IN PARTNERSHIP WITH:
Université de Bologne (Italie)

Activity Report 2014

Project-Team FOCUS

Foundations of Component-based Ubiquitous Systems

IN COLLABORATION WITH: Dipartimento di Informatica - Scienza e Ingegneria (DISI), Universita' di Bologna

RESEARCH CENTER
Sophia Antipolis - Méditerranée

THEME
Distributed programming and software engineering

Table of contents

Project-Team FOCUS

Keywords: Semantics, Type Systems, Service Orchestration, Service Oriented Architecture, Ubiquitous Computing

Creation of the Project-Team: 2010 January 01.

1. Members

Research Scientist
Elena Giachino [University Bologna, Envisage project]

Faculty Members
Davide Sangiorgi [Team leader,University Bologna, Professor, HdR]
Mario Bravetti [University Bologna, Associate Professor]
Ugo Dal Lago [University Bologna, Associate Professor]
Maurizio Gabbrielli [University Bologna, Professor]
Ivan Lanese [University Bologna, Associate Professor]
Cosimo Laneve [University Bologna, Professor]
Simone Martini [University Bologna, Professor, HdR]
Gianluigi Zavattaro [University Bologna, Associate Professor]

Engineer
Jacopo Mauro [Univ. Paris VII, granted by ANR AEOLUS project]

PhD Students
Michele Alberti [Institut de Mathématiques de Luminy Marseille]
Roberto Amadini [University Bologna]
Alberto Cappai [University Bologna]
Ornela Dardha [University Bologna]
Saverio Giallorenzo [University Bologna]
Tudor Alexandru Lascu [University Bologna]
Jean-Marie Madiot [ENS Lyon]
Giulio Pellitta [University Bologna]
Marco Solieri [Université Paris 13]
Valeria Vignudelli [University Bologna]

Post-Doctoral Fellows
Martin Avanzini [University Bologna, from May 2014]
Alexis Bernadet [Inria, since Nov 2014]
Michael Lienhardt [University Bologna, Envisage project]

Administrative Assistant
Christine Riehl

Others
Raphaelle Crubille [ENS Lyon, stagiaire, from Sep 2014]
Claudio Guidi [External Collaborator, italianaSoftware s.r.l.]
Daniel Hirschkoff [External Collaborator, ENS Lyon]
Fabrizio Montesi [External Collaborator, IT University of Copenhagen]

2. Overall Objectives

2.1. Overall Objectives

Ubiquitous Computing refers to the situation in which computing facilities are embedded or integrated into everyday objects and activities. Networks are large-scale, including both hardware devices and software agents. The systems are highly mobile and dynamic: programs or devices may move and often execute in networks owned and operated by others; new devices or software pieces may be added; the operating environment or the software requirements may change. The systems are also heterogeneous and open: the pieces that form a system may be quite different from each other, built by different people or industries, even using different infrastructures or programming languages; the constituents of a system only have a partial knowledge of the overall system, and may only know, or be aware of, a subset of the entities that operate on the system.

A prominent recent phenomenon in Computer Science is the emerging of interaction and communication as key architectural and programming concepts. This is especially visible in ubiquitous systems. Complex distributed systems are being thought of and designed as structured composition of computational units, usually referred to as *components*. These components are supposed to interact with each other and such interactions are supposed to be orchestrated into conversations and dialogues. In the remainder, we will write *CBUS* for Component-Based Ubiquitous Systems.

In CBUS, the systems are complex. In the same way as for complex systems in other disciplines, such as physics, economics, biology, so in CBUS theories are needed that allow us to understand the systems, design or program them, analyse them.

Focus investigates the semantic foundations for CBUS. The foundations are intended as instrumental to formalizing and verifying important computational properties of the systems, as well as to proposing linguistic constructs for them. Prototypes are developed to test the implementability and usability of the models and the techniques. Throughout our work, ‘interaction’ and ’component’ are central concepts.

The members of the project have a solid experience in algebraic and logical models of computation, and related techniques, and this is the basis for our study of ubiquitous systems. The use of foundational models inevitably leads to opportunities for developing the foundational models themselves, with particular interest for issues of expressiveness and for the transplant of concepts or techniques from a model to another one.

3. Research Program

3.1. Models

The objective of Focus is to develop concepts, techniques, and possibly also tools, that may contribute to the analysis and synthesis of CBUS. Fundamental to these activities is *modeling*. Therefore designing, developing and studying computational models appropriate for CBUS is a central activity of the project. The models are used to formalize and verify important computational properties of the systems, as well as to propose new linguistic constructs.

The models we study are in the process calculi (e.g., the π-calculus) and λ-calculus tradition. Such models, with their emphasis on algebra, well address compositionality—a central property in our approach to problems. Accordingly, the techniques we employ are mainly operational techniques based on notions of behavioral equivalence, and techniques based on algebra, mathematical logics, and type theory.

The sections below provide some more details on why process calculi, λ-calculi, and related techniques, should be useful for CBUS.

4. Application Domains

4.1. Ubiquitous Systems

The main application domain for Focus are ubiquitous systems, broadly systems whose distinctive features are: mobility, high dynamicity, heterogeneity, variable availability (the availability of services offered by the constituent parts of a system may fluctuate, and similarly the guarantees offered by single components may not be the same all the time), open-endedness, complexity (the systems are made by a large number of components, with sophisticated architectural structures). In Focus we are particularly interested in the following aspects.

- *Linguistic primitives* for programming dialogues among components.
- *Contracts* expressing the functionalities offered by components.
- *Adaptability and evolvability* of the behaviour of components.
- *Verification* of properties of component systems.
- Bounds on component *resource consumption* (e.g., time and space consumed).

4.2. Service Oriented Computing and Cloud Computing

Today the component-based methodology often refers to Service Oriented Computing. This is a specialized form of component-based approach. According to W3C, a service-oriented architecture is "a set of components which can be invoked, and whose interface descriptions can be published and discovered". In the early days of Service Oriented Computing, the term services was strictly related to that of Web Services. Nowadays, it has a much broader meaning as exemplified by the XaaS (everything as a service) paradigm: based on modern virtualization technologies, Cloud computing offers the possibility to build sophisticated service systems on virtualized infrastructures accessible from everywhere and from any kind of computing device. Such infrastructures are usually examples of sophisticated service oriented architectures that, differently from traditional service systems, should also be capable to elastically adapt on demand to the user requests.

5. New Software and Platforms

5.1. Jolie

Members of Focus have developed Jolie [8] (Java Orchestration Language Interpreter Engine, see http://www.jolie-lang.org/). Jolie is a service-oriented programming language. Jolie can be used to program services that interact over the Internet using different communication protocols. Differently from other Web Services programming languages such as WS-BPEL, Jolie is based on a user-friendly C/Java-like syntax (more readable than the verbose XML syntax of WS-BPEL) and, moreover, the language is equipped with a formal operational semantics. This language is used for the *proof of concepts* developed around Focus activities. For instance, contract theories can be exploited for checking the conformance of a Jolie program with respect to a given contract. A spin-off, called "Italiana Software", has been launched around Jolie, its general aim is to transfer the expertise in formal methods for Web Services matured in the last few years onto Service Oriented Business Applications. The spin-off is a software producer and consulting company that offers service-oriented solutions (for instance, a "single sign-on" application) based on the Jolie language.

