INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Team Comète

Concurrence, Mobilité et Transactions

Futurs

THEME COM

Table of contents

1. Team

Joint team with LIX (Laboratoire d'Informatique de l'Ecole Polytechnique) and CNRS.

Head of project-team
Catuscia Palamidessi [DR-INRIA]

Administrative assistant
Catherine Moreau

Staff member Inria
Fabrice Le Fessant [CR]

Staff member CNRS
Franck Valencia [CR. Since 1/10/2004]

Associated researchers
Bernadette Charron-Bost [CR CNRS]

Invited researchers
André Schiper [Professor, Ecole Polytechnique de Lausanne. Since 1/9/2004]
Maria Grazia Vigliotti [Researcher, Imperial College. From 1/12/2004 till 10/12/2004]
Elaine Pimentel [Assoc. Prof., Univ. Federal de Minas Gerais, Brazil. From 10/11/2004 till 20/11/2004]

Ph. D. student
Kostas Chatzikokolakis [Allocataire École Polytechnique. Since 1/10/2004]
Yuxin Deng [Allocataire École des Mines de Paris. Since 1/4/2004]
Axelle Ziegler [Allocataire École Normale Supérieure. Since 1/10/2004]

Post-doctoral fellow
Tom Chothia [Post Doctorant CNRS. Since 1/9/2004]
Jun Pang [Post Doctorant INRIA. Since 1/8/2004]

Student intern
Adrian Balan [Ecole Polytechnique. Since 1/11/2004]
Mohit Bhargava [IIT Delhi, India. From 1/5/2004 till 31/7/2004]
Jean-Baptiste Bianquis [DEA Programmation. From 1/4/2004 till 30/9/2004]
Kostas Chatzikokolakis [DEA Programmation. From 1/4/2004 till 30/9/2004]
Axelle Ziegler [DEA Programmation. From 1/4/2004 till 30/9/2004]

2. Overall Objectives

The research of the COMETE team focuses on the theoretical foundations of distributed and mobile systems. The project follows two main directions: the study, implementation and applications of the probabilistic π-calculus, a variant of the π-calculus, and the use of higher-order functional programming languages for distributed applications, in particular in the context of peer-to-peer systems.

Our main field of application are large-scale Distributed Mobile Systems (DMS) of computing devices of varying character providing diverse services. In this context, it is a daunting technical and scientific challenge to develop reasoning techniques which allow us to build systems guaranteeing that processes and data move in a secure, highly distributed network of devices which may individually exhibit failures but together work as a reliable, dependable system.

Formal *Specification and Verification* is of great help for system building and reasoning. The issue is to formally verifying whether a given system complies with a given specification typically expressed as temporal/spatial logic formulas, process expressions, or automata.

Model checking prevails in today's verification techniques. However, model checking usually needs a *finite-state* representation of systems, while most DMS are inherently open: there is no bound on the number of resources/devices that can be part of a system. In other words, many DMS's phenomena are best represented

in models providing for unbounded or infinite systems. We consider the challenging problem of extending model checking techniques, possibly by combining them with deductive techniques, for the verification of DMS in *unbounded or (infinite)* scenarios.

Fault tolerance is a fundamental issue of DMS as they must often provide reliable services despite the occurrence of various types of failure. The use of specifications enriched with *stochastic* information and *probabilistic* reasoning provides a powerful mathematical tool for analyzing DMS that may exhibit failures. For example, stochastic information with probabilistic techniques can be used for specifying the rate at which faulty communication channels drop messages and for verifying message-delivery properties of the corresponding system. The probabilistic specification and verification of DMS is one of goals of COMETE.

The highly distributed and mobile nature of the systems under consideration makes them more accessible and hence more vulnerable. *Security* is therefore crucial for these systems. The specification and verification of security properties has until now mainly addressed finite-state, deterministic processes (or protocols). We believe that more attention needs to be paid to infinite-state and probabilistic frameworks for the faithful modeling of features such as *nonce generation*, *cryptographic attacks*, and an *open number of participants*. Such features are prominently present in the DMS we are interested.

Our general goal is to provide rigorous theories and tools for the specification and verification of DMS. In particular, we shall deal with the following fundamental specific issues in the specification and verification of DMS: *Infinite (or Unbounded) Systems*, *Probabilistic Specifications* and *Specification and Verification of Security*. Our approach will involve the use of tools from Process Calculi, Constraint Technology and Probabilistic Methods. We shall introduce these tools before describing our project approach.

3. Scientific Foundations

3.1. Process calculi

Participants: Catuscia Palamidessi, Frank Valencia, Yuxin Deng, Jun Pang, Tom Chothia.

identification Calculi for expressing and formalizing the basic features of concurrent systems

Process calculi treat processes much like the λ-calculus treats computable functions. They provide a language in which the structure of *terms* represents the structure of processes together with an *operational semantics* to represent computational steps. For example, the term $P \parallel Q$, which is built from P and Q with the *constructor* $\parallel$, represents the process that results from the parallel execution of those represented by P and Q. An operational semantics may dictate that if P can evolve into P' in a computational step P' then $P \parallel Q$ can also evolve into $P' \parallel Q$ in a computational step.

An appealing feature of process calculi is their *algebraic* treatment of processes. The constructors are viewed as the *operators* of an algebraic theory whose equations and inequalities among terms relate process behavior. For instance, the construct $\parallel$ can be viewed as a commutative operator, hence the equation $P \parallel Q \equiv Q \parallel P$ states that the behavior of the two parallel compositions are the same. Because of this algebraic emphasis, these calculi are often referred to as *process algebras*.

Typically the operational semantics of process calculi interpret process term by using transitions (labeled or not) specifying its computational steps ([2]). A labeled transition $P \xrightarrow{\mu} Q$ specifies that P performs μ and then behaves as Q. The relations $\xrightarrow{\mu}$ are defined according to the process calculus under consideration. In the next section we shall see those for the π-calculus ([46][47]) which is perhaps the most prominent representative of calculi for mobile systems.

3.1.1. The π-calculus

In the early 90's Milner, Parrow, and Walker proposed the π-calculus ([46][47]), a small paradigm for concurrency similar to CCS (the calculus for Communicating Systems, [45]) but enriched with constructs to support the the novel and powerful notion of link mobility. This proposal has had a tremendous impact on the community of Formal Methods for Concurrency, and stimulated or influenced research in other areas too, like for instance Security (cfr. the spi-calculus, [20]).

