaft ISST.

[BMM99] Bruni, R. and Meseguer, J. and Montanari, U., Executable Tile Specifications for Process Calculi, FASE'99, Fundamental Approaches to Software Engineering, LNCS 1577, 1999.

[BMMS99] Bruni, R. and Meseguer, J. and Montanari, U. and Sassone, V., Functorial Semantics for Petri Nets under the Individual Token Philosophy, Category Theory and Computer Science, Elsevier ENTCS 29, 1999.

[BM99b] Bruni, R. and Montanari, U., Cartesian Closed Double Categories, their Lambda-Notation, and the Pi-Calculus, LICS'99, 14th Annual IEEE Symposium on Logic in Computer Science, IEEE Computer Society, 1999.

[CFP99] F. Corradini, G. Ferrari and M. Pistore. On the Semantics of Durational Actions. *Theoretical Computer Science*, to appear 1999.

[NFP99] R. De Nicola, G. Ferrari, and R. Pugliese. Types as Specification of Access Policies In *Secure Internet Programming: Security Issues for Distributed and Mobile Objects*, LNCS State-Of-The-Art-Survey (J. Vitek and C. Jensen Eds.), LNCS 1603, 1999.

[NFPV99] R. De Nicola, G. Ferrari, R. Pugliese and B. Venneri. Types for Access Control. In *Theoretical Computer Science*, to appear.

[FMSS99] Ferrari, G., Montangero, C., Semini, L. and Semprini, S. A Refinement Calculus for Mobility: Expressing Security Policies, In Proc. OOPSLA Distributed Object Security Workshop, 1999.

[GM99] F. Gadducci, U Montanari. Comparing Logics for Rewriting: Rewriting Logic, Action Calculi and Tile Logic. Submitted to a special issue of *Theoretical Computer Science*, 1999.

[MP99] Montanari , U. and Pistore, M. Finate State Verification for the Asynchronous π-calculus. In Proc. TACAS'99, LNCS 1579, 1999.

[Tu99] Tuosto, E. An `Ada95` Implementation of a Network Coordination Language with Code Mobility, Ada-Europe International Conference on Reliable Software Technologies Santander, Spain, June 1999

[CH98] Ilaria Castellani and Matthew Hennessy. Testing theories for asynchronous languages. In V Arvind and R Ramanujam, editors, *18th Conference on Foundations of Software Technology and Theoretical Computer Science (Chennai, India, December 17–19, 1998)*, lncs. sv, December 1998.

[Hen99] Y. Yoshida M. Hennessy. Subtyping and locality in distributed higher order mobile processes (extended abstract). In *CONCUR '99*. Springer-Verlag, 1999. Full version available as Computer Science Technical Report 1/99, University of Sussex, 1997. Available from `http://www.cogs.susx.ac.uk/`.

[JR99] A. Jeffrey and J. Rathke. Towards a theory of bisimulation for local names. In *Proc. LICS'99, 14^{th} Annual Symposium on Logic in Computer Science, Trento*, pages 56–66. IEEE Computer Society Press, 1999.