During 2014, the development of Jolie 1.1 has been completed (the release is due for the first half of January 2015). It is the result of about 600 commits, including more than 30 new standard library APIs, 100 bugfixes, and 100 improvements to the Jolie interpreter and libraries. Highlights:

- A new hierarchical semantics for handling sub-programs loaded in higher-order Jolie services.
- Support for "abstract locations". This enables the writing of extensions that automatically fetch the bindings to the external services needed by a Jolie program. We plan to use this feature to develop binding procedures that ensure correctness.
- Introduction of a tracer option for the Jolie interpreter, which displays the execution trace of a Jolie program (useful for debugging).
- Substantial improvements to memory management of higher-order programs.
- Improved integration with web applications, which supports new techniques for handling the evolution of legacy web applications using the composition primitives of Jolie.

Moreover Jolie Enterprise has been released: this is an administrative tool that allows one to deploy Jolie services on remote nodes. Jolie Enterprise is able to manage services that run on different nodes on different machines, tracking all messages exchanged between services and viewing the log on GUI so that one can have a report of what happened in the system. Currently there are about 15 installations of Jolie Enterprise at SME in clothing, construction and manufacturing.

5.2. Others

Below we list some software that has been developed, or is under development, in Focus.

- *Deadlock analysis* (http://df4abs.nws.cs.unibo.it/).

  We have prototyped a framework for statically detecting deadlocks in a concurrent object-oriented language with asynchronous method calls and cooperative scheduling of method activations (the language is inspired by the ABS language developed in the EU project HATS and currently extended with primitives for cloud-computing in the EU project ENVISAGE). Since this language features recursion and dynamic resource creation, deadlock detection is extremely complex and state-of-the-art solutions either give imprecise answers or do not scale. In order to augment precision and scalability we propose a modular framework that allows several techniques to be combined. The basic component of the framework is a front-end inference algorithm that extracts abstract behavioural descriptions of methods, called contracts, which retain resource dependency information. Then these contracts are analysed by a back-end that uses a fix-point technique to derive in a deterministic way the deadlock information.

- *CaReDeb* (http://www.cs.unibo.it/caredeb).

  Reversible debugging provides developers with a way to execute their applications both forward and backward, seeking the cause of an unexpected or undesired event. We have developed CaReDeb, the first prototype of a causal-consistent reversible debugger. Causal consistent here means that independent actions are undone independently, while dependent actions are undone in reverse order. This allows the programmer to concentrate on the threads responsible of the bug, independently of the actual interleaving. CaReDeb provides primitives that given a misbehaviour, e.g., a variable has not the expected value, allow one to go back to the action responsible for it, e.g., the one that assigned the wrong value to the variable. Notably, the programmer has no need to know which thread the action belongs to, since this is found automatically by the debugger. The procedure can be iterated till the bug is found. CaReDeb targets a fragment of the language Oz, which is at the basis of Mozart. The considered fragment provides functional variables, procedures, threads, and asynchronous communication via ports.

- *AIOCJ* (http://www.cs.unibo.it/projects/jolie/aiocj.html).

AIOCJ is a framework for programming adaptive distributed systems based on message passing. AIOCJ comes as a plugin for Eclipse, AIOCJ-ecl, allowing to edit descriptions of distributed systems as adaptive interaction-oriented choreographies (AIOC). From interaction-oriented choreographies the description of single participants can be automatically derived. Adaptation is specified by rules allowing to replace predetermined parts of the AIOC with a new behaviour. A suitable protocol ensures that all the participants are updated in a coordinated way. As a result, the distributed system follows the specification given by the AIOC under all changing sets of adaptation rules and environment conditions. In particular, the system is always deadlock-free. AIOCJ can interact with external services, seen as functions, by specifying their URL and the protocol they support (HTTP, SOAP, ...). Deadlock-freedom guarantees of the application are preserved provided that those services do not block.

- *METIS* (https://github.com/aeolus-project/metis)

  As partners of the Aeolus project we have developed a tool for the automatic synthesis of deployment plans. A deployment plan is a sequence of actions that, when performed, allows the deployment of a given configuration of components. METIS (Modern Engineered Tool for Installing Software systems) is a tool that enables one to automatically generate a deployment plan, starting from a description of the configuration following the Aeolus model. The software is open source. It is written entirely in OCaml and is about 3.5K lines of source code. The tool is based on theoretical results that guarantee its soundness and completeness, while maintaining polynomial computational complexity. METIS already showed its effectiveness in practice by handling synthesized problem instances with hundreds of components in less than a minute. We are currently validating Metis in a production environment by integrating it in Armonic, an infrastructure for cloud application deployment in OpenStack cloud systems developed by the Mandriva company.

- *SUNNY-CP* (https://github.com/jacopoMauro/sunny-cp)

  Within the Constraint Programming (CP) paradigm, a portfolio approach enables to combine a number of different constraint solvers in order to create a globally better solver, dubbed a portfolio solver. After several empirical evaluations (e.g., [22], [23], [24]) we have decided to develop *SUNNY-CP*, a portfolio solver for solving both Constraint Satisfaction Problems and Constraint Optimization Problems. The goal of *SUNNY-CP* is to provide a flexible, configurable, and usable CP portfolio solver that can be set up and executed just like a regular individual CP solver. To the best of our knowledge, *SUNNY-CP* is the only sequential portfolio solver able to solve generic CP problems, and it was the only portfolio solver that attended the MiniZinc Challenge 2014 (i.e., the only active international competition to evaluate the performance of CP solvers). (*SUNNY-CP* performed very well, ranking 4th in the competition and receiving an honourable mention by the challenge organizers.) The application of *SUNNY-CP* in the optimization problems defined within the Aeolus project have lead to time improvements beyond an order of magnitude. *SUNNY-CP* is mainly written in Python, and we are currently enhancing the tool in order to make it more usable, flexible, and parallel (i.e., able to properly exploit multiple cores).

The sofware below have not undergone substantial modifications during 2014.

- *Croll-pi Interpreter* (http://proton.inrialpes.fr/~mlienhar/croll-pi/implem/). Croll-pi is a concurrent reversible language featuring a rollback operator to undo a past action (together with all the actions depending on it), and a compensation mechanism to avoid cycling by redoing the same action again and again. We have developed an interpreter for croll-pi using Maude.
- *IntML* is a functional programming language guaranteeing sublinear space bounds for all programs [51]. See the Activity Reports of previous years (in particular 2010) for more details.
- *Lideal* (http://lideal.cs.unibo.it/) is an experimental tool implementing type inference for dependently linear type systems. The tool reduces the problem of evaluating the complexity of PCF (i.e. functional programs with primitive integers and recursive definitions) to checking a set of first-order inequalities for validity. The latter can then be handled through SMT solvers or put in a form suitable for managing them with tools such as CoQ. See the Activity Reports of previous years (in particular

2010) for more details.

6. New Results

6.1. Highlights of the Year

Valeria Vignudelli has received the AILA (Associazione Italiana di Logica e sue Applicazioni) award for her 2014 master thesis.

6.2. Service-oriented computing

Participants: Maurizio Gabbrielli, Elena Giachino, Saverio Giallorenzo, Claudio Guidi, Mario Bravetti, Ivan Lanese, Michael Lienhardt, Jacopo Mauro, Fabrizio Montesi, Gianluigi Zavattaro.