3.1.2. The asynchronous π-calculus

The π-calculus, like CCS, models communication by handshaking, namely as a *synchronous* interaction of both partners (rules COM and CLOSE). A few years after the introduction of the π-calculus, Honda and Tokoro ([41]) and, independently, Boudol ([25]), proposed a variant which models asynchronous communication instead. This variant has become known under the name of asynchronous π-calculus (π_a-calculus for short).

3.1.3. π versus π_a: the trade-off between expressiveness and distributed implementation

The π_a-calculus became quickly very popular, for several reasons:

- it is an elegant model of asynchronous communication, more abstract and more symmetric than previously proposed calculi for asynchronous communication,
- it has been "faithfully" implemented ([52]),
- it is simpler than the π-calculus, because it has fewer constructs, and yet
- it was believed to have the same expressive power as the π-calculus. This equivalence was not formally proved, but there were several hints in this direction: Milner's encoding of the lambda calculus in the π-calculus was re-done for π_a ([25]), it was shown that output prefix can be simulated ([41][25]), and input-guarded choice as well ([51]). Note that this justifies the more recent presentations of the π_a-calculus, which include input-guarded choice as an explicit operator ([24][21]).

It was not only until some years later that the claim of equivalence was refuted: in [5] it was shown that the π-calculus is strictly more expressive than the π_a-calculus, in the sense that it is not possible to encode the first into the latter in a *uniform* way while preserving a *reasonable* semantics. Uniform essentially means homomorphic with respect to the parallel and the renaming operators, and reasonable means sensitive to the capability of achieving success in all possible computations. This result is based on the fact that in the π-calculus it is possible to define an algorithm for leader election in a symmetric network, while this cannot be done with the π_a-calculus. In [50] it was shown that the additional expressive power is due exactly to the mixed choice construct: choices with homogeneous guards (i.e. with input guards only, or output guards only) can be eliminated.

A consequence of the above results, however, is that the π-calculus cannot be implemented deterministically[1] in a fully distributed way. In fact, problems like the leader election in a symmetric network are known to have no deterministic solution in a distributed (asynchronous) system. The reason is that if processes follow a deterministic program then an adversary scheduler can always interleave the activities in such a way that the initial symmetry is never broken. See [54] for a proof of impossibility of this kind.

3.2. Specification logics

Participants: Catuscia Palamidessi, Frank Valencia.

identification Logics for expressing and formalizing properties of concurrent systems

In COMETE we are interested in verifying whether a given process satisfies certain properties. These properties are often expressed in some logical formalism.

[1] The term "deterministic" here means "non-probabilistic".

3.2.1. Hennesy-Milner's modal logic.

A way of expressing process specifications is by using a process logic. One such a logic is the Hennesy-Milner's modal logic. The discriminating power of this logic with respect to a finite processes (i.e., recursion-free processes) coincides with strong bisimilarity (see [58]). That is, two finite processes are strongly bisimilar if and only if they satisfy the same formulas in the Hennessy-Milner's logic.

3.2.2. Temporal logics.

Hennesy-Milner's logic can express local properties such as "an action must happen next" but it cannot express long-term properties such as "an action eventually happens". This kind of property, which falls into the category of *liveness properties* (expressing that "something good eventually happens"), and also *safety properties* (expressing that "something bad never happens") have been found to be useful for reasoning about concurrent systems. The modal logics attempting to capture properties of the kind above are often referred to as *temporal-logics*.

Temporal logics were introduced into computer science by Pnueli ([53]) and thereafter proven to be a good basis for specification as well as for (automatic and machine-assisted) reasoning about concurrent systems. Temporal logics can be classified into linear and branching time logics. In the *linear* case at each moment there is only one possible future whilst in the *branching* case at each moment time may split into alternative futures.

3.3. Infinite systems

Participants: Catuscia Palamidessi, Frank Valencia.

This research is carried over in cooperation with Biorn Victor (Uppsala University), Vijay Saraswat (IBM, USA), and Stefan Dantchev (University of Durham, UK)

identification Constraints and process calculi approaches for proving properties of infinite-state systems

Verifying infinite systems is a particularly challenging and a relatively new area. Practical applications of this are still at a preliminary stage.

3.3.1. Constraints approach

Constraint-based verification ([38][34]) has shown to be promising approach for infinite systems since a constraint formula is a natural symbolic representation of an infinite state set.

Open Constraint Satisfaction Problems have been recently introduced for specifying and solving constraints problems in highly distributed networks. In such a context typically there is no bound on the number of devices/resources that can be part of a given network. Algorithms for this kind of problems and their applications have been considered in [23][26][37]. Nevertheless little attention has been paid to the computational limits of these problems. I.e., studies establishing, for interesting classes of these problems are actually computationally solvable. This is certainly an issue when you allow unbounded number of resources as it is the case in DMS.

3.3.2. Process calculi approach

The study of expressive power of different forms of specifying infinite-behavior in Process Calculi is a recent line of research bringing understanding for infinite behavior of concurrent systems in terms of decidability.

Our work in [3] (see also [4][6]), to our knowledge the first of this kind, deepened the understanding of process calculi for concurrent constraint programming by establishing an expressive power hierarchy of several temporal ccp languages which were proposed in the literature by other authors. These calculi, differ in their way of defining infinite behavior (i.e., replication or recursion) and the scope of variables (i.e., static or dynamic scope). In particular, it is shown that (1) recursive procedures with parameters can be encoded into parameterless recursive procedures with dynamic scoping, and vice-versa; (2) replication can be encoded into parameterless recursive procedures with static scoping, and vice-versa; (3) the calculi from (1) are strictly more expressive than the calculi from (2). Moreover, it is shown that the behavioral equivalence for these

calculi is undecidable for those from (1), but decidable for those from (2). Interestingly, the undecidability result holds even if the variables in the corresponding languages take values from a fixed finite domain whilst the decidability holds for arbitrary domains. The works [27][28][29] present similar results in the context of the calculus for communicating systems (CCS).

Both the expressive power hierarchy and decidability/undecidability results give theoretical distinctions among different ways of expressing infinite behavior. The above work, however, pay little attention to the existence efficient algorithms for the corresponding decidability questions or the existence of semi-decision procedures for the undecidable cases. These issues are fundamental if we wish to verify infinite-state process specifications, and hence we shall address it in this project.