6.2.1. Orchestrations

Orchestration models and languages in the context of Service-Oriented Architectures (SOA) are used to describe the composition of services focusing on their interactions. Coloured Petri nets (CPN) offer a formal yet easy tool for modelling interactions in SOAs, however mapping abstract SOAs into executable ones requires a non-trivial and time-costly analysis. In [34], we propose a methodology that maps CPN-modelled SOAs into Jolie SOAs (our target language), exploiting a collection of recurring control-flow patterns, called Workflow Patterns, as composable blocks of the translation. We validate our approach with a realistic use case. In addition, we pragmatically assess the expressiveness of Jolie with respect to the considered WPs.

6.2.2. Choreographies

Choreographies are high-level descriptions of distributed interacting systems featuring as basic unit a communication between two participants. A main feature of choreographies is that they ensure deadlock-freedom by construction. From a choreography one can automatically derive a description of the behaviour of each participant using a notion of projection. Choreographies can be used both at the level of types (multiparty session types) or as a programming language. In [18] we surveyed the work on choreographies and behavioural contracts in multiparty interactions. In [28] we explored the notion of deadlock freedom (the system never gets stuck), and the related notions of lock freedom (each action is eventually executed under a fair scheduling) and progress (each session never gets stuck). Previous work studied how to define progress in an open setting by introducing the notion of catalysers, execution contexts generated from the type of a process. We refined the notion of catalysers leading to a novel characterization of progress in terms of the standard notion of lock-freedom. We applied our results both to binary session types and in an untyped session-based setting. We combined our results with existing techniques for lock-freedom, obtaining a new methodology for proving progress. Our methodology captures new processes w.r.t. previous progress analysis based on session types. The two following works consider the extension of choreographies, which traditionally have a static structure, to deal with adaptation, i.e., dynamic changes of the structure of choreographies. A preliminary analysis of adaptable choreographies at the level of types is presented in [27]. This work considers both updates from inside the system (self-adaptation), and external updates. Adaptable choreographies as a programming language are considered in [33], where we presented AIOCJ, a framework for programming distributed adaptive applications. AIOCJ allows the programmer to specify which parts of the application can be adapted. Adaptation takes place at run-time by means of rules, which can change during the execution to tackle possibly unforeseen adaptation needs. AIOCJ relies on a solid theory that ensures applications to be deadlock free by construction also after adaptation.

6.3. Models for reliability

Participants: Mario Bravetti, Elena Giachino, Ivan Lanese, Michael Lienhardt, Gianluigi Zavattaro.

6.3.1. Reversibility

We have continued the study of causal-consistent reversibility started in the past years. In [17] we presented an overview of causal-consistent reversibility, summarizing the main approaches in the literature, and the related results and applications. An interesting application is debugging. Reversible debugging provides developers with a way to execute their applications both forward and backward, seeking the cause of a misbehaviour. In a concurrent setting, reversing actions in the exact reverse order they have been executed may lead to undo many actions that were not related to the bug under analysis. On the other hand, undoing actions in some order that violates causal dependencies may lead to states that could not be reached in a forward execution. In [36] we proposed a new approach, where each action can be reversed if all its consequences have already been reversed. The main feature of the approach is that it allows the programmer to easily individuate and undo exactly the actions that caused a given misbehaviour till the corresponding bug is reached. We discussed the appropriate primitives for causal-consistent reversible debugging and presented their prototype implementation in the CaReDeb tool.

6.3.2. Fault models

We have continued the study of primitives for fault handling in a concurrent setting. In [19] we critically discussed the different choices that have to be made when defining a fault model for a concurrent object-oriented programming language. We consider in particular the ABS language, and analyse the interplay between the fault model and the main features of ABS, namely the cooperative concurrency model, based on asynchronous method invocations whose return results via futures, and its emphasis on static analysis based on invariants.

6.4. Cloud Computing

Participants: Roberto Amadini, Maurizio Gabbrielli, Elena Giachino, Saverio Giallorenzo, Claudio Guidi, Cosimo Laneve, Michael Lienhardt, Tudor Alexandru Lascu, Jacopo Mauro, Gianluigi Zavattaro.

6.4.1. Cloud application deployment

Configuration and management of applications in the cloud is a complex task that requires novel methodologies and tools. In [16] we have performed a foundational study of the complexity boundaries for the automatic deployment problem, showing that in the general case this problem is undecidable, it is decidable but non-primitive recursive if capacity constraints are not taken into account, while it turns out to be polynomial time if also conflicts between software components are not considered. Starting from these foundational observations, we have investigated the exploitability in this specific context of state-of-the-art constraint optimization techniques, a well established approach for the modeling and solution of complex optimization problems. In particular, in [23], [24] we have studied how the "portfolio technique" approach can be applied to optimization problems, combining and exploiting the performances of existing solvers to get a global, more robust and fast solver. Encouraged by these results, we have developed SUNNY-CP [13], [22]: a porftolio constraint solver for constraint satisfaction and optimization problems. SUNNY-CP has proven to have remarkable performances, ranking 4th in the annual MiniZinc challenge (i.e., the international competition to evaluate the performances of constraint solvers) and receiving a ‘honorable’ mention by the challenge organizers.

6.4.2. Cloud resource management

The management of cloud resources from client programs requires the definition of Application Programming Interfaces (APIs) that expose specific functionalities to external invokers. Programs can be built that compose existing APIs in order to obtain new functionalities. However API composition easily becomes a frustrating and time-costly task that hinders API reuse. The issue derives from technology-dependent features of API composition such as the need of extensive documentation, protocol integration, security issues, etc.. In [39] we introduce the perspective of the API-as-a-Service (APIaaS) layer as tool to ease the development and deployment of applications based on API composition, abstracting communication protocols and message formats. We elicit the desirable features of such a layer and provide a proof-of-concept prototype implemented using a service-oriented language.

Another critical aspect in this context deals with the problem of dynamic reallocation of resources. In [38] we study a type-based technique for modeling and analysis of systems in which concurrent object-oriented programs dynamically create and move resources. The type of a program is behavioural, namely it expresses the resource deployments over periods of (logical) time. Our technique admits the inference of types and may underlie the optimisation of the costs and consumption of resources.

6.5. Resource Control

Participants: Michele Alberti, Alberto Cappai, Ugo Dal Lago, Simone Martini, Giulio Pellitta, Davide Sangiorgi, Marco Solieri, Valeria Vignudelli.

6.5.1. Probabilistic higher-order calculi

The first results of our efforts on probabilistic higher-order systems and languages have started to appear in 2014. In particular, we have focused our attention on the impact of probability to the classical notion of context equivalence for the lambda-calculus, showing that applicative bisimilarity continues to be a congruence [31], and that it even coincides with context equivalence when evaluation is done in the call-by-value order [29]. The expressive power of higher-order concurrent contexts has been compared to the expressive power of lambda-calculi contexts and put in relation with other equivalences when the observed process is either an ordinary Labelled Transition Systems (LTS) or a reactive probabilistic transition system [25]. The obtained spectrum of equivalences for reactive probabilistic processes has been shown to be finer than the one for classic LTSs. We have also analysed the expressive power of different first-order testing equivalences (with nondeterministic tests, probabilistic tests, and both nondeterministic and probabilistic tests) in the spectrum for reactive probabilistic processes [26].

6.5.2. Resource consumption

The main result about resource consumption has been about an open problem on the λ-calculus: we proved that the number of leftmost-outermost steps to normal form is indeed an invariant cost model in the sense of Slot and van Emde Boas' weak invariance thesis [21]. We also introduced a new recursion theoretic framework for probabilistic computation in which one is able to capture probabilistic polynomial time through Leivant's Tiering [32].

6.5.3. Geometry of interaction

Novel results have been obtained for Geometry of Interaction (GoI), itself a semantics framework for linear logic introduced by Jean-Yves Girard thirty years ago. In particular, we have shown how the most concrete presentations of GoI, namely so-called token machines, can go *parallel*, thus exploiting the potential parallelism in functional programs (through the Curry-Howard Correspondence). This has been made concrete by studying extensions of multiplicative linear logic in which synchronization becomes an operator where tokens can indeed synchronize [30]. This has been later shown to be necessary to model quantum computation [44]. A simple, minimalistic GoI model of the resource λ-calculus has also been introduced [43].

6.6. Verification techniques for extensional properties

Participants: Daniel Hirschkoff, Elena Giachino, Michael Lienhardt, Cosimo Laneve, Jean-Marie Madiot, Davide Sangiorgi.

Extensional refers to properties that have to do with behavioural descriptions of a system (i.e., how a system looks like from the outside). Examples of such properties include classical functional correctness and deadlock freedom. Related to techniques for extensional properties are the issues of decidability (the problem of establishing whether certain properties are computationally feasable).

6.6.1. Coinductive techniques

Coinductive techniques, notably those based on bisimulation, are widely used in concurrency theory to reason about systems of processes. The bisimulation proof method can be enhanced by employing 'bisimulations up-to' techniques. A comprehensive theory of such enhancements has been developed for first-order (i.e., CCS-like) LTSs and bisimilarity, based on the notion of compatible function for fixed-point theory. We have transported this theory onto languages whose bisimilarity and LTS go beyond those of first-order models [40]. The approach consists in exhibiting fully abstract translations of the more sophisticated LTSs and bisimilarities onto the first-order ones. This allows us to reuse directly the large corpus of up-to techniques that are available on first-order LTSs. We have investigated the method on the π-calculus, the Higher-Order π-calculus, and a (call-by-value) λ-calculus with references.

In [20], mostly a tutorial paper, a few forms of bisimulation and of coinductive techniques that have been proposed for higher-order languages are discussed, beginning with the pure lambda-calculus and then moving to extensions of it, notably those with non-determinism and probabilities.

6.6.2. Deadlock detection

Deadlock detection in concurrent programs that create networks with an arbitrary number of nodes is extremely complex and solutions either give imprecise answers or do not scale. To enable the analysis of such programs, we have studied an algorithm for detecting deadlocks [37], [35], in a basic model featuring recursion and fresh name generation, called Lam. We then have designed a type system that associates Lams to processes. As a byproduct of these two techniques, we have an algorithm that is more powerful than previous ones and that can be easily integrated into the current release of TyPiCal, a type-based analyser for π-calculus.

6.6.3. Expressiveness and decidability in actor-like systems

Refining work in previous years, we have studied [15] the expressive power of an actor-like language, featuring concurrent objects and asynchronous message-passing. We have identified the presence/absence of fields as a crucial feature: the dynamic creation of names in combination with fields gives rise to Turing completeness. On the other hand, restricting to stateless actors gives rise to systems for which properties such as termination are decidable. This decidability result still holds for actors with states when the number of actors is bounded and the state is read-only.

7. Partnerships and Cooperations

7.1. National Initiatives

- AEOLUS (Mastering the Cloud Complexity) is an ANR-ARPEGE project started on December 2010 that finished on December 2014. AEOLUS studies the problem of installation, maintenance and update of package-based software distributions in cloud-based distributed systems. The problem consists of representing the dependencies of packages and the inter-relationships among the services, in such a way that starting from a declarative description of the application to be deployed on the cloud, it is possible to automatically compute the resources (ie. virtual machines) to be acquired, and the allocation of such resources to the software services needed to run the application. Main persons involved: Gabbrielli, Lienhardt, Mauro, Zavattaro.
- REVER (Programming Reversible Recoverable Systems) is an ANR project that started on 1st December 2011 and with a 48-month duration. REVER aims to study the possibility of defining semantically well-founded and composable abstractions for dependable computing on the basis of a reversible programming language substrate, where reversibility means the ability to undo any distributed program execution, possibly step by step. The critical assumption behind REVER is that by adopting a reversible model of computation, and by combining it with appropriate notions of compensation and modularity, one can develop systematic and composable abstractions for recoverable and dependable systems. Main persons involved: Giachino, Lienhardt, Lanese, Laneve, Zavattaro.

- PACE (Processus non-standard: Analyse, Coinduction, et Expressivité) is an ANR project that started in 2013. The project targets three fundamental ingredients in theories of concurrent processes, namely coinduction, expressiveness, and analysis techniques. The project aims at processes that are beyond the realm of "traditional" processes. Specifically, the models studied exhibit one or more of the following features: probabilities, higher-order, quantum, constraints, knowledge, and confidentiality. These models are becoming increasingly more important for today's applications. Coinduction is intended to play a pivotal role. Indeed, the approaches to expressiveness and the analysis techniques considered in the project are based on coinductive equalities. Main persons involved: Hirschkoff (project coordinator), Dal Lago, Lanese, Sangiorgi, Zavattaro.
- ELICA (Expanding Logical Ideas for Complexity Analysis) is an ANR project which started on October 2014 and that we will finish on September 2018. ELICA is a project about methodologies for the static analysis of programs as for their resource consumption. The project's aim is to further improve on logical methodologies for complexity analysis (type systems, rewriting, etc.). More specifically, one would like to have more powerful techniques with less false negatives, being able at the same time to deal with nonstandard programming paradigms (concurrent, probabilistic, etc.). Main persons involved: Avanzini, Cappai, Dal Lago, Hirschkoff, Martini, Sangiorgi.

7.2. European Initiatives

7.2.1. FP7 & H2020 Projects

- ENVISAGE (Engineering Virtualized Services) is a EU FP7 project, with starting date October 1st, 2013, and with a 3-year duration. The project is about model-based development of virtualized services, including tool support for resource analysis. Most Focus members are involved.

7.2.2. Collaborations in European Programs, except FP7

- The ICT COST Action BETTY (Behavioural Types for Reliable Large-Scale Software Systems), initiated in October 2012 and with a four-year duration, uses behavioural type theory as the basis for new foundations, programming languages, and software development methods for communication-intensive distributed systems. Behavioural type theory encompasses concepts such as interfaces, communication protocols, contracts, and choreography. Main persons involved: Bravetti, Giachino, Hirschkoff, Lanese, Laneve, Mauro, Sangiorgi, Zavattaro.

7.2.3. Collaborations with Major European Organizations

Simone Martini is a member of the Executive Board of EQANIE (European Quality Assurance Network for Informatics Education), from October 2014.

We list here the cooperations and contacts with other groups, without repeating those already listed in previous sections.

- ENS Lyon (on concurrency models and resource control). Contact person(s) in Focus: Dal Lago, Martini, Sangiorgi, Vignudelli. Some visit exchanges during the year, in both directions. One joint PhD supervision (J.-M. Madiot).
- Inria EPI Spades (on models and languages for components, reversibility). Contact person(s) in Focus: Lanese. Some visit exchanges during the year, in both directions.
- Laboratoire d'Informatique, Université Paris Nord, Villetaneuse (on implicit computational complexity). Contact person(s) in Focus: Dal Lago, Martini. An Italian PhD student (Marco Solieri) is working on his PhD thesis with joint supervision (Martini, Guerrini).
- Institut de Mathématiques de Luminy, Marseille (on lambda-calculi, linear logic and semantics). Contact person(s) in Focus: Dal Lago, Martini. One joint PhD supervision (Michele Alberti).
- Team PPS, University of Paris-Diderot Paris 7 (on logics for processes, resource control). Contact person(s) in Focus: Dal Lago, Martini, Sangiorgi. Some short visits in both directions during the year.