3.4. Security

Participants: Catuscia Palamidessi, Frank Valencia, Kostas Chatzikokolakis.

identification Formalisms to express security properties and protocols and to verify them

Security protocols, also known as cryptographic protocols, are small concurrent programs designed to provide various security services across a distributed system. These goals include: authentication of agents and nodes, establishing session keys between nodes, ensuring secrecy, integrity, anonymity, non-repudiation, fairness, and so on. The challenge comes from the fact that we want to guarantee security of exchanges between participants using non-secure mediums, whose weaknesses can be exploited by malicious adversaries. In certain cases, like in the non-repudiation and fairness problems, we cannot even be sure that the participants are honest.

With the increasing degree of distribution and mobility of modern systems, and the increasing number of applications such as electronic commerce, electronic vote, etc, these protocols are becoming more and more used, and their correctness more and more crucial. Establishing the correctness of these protocols, however, is not an easy task; the difficulties arise from a number of considerations:

- The properties that they are supposed to ensure are extremely subtle; the precise meaning of a property is often a matter of debate and needs to be formally specified.
- The capabilities of adversaries (intruders, attackers, ...) are difficult to capture.
- By their nature security protocols involve a high degree of concurrency, which makes the analysis much more complicated.

Several formalisms have been proposed for the specification of the protocols and intruders, for the description of the security properties, and for proving correctness. For example, the Strand spaces ([39][30]), the spi-calculus ([20]) and other process calculi ([42][55][56][22]), formalisms based on linear logic ([33][44]), on set-rewriting ([43][31]), on rewriting logic ([35]), on tree automata ([49][40]), and on set constraints ([32]).

4. Application Domains

4.1. Panorama

Keywords: *distributed applications*, *distributed systems*, *mobile systems*, *security*, *telecommunications*.

The foundational research of COMETE (process calculi, communication and mobility, probablistic studies, semantics and logics for concurrency, etc.) and the software tools we develop address the needs of many application domains. They are virtually applicable to any system or protocol made of distributed agents communicating by asynchronous messages, and where, possibly, the communication structure can change dynamically. Here we list the main domains of applications we envisage:

- Distributed and mobile systems: election algorithms, dynamic reconfiguration algorithms, fault tolerance algorithms;

- Databases: transaction protocols, distributed knowledge bases;
- Security protocols: authentication, electronic transactions;
- Telecommunications: mobile telephony, active network management, hot reconfigurations, feature interaction detection;

5. Software

5.1. MLdonkey

MLdonkey is a multi-networks peer-to-peer file-sharing client. It is widely used in the Unix community (more than 10,000 users), and allows connections to the most important networks (eDonkey, Overnet, Kazaa, Gnutella, etc...). It is available at http://www.mldonkey.net. See also the forum for mldonkey users: http://www.mldonkeyworld.com/.

5.2. Ocaml Memory Profiler

The *Ocaml Memory Profiler* is a tool to instrument Objective-Caml programs, to gather information about how memory is used during an execution, and how the memory footprint of a program can be decreased. It is available at http://pauillac.inria.fr/~lefessan/src/memprof-ocaml-3.07.tar.gz.

6. New Results

6.1. The probabilistic asynchronous π-calculus

As discussed in Section 3.1.3, the π-calculus cannot be implemented in a fully distributed way. However, this is true only if we require the exact preservation of the semantics. The picture changes if we consider probabilistic methods, in fact distributed agreement can be achieved "with probability 1" by using randomized algorithms. Thus, the goal can be realized if we are willing to accept an implementation that introduces divergences, provided that they happen with probability 0.

In [7] we have proposed an extension of the π_a-calculus, to the purpose of using it as an intermediate language for the implementation of the π-calculus. Of course, in order to be able to write a randomized encoding, we needed to enhance π_a with a construct for random draws. Furthermore, we wanted the implementation to be robust with respect to adverse conditions, namely "bad interleaving sequences". In other words we needed to express, in the execution model of the intermediate language, the nondeterministic (i.e. unpredictable) decisions of an external scheduler. Thus we needed two notions of choice: one probabilistic, associated to the random draws controlled by the process, and one nondeterministic, associated to the possible interleavings generated by the scheduler.

Our proposal, π_{pa} (probabilistic asynchronous π-calculus) is based on the model of probabilistic automata of Segala and Lynch ([57]), which has the above discussed characteristic of distinguishing between probabilistic and nondeterministic behavior.

6.1.1. *Encoding π in π_{pa}*

In [8] we have defined a uniform, compositional encoding from π to π_{pa}. Our encoding involves solving a resource allocation problem that can be regarded as a generalization of the dining philosophers problem, in the sense that there may be more philosophers than forks [1]. The correctness of the encoding is proved with respect to a notion of testing semantics adapted to the probabilistic asynchronous pi-calculus. More precisely, we have shown that our encoding is correct with probability 1 with respect to any adversary.

6.1.2. *Implementation of π_{pa}*

An implementation of π_{pa} into a Java-like language was outlined in [7]. The idea is to compile π processes into threads, and channels as (shared) objects with synchronized methods for the send and receive operations.

Although Java is shared-memory, we regard the proposed implementation as a basis for a distributed implementation. In fact, in the compiled code, the threads corresponding to processes communicate only via the objects representing the channels. Furthermore the translation is distributed in the strong sense of being homomorphic with respect to the parallel operator. Namely, the translation does not introduce any "central process" to coordinate the activities of the treads corresponding to the original processes.

In the future we plan to develop a fully distributed, efficient, and failure-robust implementation for pi_{pa}. Copying with failures may influence the design of the language as well.

6.2. Axiomatizations of Probabilistic Finite-State Behaviors

One of the medium-term goals of COMETE is to provide an axiomatization for the probabilistic asynchronous π-calculus described in Section 6.1.

Some preliminary steps in this direction have been achieved in [12]. In that paper we have studied a process calculus which combines both nondeterministic and probabilistic behavior in the style of Segala and Lynch's probabilistic automata. We have considered various strong and weak behavioral equivalences, and we have provided complete axiomatizations for finite-state processes, restricted to guarded definitions in case of the weak equivalences. We conjecture that in the general case of unguarded recursion the "natural" weak equivalences are undecidable.

This is the first work, to our knowledge, that provides a complete axiomatization for weak equivalences in the presence of recursion and both nondeterministic and probabilistic choice.

6.3. Metrics for Quantitative Transition Systems

In systems that model quantitative processes, steps are associated with a given quantity, such as the probability that the step will happen or the resources (e.g. time or cost) needed to perform that step. The standard notion of bisimulation can be adapted to these systems by treating the quantities as labels, but this does not provide a robust relation, since quantities are matched only when they are identical. Processes that differ for a very small probability, for instance, would be considered just as different as processes that perform completely different actions. This is particularly relevant to security systems where specifications can be given as perfect, but impractical processes and other, practical processes are considered safe if they only differ from the specification with a negligible probability.

To find a more flexible way to differentiate processes, we have considered the notion of metric, which is a function that associates a real number (distance) with a pair of elements. In [18], we have studied metric semantic for a general framework that we call *Action-labeled Quantitative Transition Systems* (AQTS). This framework subsumes some other well-known quantitative systems such as probabilistic automata ([57]), reactive and generative models ([59]), and (a simplified version of) weighted automata ([36][48]).

The metric semantics that we have investigated in [18] is based on rather sophisticated techniques. In particular, we needed to resort to the notion of Hutchinson distance.

Still in [18], we have considered two extended examples which show that our results apply to both probabilistic and weighted automata as special cases of AQTS. In particular, we have shown that the operators of the corresponding process algebras are non-expansive, which is the metric correspondent of the notion of congruence.

Our ultimate goal is to investigate metric semantics for the probabilistic asynchronous π-calculus described in Section 6.1.

6.4. Probabilistic Anonymity

The concept of anonymity comes into play in a wide range of situations, varying from voting and anonymous donations to postings on bulletin boards and sending mails. A formal definition of this concept has been given in literature in terms of nondeterminism.

In [17], we have investigated a notion of anonymity based on probability theory, and we show that it can be regarded as a generalization of the nondeterministic one. We have then formulated this definition in terms

of observables for processes in the probabilistic asynchronous π-calculus, and proposed a method to verify automatically the anonymity property. We have illustrated the method by using the example of the dining cryptographers.

6.5. Deacidability results for Linear Temporal Logic

The ntcc process calculus [3][4] is a timed concurrent constraint programming (ccp) model equipped with a first-order linear-temporal logic (LTL) for expressing process specifications. A typical behavioral observation in ccp is the strongest postcondition. In ntcc strongest postcondition denotes the set of all infinite output sequences that a given process can exhibit. The verification problem is then whether the sequences in the strongest postcondition of a given process satisfy a given ntcc LTL formula.

In [9] we have established new positive decidability results for timed ccp as well as for LTL. In particular, we have proved that the following problems are decidable: (1) The strongest postcondition equivalence for the so-called locally-independent ntcc fragment; unlike other fragments for which similar results have been published, this fragment can specify infinite-state systems. (2) Verification for locally-independent processes and negation-free first-order formulas of the ntcc LTL. (3) Implication for such formulas. (4) Satisfiability for a first-order fragment of Manna and Pnueli' LTL. The purpose of the last result is to illustrate the applicability of ccp to well-established formalisms for concurrency.

6.6. Infinite Behavior and Name Scoping in Process Calculi

In literature there are several process calculi differing in the constructs for the specification of infinite behavior and in the scoping rules for channel names. In [13] we have studied various representatives of these calculi based upon both their relative expressiveness and the decidability of divergence. We regard any two calculi as being equally expressive if and only if for every process in each calculus, there exists a weakly bisimilar process in the other. By providing weak bisimilarity preserving mappings among the various variants, we showed that in the context of relabeling-free and finite summation calculi: (1) CCS with parameterless (or constant) definitions is equally expressive to the variant with parametric definitions. (2) The CCS variant with replication is equally expressive to that with recursive expressions and static scoping. We also stated that the divergence problem is undecidable for the calculi in (1) but decidable for those in (2). We obtain this from (un)decidability results by Busi, Gabbrielli and Zavattaro, and by showing the relevant mappings to be computable and to preserve divergence and its negation. From (1) and the well-known fact that parametric definitions can replace injective relabellings, we showed that injective relabellings are redundant (i.e., derived) in CCS (which has constant definitions only).

6.7. Open Constraint Satisfaction Problems

In [11] we have investigated the computational limits of open constraint satisfaction problems. We have proposed a new framework which we have used to identify some interesting classes which are not computationally solvable as well as connections to *automata over infinite-sequences*.

In particular, we have studied open constraint satisfaction problems allowing *infinitely*, or *unboundedly*, many *indexed* variables as in, e.g., $x_i > x_{i+2}$ for each $i = 1, 2, \ldots$. We first showed that (1) if the indices are one-dimensional and specified in the theory of the natural numbers with linear order (the theory of $(\mathbb{N}, 0, successor, <)$) then the satisfiability problem is *decidable*. We then proved that, in contrast to (1), (2) if we move to the two-dimensional case then the satisfiability problem is *undecidable* for indices specified in $(\mathbb{N}, 0, successor, <)$ and even in $(\mathbb{N}, 0, successor)$. Finally, we showed that, in contrast to (1) and (2), already in the one-dimensional case (3) if we also allow addition, we get undecidability. I.e., if the one-dimensional indices are specified in *Presburger arithmetic* (i.e., the theory of $(\mathbb{N}, 0, successor, <, +)$) then satisfiability is *undecidable*.

6.8. Efficient Arc-Consistency techniques for Constraint Satisfaction Problems

Arc-Consistency (AC) techniques have been used extensively in the study of Constraint Satisfaction Problems (CSP). These techniques are used to simplify the CSP before or during the search for its solutions.

Some of the most efficient algorithms for AC computation are AC6++ and AC-7. The novelty of these algorithms is that they satisfy the so-called four desirable properties for AC computation. The main purpose of these interesting properties is to reduce as far as possible the number of constraint checks during AC computation while keeping a reasonable space complexity.

In [16] we have proved that, despite providing a remarkable reduction in the number of constraint checks, the four desirable properties do not guarantee a minimal number of constraint checks, thus refusing the minimality claim in the paper introducing these properties. Furthermore, we have proposed a new desirable property for AC computation and extended AC6++ and AC-7 to consider such a property. We have showed theoretically and experimentally that the new property provides a further substantial reduction in the number of constraint checks.