- IRILL Lab, Paris (on models for the representation of dependencies in distributed package based software distributions). Contact person(s) in Focus: Mauro, Zavattaro. Some short visits in both directions during the year.
- EPI Carte, Inria-Nancy Grand Est and LORIA (on implicit computational complexity). Contact person(s) in Focus: Dal Lago.
- LMU Munich (M. Hofmann) (on Implicit computational complexity and IntML). Contact person(s) in Focus: Dal Lago.
- IMDEA Software, Madrid (G. Barthe) (on Implicit computational complexity for cryptography). Contact person(s) in Focus: Dal Lago,Sangiorgi. Some visits during 2014.
- Facultad de Informatica, Universidad Complutense de Madrid (on web services). Contact person(s) in Focus: Bravetti. Bravetti is an external collaborator in the project "ESTuDIo: ESpecificacion y Testing de sistemas altamente DIstribuidos" (Specification and Testing of Highly Distributed Systems) January 1, 2013 - December 31, 2015 (3 years), funded by the Spanish Ministerio de Economia y Competitividad.

7.3. International Research Visitors

7.3.1. Visits of International Scientists

The following researchers have visited Focus for short periods; we list them together with the title of the talk they have given during their stay, or the topic discussed during their stay.

- Luca Padovani (Turin) "Deadlock and lock freedom in the linear pi-calculus".
- Jean-Bernard Stefani (Inria Grenoble), "Strong isolation in actor systems".
- Mauro Caporuscio (Milan): "Prime: A middleware support for fluid distributed systems".
- Fabrizio Montesi (Copenhagen): "Choreographic Programming".
- Marco Carbone (Copenhagen): "Behavioural types for adaptable service composition".
- Sandro Etalle (Eindhoven), "Signature-Less Network Intrusion Detection: from the research table to the production environments."
- Wolf Zimmermann (Halle, Germany), "Automatic Protocol Conformance Checking in Component-Based and Service-Oriented Systems."
- Lars Kotthoff (Cork, Ireland), "Towards an algorithm selection standard: data format and tools."
- Herbert Wiklicky (London), "Quantitative Aspects in Program Synthesis."
- Naoki Kobayashi (Tokyo), "Model checking higher-order programs".
- Benoit Valiron and Claudia Faggian (Paris), "Geometry of Synchronization"
- Marc Bagnol (Marseille), "On the Resolution Semiring"
- Irek Ulidowski (Leicester), on the topic of reversibility.

7.3.1.1. Internships

Raphaelle Crubille, from ENS Lyon, has begun a stage in Focus during 2014, under the supervision of Ugo Dal Lago.

8. Dissemination

8.1. Promoting Scientific Activities

M. Bravetti: Program Committee member of the following conferences. BigData 2014 (IEEE International Conference on Big Data), WWV 2014 (10th International Workshop on Automated Specification and Verification of Web Systems), WS-FMDS 2014 (4th WorkShop on Formal Methods in the Development of Software), SEFM 2014 (12th International Conference on Software Engineering and Formal Methods), WS-FM 2014 (11th Workshop on Web Services and Formal Methods), ESOCC 2014 (3rd European Conference on Service-Oriented and Cloud Computing).

Invited speaker at MOD* 2014 (1st Int. Workshop on Logics and MODel-checking for self-* systems).

Organiser of Int. Workshop on Contracts for Efficient and Reliable Services.

U. Dal Lago: Program Committee co-chair and main orgainser for FOPARA 2013: Third International Workshop on Foundational and Practical Aspects of Resource Analysis (August 29th to 31st, 2013, Bertinoro, Italy).

Program Committee member of the conferences RTA-TLCA 2014, LOLA 2014 and IFIP-TCS 2014.

PC chair for DCM 2014.

Invited speaker at the "Mathematical Structures of Computation" event in Lyon.

Lecturer at the school BISS 2014, Bertinoro, Italy, March 2014.

D. Hirschkoff: Co-organisee of the international event on 'Concurrency, Logic and Types', as part of the 'Mathematical Structures of Computation', in february 2014 in Lyon (see http://smc2014.univ-lyon1.fr/doku.php?id=week5). Several Focus members took part in the event.

E. Giachino: Program Committee member of the conference iFM 2014 and the workshops BEAT 2014 and ICE 2014.

Workshop chair for the iFM and FACS 2014.

I. Lanese: Program Committee member of the following workshops and conferences: EXPRESS/SOS 2014 (Combined 21st International Workshop on Expressiveness in Concurrency and 11th Workshop on Structured Operational Semantics), WS-FM:FASOCC 2014 (11th International Workshop on Web Services and Formal Methods: Formal Aspects of Service-Oriented and Cloud Computing), IFIP TCS 2014 (Theoretical Computer Science 2014), SAC 2014 (Service Oriented Architectures and Programming track of the 29th Annual ACM Symposium on Applied Computing).

Steering committee for the Service Oriented Architectures and Programming (SOAP) track of the Annual ACM Symposium on Applied Computing (SAC), the International Symposium on Formal Aspects of Component Software (FACS).

Co-chair of the conferences 11th International Symposium on Formal Aspects of Component Software (FACS 2014), and 7th Interaction and Concurrency Experience (ICE 2014).

Member of the editorial board of the Scientific World Journal, the Open Journal of Communications and Software, and the journal of Scientific Programming.

C. Laneve: Tutorialist at the PhD School: SFM-14:ESM (14th International School on Formal Methods for the Design of Computer, Communication and Software Systems: Executable Software Models)

M.Gabbrielli: Member of the editorial board of the Int'l Journal Theory and Practice of Logic Programming.

S. Martini: Program Committee member of International Workshop Linearity 2014. Invited speaker at the Symposium on History and Philosophy of Computing (Annual Meeting of the International Association for Computing and Philosophy), Thessaloniki, Greece.

F. Montesi: Invited speaker at WS-FM: FASOCC (11th International Workshop on Web Services and Formal Methods: Formal Aspects of Service-Oriented and Cloud Computing)

Track program chair for 28th ACM Symposium on Applied Computing (SAC) track on Service-Oriented Applications and Programming track (SOAP).

D. Sangiorgi: Member of the editorial board of the journals: Logical Methods in Computer Science, Acta Informatica, Distributed Computing, Foundations and Trends in Programming Languages, and RAIRO Theoretical Informatics and Applications.

Invited speaker at the workshops "Mathematical Structures of Computation" (Lyon, February 2014), CMCS'14 (12th International Workshop on Coalgebraic Methods in Computer Science), and "Open Problems in Concurrency Theory" (Bertinoro, Italy, June 2014).

Program Committee member of the conferences FOSSACS'14 (17th International Conference on Foundations of Software Science and Computation Structures) and LATA'14 (8th International Conference on Language and Automata Theory and Applications).

V. Vignudelli: Invited talks at the workshop "Mathematical Structures of Computation" (Lyon, February 2014), at the workshop "Open Problems in Concurrency Theory", Bertinoro, Italy, June 2014, and at the seminar series "CHoCoLa", at ENS Lyon, September 2014.

G. Zavattaro: Program Committee member of the following conferences and workshops: 11th International Workshop on Web Services and Formal Methods: Formal Aspects of Service-Oriented and Cloud Computing (WS-FM:FASOCC'14); 2014 European Conference on Service-Oriented and Cloud Computing (ESOCC'14); 2nd Workshop on Dependability and Interoperability in Heterogeneous Clouds (DIHC14), Porto, Portugal, August 25-29, 2014; 2nd IEEE International Workshop on Formal Methods Integration (FMi 2014); 7th IEEE International Conference on Cloud Computing (CLOUD 2014); 16th International Conference on Coordination Models and Languages (COORDINATION'14).

General chair of the joint conferences 11th International Symposium on Formal Aspects of Component Software (FACS 2014) and 11th International Conference on Integrated Formal Methods (iFM 2014, Bertinoro, Italy, 9–12 September 2014).

Organiser of iFM and FACS 2014 (Bertinoro, Italy, 9–12 September 2014).

8.2. Teaching - Supervision - Juries

8.2.1. Teaching

Below are the teaching activities of the Focus members during 2014.

- Mario Bravetti

  Undergraduate: "Tecnologie Web", 66 hours, 3rd year, University of Bologna, Italy.

  Master: "Applicazioni e Servizi Web", 54 hours, 2nd year, University of Bologna, Italy.

- Ugo Dal Lago

  Undergraduate: "Introduction to Programming in Python", 20 hours, 1st year, University of Bologna, Italy.

  Undergraduate: "Optimization", 36 hours, 2nd year, University of Bologna, Italy.

- Maurizio Gabbrielli

  Undergraduate: "Programming languages", 40 hours, 2nd year, University of Bologna, Italy.

  Master: "Artificial Intelligence", 60 hours, 2nd year, University of Bologna, Italy.

- Elena Giachino

  Undergraduate: "Programmazione", 40 hours, 1st year, University of Bologna, Italy.

- Ivan Lanese

  Undergraduate: "Programmazione", 32 hours, 1st year, University of Bologna, Italy.

  Master: "Business Process Management", 22 hours, 2nd year, University of Bologna, Italy.

- Cosimo Laneve

  Undergraduate: "Programmazione", 70 hours, 1st year, University of Bologna, Italy.

  Master: "Analisi di Programmi", 42 hours, 1st year, University of Bologna, Italy.

- Simone Martini

  Undergraduate: "Introduction to programming in Python", 58 hours, 1st year, University of Bologna, Italy.

Undergraduate: “Computer abilities for biologists”, 8 hours, 1st year, University of Bologna, Italy.
Master: “Logical Foundations of Computer Science”, 48 hours, 2nd year, University of Bologna, Italy.
PhD: “Proof theory for computer scientists”, 20 hours, University of Bologna, Italy.

- Jacopo Mauro
  Undergraduate: “Algoritmi e strutture dati”, 20 hours, 1st year, University of Bologna, Italy.
- Giulio Pellitta
  Undergraduate: “Laboratorio di Operating Systems”, 36 hours, 2nd year, University of Bologna, Italy.
  Undergraduate: “Attivita’ di supporto alla didattica universitaria di Informatica nell’ambito della Piattaforma di e-learning A3”, 140 hours, 1st year, University of Bologna, Italy.
- Davide Sangiorgi
  Undergraduate: “Operating Systems”, 110 hours, 2nd year, University of Bologna, Italy.
  Master: “Models for concurrency”, 15 hours, 2nd year, University of Bologna, Italy.
- Marco Solieri
  Master: “Administration Système”, 29 hours, 3rd year, école d’ingénieurs, Institut Galilé, Université Paris 13, France
  Master: “Intranet, Internet et Web”, 35 hours, 2nd year, école d’ingénieurs, Institut Galilé, Université Paris 13, France
- Gianluigi Zavattaro
  Master: “Linguaggi di Programmazione e Modelli Computazionali”, 120 hours, 1st year, University of Bologna, Italy.

8.2.2. Supervision

PhD thesis completed in 2014:

- Michele Alberti, “On operational properties of quantitative extensions of the λ-calculus”. December 2014. Joint degree: Docteur en Mathematiques, Aix-Marseille Université; Dottorato in Informatica, University of Bologna. Supervisors: L. Regnier, S. Martini.
- Ornela Dardha, “Type Systems for Distributed Programs: Components and Sessions”, May 2014. University of Bologna. Supervisor: D. Sangiorgi.
- Tudor Alexandru Lascu, “Automatic deployment of applications in the cloud”, May 2014. University of Bologna. Supervisor: G. Zavattaro.
- Giulio Pellitta, “Extending Implicit Computational Complexity and Abstract Machines to Languages with Control”, May 2014. University of Bologna. Supervisors: U. Dal Lago and S. Martini.

Below are the details on the PhD students in Focus: starting date, topic or provisional title of the thesis, supervisor(s). These are all PhDs in progress.

- Roberto Amadini, January 2012, Constraint Programming, M. Gabbrielli.
- Ferdinanda Camporesi, January 2009, Analysis of system biology, R. Cousot (Ecole Normale, Paris) and M. Gabbrielli
- Alberto Cappai, January 2013; Security, ICC and probabilistic bisimulation, U. Dal Lago.
- Abel Garcia, January 2014, Analysis of Cloud Computing Systems, C. Laneve.
- Saverio Giallorenzo, January 2013, Safe-adaptive services and choreography languages, M. Gabbrielli.
- Jean-Marie Madiot, September 2011, Types, proofs and proof techniques for process calculi, D. Hirschkoff and D. Sangiorgi.
- Alessandro Rioli, January 2012; Bisimulation equivalence in quantum lambda calculi, U. Dal Lago.
- Marco Solieri, December 2011; Géometrie de l’interaction, réseaux, et lambda-calcul, S. Guerrini (Paris Nord) and S. Martini.
- Valeria Vignudelli, January 2014, Probabilities in Higher-Order languages, D. Sangiorgi.

8.2.3. Juries

U. Dal Lago has been rapporteur and member of the PhD evaluation committee for Marc Bagnol (University Aix-Marseille) and Michele Alberti (University Aix-Marseille).

D. Hirschkoff has been rapporteur for the PhD defense of J.B. Midez (December 2014, Université Aix Marseille).

G. Zavattaro has been member of the 2014 PhD evaluation committee at the department of Computer Science of the University of Genova, Italy.

8.3. Popularization

Simone Martini has carried out extended work of scientific divulgation:

- member of the technical committee of Olimpiadi del Problem Solving (at Italian Ministry of Education), http://www.olimpiadiproblemsolving.com;
- member of the technical committee of two national projects of Italian Ministry of Education, founded by ESF - European Social Fund: “Robotica educativa”, and “Informatica nei licei”;
- various talks at institutes and workshops on the teaching methods for Computer Science.
- coordinator of some initiatives for the ‘Hour of Code’, see italia.code.org and programmailfuturo.it.

D. Hirschkoff takes part in several popularization activities in schools, in Lyon (association "Maths en Jeans").

8.4. Institutional commitment

Maurizio Gabbrielli has been elected Head of the Doctoral School in the EIT ICT Labs.