6.9. Distributed Algorithms

We studied the solvability of agreement algorithms in synchronous systems with failures, from their complexity point of view. In particular, we designed algorithms to solve a family of agreement problems, called the k-Tag problems, in the optimal number of rounds depending only on the number of failures appearing in the system. k-Tag problems share the *Uniformity* and *Termination* conditions of the Consensus problem, while the *Validity* condition is parameterized on k. This family includes problems where the decided value must be proposed by a majority of members, together with the well-known Uniform Consensus and Non-Blocking Atomic Commitment problems. This work has been published in SOFSEM'2004 [10].

6.10. Peer-to-peer algorithms

This work has been done in collaboration with Anne-Marie Kermarrec (IRISA), Laurent Massoulie (Microsoft Research Cambridge), Simon Patarin (University of Bologna) and Sidath Handurukande (EPFL). We designed a protocol to improve searches in file-sharing networks, using semantic links between peers based on semantic clustering (the proximity of users' tastes). The originality of this work, published in SIGOPS'2004 [14], resides in the use of real traces we collected from the Edonkey network to measure the efficiency of the algorithm. These traces were collected during a few months using a modified version of MLdonkey, a peer-to-peer client written in a functional language, Objective-Caml by Fabrice Le Fessant. An analysis of these traces also appeared in the IPTPS'2004 workshop [15] and a more detailed one has been submitted recently to the conference NSDI'2005 [19]. We also worked on the collection of another trace of the encrypted part of the Kazaa protocol, to obtain real information on the publications and requests issued by the clients, together with the topology on the network, which are currently not available in the research community and necessary to evaluate discovery protocols and to model real peer-to-peer networks. The interest of this topic is shown by the increasing number of collaborators joining our working group, such as Matthieu Latapy (LIAFA), Etienne Rivière (IRISA) and Laurent Viennot (INRIA).

6.11. Memory Profiling

Distributed applications differ from standard applications by their lifetime, ranging from several days to several months, and thus, relying heavily on the efficiency of the automatic memory-management to reclaim unnecessary memory. In particular, a special kind of memory leak appears almost only in these applications: unnecessary data which is however reachable from a global root, and thus cannot be reclaimed by automatic memory-management. We worked on a memory profiler for Objective-Caml, that could be used to analyze the memory footprint of running applications (using snapshots dumped in files), to detect where the memory is used, and which global roots might be responsible for memory leaks. A first version of this profiler was released in the current of the year.

7. Other Grants and Activities

7.1. Actions nationales

7.1.1. Project ACI Securité ROSSIGNOL

Participants: Catuscia Palamidessi, Kostas Chatzikokolakis.

The project ROSSIGNOL has started in 2003 and includes the following participants:

- LIF. Responsable: D. Lugiez
- INRIA Futurs. Responsable: C. Palamidessi
- LSV. Responsable: F. Jacquemard
- VERIMAG.Responsable: Y. Lakhnech

ROSSIGNOL focuses on the foundations of Security Protocols. The goal of this project is the development of abstract models, simple enough to be used for the definition of a comprehensible semantics for the language of security properties. In particular, the project focuses on probabilistic models.

7.2. Actions internationales

7.2.1. Integrated Action within the EGIDE/PAI PICASSO program

Participants: Catuscia Palamidessi, Frank Valencia, Kostas Chatzikokolakis, Axelle Ziegler.

The EGIDE/PAI program PICASSO aims at promoting the scientific and technological exchanges between France and Spain. The equip COMETE is participating, within this program, to a project whose participants are:

- INRIA Futurs. Responsables: C.Palamidessi and D. Miller
- Universidad Politécnica de Madrid. Responsables: James Lipton and Manuel Hermenegildo

The main aims of our project are the integration of the approaches developed by the INRIA and the UPM teams to the analysis and implementation of Higher-Order Languages (both sequential and concurrent), coinductive techniques (with special emphasis on lazy features), and in the areas of code validation, proof carrying code and security.

This project has been accepted in 2004 and will start its activities in January 2005.

8. Dissemination

8.1. Services to the Scientific Community

Note: In this section we include only the activities of the permanent internal members of COMETE.

8.1.1. Organization of seminars

- Frank D. Valencia and Kostas Chatzikokolakis are the organizers of the Comète-Parsifal Seminar. This seminar takes place weekly at LIX, and it is meant as a forum where the members of Comète and Parsifal present their current works and exchange ideas. See http://www.lix.polytechnique.fr/comete/seminarFrame.html.
- Catuscia Palamidessi has been the co-organizer (together with Pierre-Louis Curien and Vincent Danos) of the “Groupe de travail en Concurrence” (Working group on Concurrency) at PPS, University of Paris VII, during the academic year 2003/04.

8.1.2. Editorial activity

- Catuscia Palamidessi is member of the Editorial Board of the journal on Theory and Practice of Logic Programming, published by the Cambridge University Press.
- Catuscia Palamidessi is member of the Editorial Board of the Electronic Notes of Theoretical Computer Science, Elsevier Science.
- Frank D. Valencia is area editor (for the area of Concurrency) of the ALP Newsletter.

8.1.3. Organization of conferences

- Catuscia Palamidessi is the Program Committee Chair of ICALP 2005 Track B. 32nd International Colloquium on Automata, Languages and Programming. Lisboa, Portugal, 11-15 July 2005.

8.1.4. Participation in program committees

Catuscia Palamidessi has been a member of the program committees of the following conferences:

- ICLP 2005. International Conference on Logic Programming. Barcelona, Spain, October 2005.
- CONCUR 2005. International Conference on Concurrency Theory. San Francisco, California, USA, August 2005.
- ESOP 2005. European Symposium on Programming. (Part of ETAPS 2005.) Edinburgh, Scotland, April 2005.
- SOFSEM 2005 Track on Foundations of Computer Science. 31st Annual Conference on Current Trends in Theory and Practice of Informatics, Liptovsky Jan, Slovak Republic, January 2005.
- CONCUR 2004. International Conference on Concurrency Theory. London, UK, August 2004.
- TCS 2004. Track 2. 3rd IFIP International Conference on Theoretical Computer Science. Toulouse, France, August 2004.
- ICALP 2004 Track B. 31st International Colloquium on Automata, Languages and Programming. Turku, Finland, July 2004.