9. Bibliography

Major publications by the team in recent years

[1] M. BRAVETTI, G. ZAVATTARO. *A Foundational Theory of Contracts for Multi-party Service Composition*, in "Fundam. Inform.", 2008, vol. 89, n^{o} 4, pp. 451-478

[2] N. BUSI, M. GABBRIELLI, G. ZAVATTARO. *On the expressive power of recursion, replication and iteration in process calculi*, in "Mathematical Structures in Computer Science", 2009, vol. 19, n^{o} 6, pp. 1191-1222

[3] P. COPPOLA, S. MARTINI. *Optimizing optimal reduction: A type inference algorithm for elementary affine logic*, in "ACM Trans. Comput. Log.", 2006, vol. 7, n^{o} 2, pp. 219-260

[4] M. GABBRIELLI, S. MARTINI. *Programming Languages: Principles and Paradigms*, Springer, 2010

[5] D. HIRSCHKOFF, É. LOZES, D. SANGIORGI. *On the Expressiveness of the Ambient Logic*, in "Logical Methods in Computer Science", 2006, vol. 2, n^{o} 2

[6] U. D. LAGO, M. GABOARDI. *Linear Dependent Types and Relative Completeness*, in "Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011", IEEE Computer Society, 2011, pp. 133-142, http://www.cs.unibo.it/~gaboardi/publication/DalLagoGaboardi11lics.pdf

[7] I. LANESE, C. A. MEZZINA, A. SCHMITT, J.-B. STEFANI. *Controlling Reversibility in Higher-Order Pi*, in "Proc. of CONCUR 2011", J.-P. KATOEN, B. KÖNIG (editors), Lecture Notes in Computer Science, Springer, 2011, vol. 6901, pp. 297-311, http://www.cs.unibo.it/~lanese/publications/fulltext/concur2011.pdf.gz

[8] F. MONTESI, C. GUIDI, G. ZAVATTARO. *Composing Services with JOLIE*, in "Fifth IEEE European Conference on Web Services (ECOWS 2007)", 2007, pp. 13-22

[9] D. SANGIORGI. *An introduction to Bisimulation and Coinduction*, Cambridge University Press, 2012

Publications of the year

Doctoral Dissertations and Habilitation Theses

[10] M. ALBERTI. *On operational properties of quantitative extensions of λ-calculus*, Aix Marseille Université ; Università di Bologna, December 2014, https://hal.inria.fr/tel-01096067

[11] O. DARDHA. *Type Systems for Distributed Programs: Components and Sessions*, Università degli studi di Bologna, May 2014, https://tel.archives-ouvertes.fr/tel-01020998

[12] G. PELLITTA. *Extending Implicit Computational Complexity and Abstract Machines to Languages with Control*, Università di Bologna, May 2014, https://hal.inria.fr/tel-01090624

Articles in International Peer-Reviewed Journals

[13] R. AMADINI, M. GABBRIELLI, J. MAURO. *SUNNY: a Lazy Portfolio Approach for Constraint Solving*, in "Theory and Practice of Logic Programming (TPLP)", July 2014, pp. 509-524, https://hal.inria.fr/hal-01088489

[14] U. DAL LAGO, B. PETIT. *Linear dependent types in a call-by-value scenario*, in "Science of Computer Programming", 2014, 24 p. [*DOI :* 10.1145/2370776.2370792], https://hal.inria.fr/hal-01091610

[15] F. DE BOER, M. M. JAGHOORI, C. LANEVE, G. ZAVATTARO. *Decidability problems for actor systems*, in "Logical Methods in Computer Science", 2014, vol. 104, pp. 1 - 29 [*DOI :* 10.2168/LMCS-10(4:5)2014], https://hal.inria.fr/hal-01090952

[16] R. DI COSMO, J. MAURO, S. ZACCHIROLI, G. ZAVATTARO. *Aeolus: a Component Model for the Cloud*, in "Information and Computation", January 2014, pp. 100–121, https://hal.inria.fr/hal-01091062

Articles in Non Peer-Reviewed Journals

[17] I. LANESE, C. A. MEZZINA, F. TIEZZI. *Causal-Consistent Reversibility*, in "Bulletin of the EATCS", November 2014, vol. 114, 17 p. , https://hal.inria.fr/hal-01089350

Invited Conferences

[18] M. BRAVETTI, G. ZAVATTARO. *Choreographies and Behavioural Contracts on the Way to Dynamic Updates*, in "Proceedings First Workshop on Logics and Model-checking for Self*-Systems", Bertinoro (FC), Italy, September 2014, pp. 12 - 31 [*DOI :* 10.4204/EPTCS.168.2], https://hal.inria.fr/hal-01090924

[19] I. LANESE, M. LIENHARDT, M. BRAVETTI, V. STOLZ, R. SCHLATTE, E. B. JOHNSEN, G. ZAVATTARO. *Fault Model Design Space for Cooperative Concurrency*, in "ISOLA", Corfu, Greece, Lecture Notes in Computer Science, Springer, October 2014, vol. 8803, pp. 22 - 36 [*DOI :* 10.1007/978-3-662-45231-8_3], https://hal.inria.fr/hal-01089341

[20] D. SANGIORGI. *Higher-Order Languages: Bisimulation and Coinductive Equivalences (Extended Abstract)*, in "Coalgebraic Methods in Computer Science", Grenoble, France, April 2014, pp. 3 - 9 [*DOI :* 10.1007/978-3-662-44124-4_1], https://hal.inria.fr/hal-01092815

International Conferences with Proceedings

[21] B. ACCATTOLI, U. D. LAGO. *Beta Reduction is Invariant, Indeed*, in "Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)", Vienna, Austria, July 2014 [*DOI :* 10.1145/2603088.2603105], https://hal.inria.fr/hal-01091551

[22] R. AMADINI, M. GABBRIELLI, J. MAURO. *An enhanced features extractor for a portfolio of constraint solvers*, in "SAC 2014", Gyeongju, South Korea, March 2014, pp. 1357 - 1359 [*DOI :* 10.1145/2554850.2555114], https://hal.inria.fr/hal-01089183

[23] R. AMADINI, M. GABBRIELLI, J. MAURO. *Portfolio Approaches for Constraint Optimization Problems*, in "LION 8", Gainesville, Florida, United States, February 2014, pp. 21 - 35, https://hal.inria.fr/hal-01088429

[24] R. AMADINI, P. J. STUCKEY. *Sequential Time Splitting and Bounds Communication for a Portfolio of Optimization Solvers*, in "CP - Principles and Practice of Constraint Programming", Lyon, France, September 2014, pp. 108 - 124 [*DOI :* 10.1007/978-3-319-10428-7_11], https://hal.inria.fr/hal-01091664

[25] M. BERNARDO, D. SANGIORGI, V. VIGNUDELLI. *On the Discriminating Power of Passivation and Higher-Order Interaction*, in "Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)", Vienna, Austria, July 2014 [*DOI :* 10.1145/2603088.2603113], https://hal.inria.fr/hal-01089467

[26] M. BERNARDO, D. SANGIORGI, V. VIGNUDELLI. *On the Discriminating Power of Testing Equivalences for Reactive Probabilistic Systems: Results and Open Problems*, in "QEST", Florence, Italy, Lecture Notes in Computer Science, September 2014, vol. 8657, pp. 281 - 296 [*DOI :* 10.1007/978-3-319-10696-0_23], https://hal.inria.fr/hal-01089484

[27] M. BRAVETTI, M. CARBONE, T. HILDEBRANDT, I. LANESE, J. MAURO, J. A. PEREZ, G. ZAVATTARO. *Towards Global and Local Types for Adaptation*, in "SEFM 2013 Collocated Workshops", Madrid, Spain, Lecture Notes in Computer Science, Springer, 2014, vol. 8368, pp. 3 - 14 [*DOI :* 10.1007/978-3-319-05032-4_1], https://hal.inria.fr/hal-01089358