Catuscia Palamidessi has been a member of the program committees of the following workshops:

- FInCo 2005. Workshop on the Foundations of Interactive Computation. Edinburgh, Scotland, April 200
- EXPRESS'04. 10th International Workshop on Expressiveness in Concurrency. London, UK, August 2004.
- DCFS 2004. Workshop on Descriptional Complexity of Formal Systems. London, Canada, July 2004.
- JFPLC 2004 Treizièmes Journées Francophones de Programmation en Logique et de programmation par Contraintes. Angers, France, June 2004.
- SBLP 2004 Brazilian Symposium on Programming Languages. Niterói, Brazil, May 2004.

Frank D. Valencia has been a member of the program committee of:

- ICLP 2005. 21st International Conference on Logic Programming. Barcelona, Spain, October 2005.

Fabrice Le Fessant has been a member of the program committee of:

- ALGOTEL 2005, Septièmes Rencontres Francophones sur les aspects Algorithmiques des Télécommunications

8.1.5. Reviews

8.1.5.1. Reviews of journal papers:

ACM Transactions on Programming Languages, Theoretical Computer Science, Journal of Algebraic and Logic Programming, Information and Computation.

8.1.5.2. Reviews of conference papers:

ESOP 2005, ICDCS 2005, SOFSEM 2005, CONCUR 2004, ICALP 2004, SOFSEM 2004, TCS 2004, ESOP 2003, POPL 2003.

8.1.6. Programming contexts

- Fabrice Le Fessant was Technical Director of the South-Western European Regional Contest of the ACM International Collegial Programming Contest, November 2004.

8.2. Teaching

Note: In this section we include only the activities of the permanent internal members of COMETE.

8.2.1. Courses in advanced schools:

- Catuscia will teach a course on "The process-calculus approach to security" at the CIMPA-UNESCO School on Security of Computer Systems and Networks. Bangalore, INDIA, Jan-Feb 2005.
- Fabrice Le Fessant has been teaching (with Cédric Fournet) a course on "Programming in JoCaml" at the Advanced Functional Programming Summer School, Oxford, August 2002.

8.2.2. Postgraduate courses:

- Catuscia Palamidessi is co-teaching (together with Jean-Jacques Lévy, Pierre-Louis Curien, Erik Gobault and James Leifer) the course "Concurrence" at the "Master Parisien de Recherche en Informatique" MPRI in Paris. Winter semester 2004-05.
- Catuscia Palamidessi has been teaching a Course on "Probabilistic Methods in Concurrency" at the PhD program in Pisa. July 2004.
- Catuscia Palamidessi has been co-teaching (together with Jean-Jacques Lévy) the course "Concurrence" at the D.E.A. Sémantique, Preuve et Programmation in Paris. Winter semester 2003-04.

8.2.3. Undergraduate courses:

- "Optimisation de programmes" (programs optimization), Fabrice Le Fessant (with Behshad Behzadi), École Polytechnique, training for the ACM Programming Contest, May 1994.
- Majeure "Programmation Système sous Unix", Fabrice Le Fessant (with Didier Rémy), January 2004.

8.3. Internship supervision

The team COMETE has supervised the following internship students during 2004:

- AdrianBalan (Ecole Polytechnique. Intern in COMETE since 1/11/2004)
- MohitBhargava (IIT Delhi, India. Intern in COMETE during 1/5–31/7/2004)
- Jean-BaptisteBianquis(DEA Programmation. Intern in COMETE during 1/4–30/9/2004)
- Kostas Chatzikokolakis(DEA Programmation. Intern in COMETE during 1/4–30/9/2004)
- Axelle Ziegler(DEA Programmation. Intern in COMETE during 1/4–30/9/2004)

9. Bibliography

Major publications by the team in recent years

[1] O. M. HERESCU, C. PALAMIDESSI. *On the Generalized Dining Philosophers Problem*, in "Proceedings of the 20th ACM Symposium on Principles of Distributed Computing", 2001, p. 81–89, http://www.lix.polytechnique.fr/~catuscia/papers/Gen_Phil/podc.ps.

[2] R. MCDOWELL, D. MILLER, C. PALAMIDESSI. *Encoding transition systems in sequent calculus*, in "Theoretical Computer Science", vol. 294, n° 3, 2003, p. 411-437, http://www.lix.polytechnique.fr/~catuscia/papers/Tran_Sys_in_SC/tcs.ps.

[3] M. NIELSEN, C. PALAMIDESSI, F. D. VALENCIA. *On the expressive power of temporal concurrent constraint programming languages*, in "Proceedings of the Fourth ACM SIGPLAN Conference on Principles and Practice of Declarative Programming", ACM Press, October 6–8 2002, p. 156–167, http://www.lix.polytechnique.fr/~catuscia/papers/Ntcc/ppdp02.ps.

[4] M. NIELSEN, C. PALAMIDESSI, F. VALENCIA. *Temporal Concurrent Constraint Programming: Denotation, Logic and Applications*, in "Nordic Journal of Computing", vol. 9, 2002, p. 145–188, http://www.lix.polytechnique.fr/~catuscia/papers/Ntcc/njc02.ps.

[5] C. PALAMIDESSI. *Comparing the Expressive Power of the Synchronous and the Asynchronous pi-calculus*, in "Mathematical Structures in Computer Science", vol. 13, n° 5, 2003, p. 685–719, http://www.lix.polytechnique.fr/~catuscia/papers/pi_calc/mscs.pdf.

[6] F. VALENCIA. *Concurrency, Time and Constraints*, in "Proc. of the Nineteenth International Conference on Logic Programming (ICLP 2003)", LNCS, Springer-Verlag, 2003, p. 72–101.

Articles in referred journals and book chapters

[7] O. M. HERESCU, C. PALAMIDESSI. *Probabilistic Asynchronous π-calculus*, in "Theoretical Computer Science", to appear, 2004, http://www.lix.polytechnique.fr/~catuscia/papers/Prob_asy_pi/report.ps.

[8] C. PALAMIDESSI, O. M. HERESCU. *A randomized encoding of the π-calculus*, in "Theoretical Computer Science", to appear, 2004, http://www.lix.polytechnique.fr/~catuscia/papers/prob_enc/report.pdf.

[9] F. VALENCIA. *Decidability of Infinite-State Timed CCP Process and First-Oder LTL*, in "Theoretical Computer Science", to appear, 2004, http://www.brics.dk/~fvalenci/tcs.pdf.

Publications in Conferences and Workshops

[10] B. CHARRON-BOST, F. LE FESSANT. *Validity Conditions in Agreement Problems*, in "Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM'2004)", January 2004.

[11] S. DANTCHEV, F. VALENCIA. *On the Computational Limits of Infinite Satisfaction*, in "Proc. of SAC2005", to appear, ACM Press, 2004, http://www.brics.dk/~fvalenci/icsp_sac05.pdf.

[12] Y. DENG, C. PALAMIDESSI. *Axiomatizations for probabilistic finite-state behaviors*, in "Proceed-

ings of FOSSACS'05", Lecture Notes in Computer Science, to appear, Springer-Verlag, 2004, http://www.lix.polytechnique.fr/~catuscia/papers/Prob_Axiom/fossacs05.pdf.

[13] P. GIAMBIAGI, G. SCHNEIDER, F. D. VALENCIA. *On the Expressiveness of Infinite Behavior and Name Scoping in Process Calculi*, in "Proceedings of FOSSACS 04", Lecture Notes in Computer Science, vol. 2987, Springer-Verlag, 2004, p. 226–240.

[14] S. HANDURUKANDE, ANNE-MARIE. KERMARREC, F. LE FESSANT, L. MASSOULIÉ. *Exploiting Semantic Clustering in the eDonkey P2P Network*, in "SIGOPS'2004", 2004.

[15] F. LE FESSANT, S. HANDURUKANDE, A.-M. KERMARREC, L. MASSOULIÉ. *Clustering in Peer-to-Peer File Sharing Workloads*, in "IPTPS'04", 2004.

[16] C. RUEDA, F. VALENCIA. *Non-Viability Deductions in Arc-Consistency Computation.*, in "Proc. of the Nineteenth International Conference on Logic Programming (ICLP 2004)", Lecture Notes in Computer Science, Springer-Verlag, 2004, p. 343–355.

Internal Reports

[17] M. BHARGAVA, C. PALAMIDESSI. *Probabilistic Anonymity*, Submitted, Technical report, INRIA Futurs, 2004, http://www.lix.polytechnique.fr/~catuscia/papers/Anonymity/BP.pdf.

[18] T. CHOTHIA, Y. DENG, C. PALAMIDESSI, J. PANG. *Metrics for Action-labelled Quantitative Transition Systems*, Submitted, Technical report, 2004, http://www.lix.polytechnique.fr/~catuscia/papers/Metrics/gts.pdf.

[19] S. HANDURUKANDE, ANNE-MARIE. KERMARREC, F. LE FESSANT, L. MASSOULIÉ, S. PATARIN. *Peer Sharing Behaviour in the eDonkey Network, and Implications for the Design of Server-less File Sharing Systems*, submitted to NSDI'2005, Technical report, 2004.

Bibliography in notes

[20] M. ABADI, A. D. GORDON. *A Calculus for Cryptographic Protocols: The Spi Calculus*, in "Information and Computation", vol. 148, n^o 1, 10 January 1999, p. 1–70.

[21] R. M. AMADIO, I. CASTELLANI, D. SANGIORGI. *On Bisimulations for the Asynchronous π-Calculus*, in "Theoretical Computer Science", An extended abstract appeared in Proceedings of CONCUR '96, LNCS 1119: 147–162, vol. 195, n^o 2, 1998, p. 291–324.

[22] R. M. AMADIO, D. LUGIEZ. *On the reachability problem in cryptographic protocols*, in "Proceedings of CONCUR 00", Lecture Notes in Computer Science, INRIA Research Report 3915, march 2000, vol. 1877, Springer, 2000.

[23] M. BODIRSKY, J. NESETRIL. *Constraint Satisfaction with Countable Homogeneous Templates*, in "Computer Science Logic", M. BAAZ, J. MAKOWSKY (editors)., LNCS, vol. 2803, Springer, August 2003, p. 44–57.

[24] M. BOREALE. *On the expressiveness of internal mobility in name-passing calculi*, in "Theoretical Computer Science", A preliminary version of this paper appeared in the Proceedings of CONCUR'96, volume 1119 of

LNCS., vol. 195, n° 2, March 1998, p. 205–226.

[25] G. BOUDOL. *Asynchrony and the π-calculus (Note)*, Rapport de Recherche, n° 1702, INRIA, Sophia-Antipolis, 1992, http://www.inria.fr/rrrt/rr-1702.html.

[26] J. BOWEN, D. BAHLER. *Conditional Existence of Variables in Generalized Constraint Networks*, in "Proc. 9th. National Conference of the American Association for Artificial Intelligence", 1991, p. 215-220.

[27] N. BUSI, M. GABBRIELLI, G. ZAVATTARO. *Replication vs. recursive definition in Channel Based Calculi*, in "Proc. of ICALP 03", LNCS, Springer-Verlag, 2003.

[28] N. BUSI, M. GABBRIELLI, G. ZAVATTARO. *Comparing Recursion, Replication, and Iteration in Process Calculi*, in "Proc. of ICALP 04", LNCS, Springer-Verlag, 2004.

[29] N. BUSI, G. ZAVATTARO. *On the Expressive Power of Movement and Restriction in Pure Mobile Ambients*, in "Theoretical Computer Science", vol. 322(3), 2004, p. 477-515.

[30] I. CERVESATO, N. A. DURGIN, P. D. LINCOLN, J. C. MITCHELL, A. SCEDROV. *Relating Strands and Multiset Rewriting for Security Protocol Analysis*, in "13th IEEE Computer Security Foundations Workshop — CSFW'00, Cambrige, UK", P. SYVERSON (editor)., IEEE Computer Society Press, 3–5 July 2000, p. 35–51.

[31] I. CERVESATO, N. A. DURGIN, P. D. LINCOLN, J. C. MITCHELL, A. SCEDROV. *A Meta-Notation for Protocol Analysis*, in "Proceedings of the 12th IEEE Computer Security Foundations Workshop — CSFW'99, Mordano, Italy", R. GORRIERI (editor)., IEEE Computer Society Press, 28–30 June 1999, p. 55–69.

[32] H. COMON, V. CORTIER, J. MITCHELL. *Tree Automata with One Memory, Set Constraints, and Ping-Pong Protocols*, in "International Colloquium on Automata, Languages and Programming", Lecture Notes in Computer Science, vol. 2076, Springer-Verlag, 2001.

[33] K. COMPTON, S. DEXTER. *Proof Techniques for Cryptographic Protocols*, in "Proceedings of the 26th International Colloquium on Automata, Languages, and Programming", Lecture Notes in Computer Science, vol. 1644, Springer, 1999, p. 25–39.

[34] G. DELZANNO, A. PODELSKI. *Model checking in CLP*, in "Proc. of TACAS'99", LNCS, vol. 1579, Springer Verlag, 1999, p. 223–239.

[35] G. DENKER, J. MESEGUER, C. TALCOTT. *Protocol specification and analysis in Maude*, in "Proceedings of Workshop on Formal Methods and Security Protocols", 1998.

[36] S. EILENBERG. *Automata, Languages, and Machines*, Academic Press, 1974.

[37] B. FALTINGS, S. MACHO. *Open Constraint Satisfaction*, in "Proc. of Principles and Practice of Constraint Programming", LNCS (editor)., vol. 2470, Springer-Verlag, 2002, p. 356-370.

[38] L. FRIBOURG. *Constraint Logic Programming Applied to Model Checking*, in "Proc. of LOPSTR'99", LNCS, vol. 1817, Springer Verlag, 1999, p. 30–41.

[39] F. J. T. FÁBREGA, J. C. HERZOG, J. D. GUTTMAN. *Strand spaces: Why is a security protocol correct?*, in "Proceedings of the 1998 IEEE Symposium on Security and Privacy", IEEE Computer Society Press, may 1998, p. 160–171.

[40] J. GOUBAULT-LARRECQ. *A method for automatic cryptographic protocol verification*, in "Proceedings of the 15 IPDPS 2000 Workshops", Lecture Notes in Computer Science, vol. 1800, Springer-Verlag, may 2000, p. 977–984.

[41] K. HONDA, M. TOKORO. *An Object Calculus for Asynchronous Communication*, in "Proceedings of the European Conference on Object-Oriented Programming (ECOOP)", P. AMERICA (editor)., Lecture Notes in Computer Science, vol. 512, Springer-Verlag, 1991, p. 133–147.

[42] G. LOWE. *Casper: A Compiler for the Analysis of Security Protocols*, in "Proceedings of 10th IEEE Computer Security Foundations Workshop", Also in Journal of Computer Security, Volume 6, pages 53-84, 1998, 1997.

[43] J. MILLEN, G. DENKER. *CAPSL and MuCAPSL*, in "Journal of Telecommunications and Information Technology", 2002.

[44] D. MILLER. *Encryption as an Abstract Data-Type: An extended abstract*, in "Proceedings of FCS'03: Foundations of Computer Security", I. CERVESATO (editor)., 2003, p. 3-14.

[45] R. MILNER. *Communication and Concurrency*, International Series in Computer Science, Prentice Hall, 1989.

[46] R. MILNER, J. PARROW, D. WALKER. *A Calculus of Mobile Processes, I and II*, in "Information and Computation", A preliminary version appeared as Technical Reports ECF-LFCS-89-85 and -86, University of Edinburgh, 1989., vol. 100, n° 1, 1992, p. 1–40 & 41–77.

[47] R. MILNER, J. PARROW, D. WALKER. *Modal logics for mobile processes*, in "Theoretical Computer Science", vol. 114, n° 1, 1993, p. 149–171.

[48] M. MOHRI. *Edit-Distance Of Weighted Automata: General Definitions And Algorithms*, in "International Journal of Foundations of Computer Science", vol. 14, n° 6, 2003, p. 957-982.

[49] D. MONNIAUX. *Abstracting Cryptographic Protocols with Tree Automata*, in "Static Analysis Symposium", Lecture Notes in Computer Science, vol. 1694, 1999, p. 149–163.

[50] U. NESTMANN. *What Is a 'Good' Encoding of Guarded Choice?*, in "Proceedings of EXPRESS '97: Expressiveness in Concurrency (Santa Margherita Ligure, Italy, September 8–12, 1997)", C. PALAMIDESSI, J. PARROW (editors)., Electronic Notes in Theoretical Computer Science, Full version to appear in Information and Computation., vol. 7, Elsevier Science Publishers, 1997.

[51] U. NESTMANN, B. C. PIERCE. *Decoding Choice Encodings*, in "Proceedings of CONCUR '96: Concurrency Theory (7th International Conference, Pisa, Italy, August 1996)", U. MONTANARI, V. SASSONE (editors).,

Lecture Notes in Computer Science, Full version to appear in Information and Computation, vol. 1119, Springer-Verlag, 1996, p. 179–194.

[52] B. C. PIERCE, D. N. TURNER. *Pict: A Programming Language Based on the Pi-Calculus*, in "Proof, Language and Interaction: Essays in Honour of Robin Milner", G. PLOTKIN, C. STIRLING, M. TOFTE (editors)., The MIT Press, 1998, p. 455–494.

[53] A. PNUELI. *The temporal logic of programs*, in "Proc. of FOCS-77", IEEE Computer Society Press, IEEE, 1977, p. 46–57.

[54] M. O. RABIN, D. LEHMANN. *On the Advantages of Free Choice: A Symmetric and Fully Distributed Solution to the Dining Philosophers Problem*, in "A Classical Mind: Essays in Honour of C.A.R. Hoare", A. W. ROSCOE (editor)., An extended abstract appeared in the Proceedings of POPL'81, pages 133-138., chap. 20, Prentice Hall, 1994, p. 333–352.

[55] A. W. ROSCOE. *Modelling and Verifying Key-Exchange Protocols Using CSP and FDR*, in "Proceedings of the 8th IEEE Computer Security Foundations Workshop", IEEE Computer Soc Press, 1995, p. 98–107.

[56] S. SCHNEIDER. *Security properties and CSP*, in "Proceedings of the IEEE Symposium Security and Privacy", 1996.

[57] R. SEGALA, N. LYNCH. *Probabilistic simulations for probabilistic processes*, in "Nordic Journal of Computing", An extended abstract appeared in Proceedings of CONCUR '94, LNCS 836: 22–25, vol. 2, n° 2, 1995, p. 250–273.

[58] C. STIRLING. *Bisimulation, Model Checking and Other Games*, 1998, Notes for Mathfit Instructural Meeting on Games and Computation.

[59] R. J. VAN GLABBEEK, S. A. SMOLKA, B. STEFFEN. *Reactive, generative, and stratified models of probabilistic processes*, in "Information and Computation", vol. 121, n° 1, 1995, p. 59–80.