[28] M. CARBONE, O. DARDHA, F. MONTESI. *Progress as Compositional Lock-Freedom*, in "Coordination Models and Languages - 16th IFIP WG 6.1 International Conference, COORDINATION", Berlin, Germany, June 2014, https://hal.inria.fr/hal-01090742

[29] R. CRUBILLÉ, U. DAL LAGO. *On Probabilistic Applicative Bisimulation and Call-by-Value λ-Calculi*, in "23rd European Symposium on Programming", Grenoble, France, April 2014 [*DOI :* 10.1007/978-3-642-54833-8_12], https://hal.inria.fr/hal-01091564

[30] U. DAL LAGO, C. FAGGIAN, I. HASUO, A. YOSHIMIZU. *The Geometry of Synchronization*, in "Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)", Vienna, Austria, July 2014 [*DOI :* 10.1145/2603088.2603154], https://hal.inria.fr/hal-01091560

[31] U. DAL LAGO, D. SANGIORGI, M. ALBERTI. *On Coinductive Equivalences for Higher-Order Probabilistic Functional Programs*, in "The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages", San Diego, United States, January 2014 [*DOI :* 10.1145/2535838.2535872], https://hal.inria.fr/hal-01091573

[32] U. DAL LAGO, S. ZUPPIROLI. *Probabilistic Recursion Theory and Implicit Computational Complexity*, in "11th International Colloquium on Theoretical Aspects of Computing", Bucharest, Romania, September 2014 [*DOI :* 10.1007/978-3-319-10882-7_7], https://hal.inria.fr/hal-01091595

[33] M. DALLA PREDA, S. GIALLORENZO, I. LANESE, J. MAURO, M. GABBRIELLI. *AIOCJ: A Choreographic Framework for Safe Adaptive Distributed Applications*, in "SLE 2014", Västerås, Sweden, September 2014, pp. 161 - 170 [*DOI :* 10.1007/978-3-319-11245-9_9], https://hal.inria.fr/hal-01089187

[34] M. GABBRIELLI, S. GIALLORENZO, F. MONTESI. *Service-Oriented Architectures: From Design to Production Exploiting Workflow Patterns*, in "11th International Symposium on Distributed Computing and Artificial Intelligence", Salamanca, Spain, June 2014 [*DOI :* 10.1007/978-3-319-07593-8_17], https://hal.inria.fr/hal-01090128

[35] E. GIACHINO, N. KOBAYASHI, C. LANEVE. *Deadlock Analysis of Unbounded Process Networks*, in "CONCUR - Concurrency Theory - 25th International Conference", Rome, Italy, September 2014, pp. 63 - 77 [*DOI :* 10.1007/978-3-662-44584-6_6], https://hal.inria.fr/hal-01091749

[36] E. GIACHINO, I. LANESE, C. A. MEZZINA. *Causal-Consistent Reversible Debugging*, in "FASE", Grenoble, France, Lecture Notes in Computer Science, Springer, April 2014, vol. 8411, pp. 370 - 384 [*DOI :* 10.1007/978-3-642-54804-8_26], https://hal.inria.fr/hal-01089270

[37] E. GIACHINO, C. LANEVE. *Deadlock Detection in Linear Recursive Programs*, in "Formal Methods for Executable Software Models - 14th International School on Formal Methods for the Design of Computer, Communication, and Software Systems (SFM)", Bertinoro, Italy, June 2014, pp. 26 - 64 [*DOI :* 10.1007/978-3-319-07317-0_2], https://hal.inria.fr/hal-01091747

[38] E. GIACHINO, C. LANEVE. *Towards the Typing of Resource Deployment*, in "Leveraging Applications of Formal Methods, Verification and Validation. Specialized Techniques and Applications - 6th International Symposium, ISoLA", Corfu, Greece, October 2014, pp. 88 - 103 [*DOI :* 10.1007/978-3-662-45231-8_7], https://hal.inria.fr/hal-01091748

[39] C. GUIDI, S. GIALLORENZO, M. GABBRIELLI. *Towards a Composition-based APIaaS Layer*, in "CLOSER - The 4th International Conference on Cloud Computing and Service", Barcelona, Spain, April 2014 [*DOI :* 10.1007/978-3-319-07593-8_17], https://hal.inria.fr/hal-01090125

[40] J.-M. MADIOT, D. POUS, D. SANGIORGI. *Bisimulations Up-to: Beyond First-Order Transition Systems*, in "The 25th Conference on Concurrency Theory (CONCUR)", Rome, Italy, September 2014, pp. 93 - 108 [*DOI :* 10.1007/978-3-662-44584-6_8], https://hal.inria.fr/hal-01090509

[41] D. SANGIORGI. *Equations, contractions, and unique solutions*, in "Proceedings POPL", Mombay, India, January 2015 [*DOI :* 10.1145/2676726.2676965], https://hal.archives-ouvertes.fr/hal-01089205

[42] D. SANGIORGI, X. XU. *Trees from Functions as Processes*, in "25th International Conference on Concurrency Theory", Rome, Italy, September 2014, pp. 78 - 92 [*DOI :* 10.1007/978-3-662-44584-6_7], https://hal.inria.fr/hal-01092809

[43] M. SOLIERI. *Geometry of Resource Interaction*, in "Proceedings Third International Workshop on Linearity", Vienna, Austria, Electronic Proceedings in Theoretical Computer Science, July 2014, forthcoming, https://hal-univ-paris13.archives-ouvertes.fr/hal-01091754

[44] A. YOSHIMIZU, I. HASUO, C. FAGGIAN, U. DAL LAGO. *Measurements in Proof Nets as Higher-Order Quantum Circuits*, in "23rd European Symposium on Programming", Grenoble, France, April 2014, pp. 371 - 391 [*DOI :* 10.1007/978-3-642-54833-8_20], https://hal.inria.fr/hal-01091582

Scientific Books (or Scientific Book chapters)

[45] *Proceedings 7th Interaction and Concurrency Experience*, Electronic Proceedings in Theoretical Computer Science, October 2014, vol. 166 [*DOI :* 10.4204/EPTCS.166], https://hal.inria.fr/hal-01089381

[46] F. MONTESI, C. GUIDI, G. ZAVATTARO. *Service-Oriented Programming with Jolie*, in "Web Services Foundations", A. BOUGUETTAYA, Q. Z. SHENG, F. DANIEL (editors), Springer, 2014, pp. 81-107, https://hal.inria.fr/hal-00909323

Research Reports

[47] R. DI COSMO, A. EICHE, J. MAURO, G. ZAVATTARO, S. ZACCHIROLI, J. ZWOLAKOWSKI. *Automatic Deployment of Software Components in the Cloud with the Aeolus Blender*, Inria Sophia Antipolis, January 2015, https://hal.inria.fr/hal-01103806

[48] J.-M. MADIOT, D. POUS, D. SANGIORGI. *Bisimulations up-to: beyond first-order transition systems*, March 2014, https://hal.archives-ouvertes.fr/hal-00990859

Other Publications

[49] D. HIRSCHKOFF, J.-M. MADIOT, X. XU. *A behavioural theory for a pi-calculus with preorders*, February 2014, https://hal.archives-ouvertes.fr/hal-00949521

[50] I. LANESE, C. A. MEZZINA, J.-B. STEFANI. *Reversibility in the higher-order π-calculus*, June 2014, https://hal.inria.fr/hal-01081714

References in notes

[51] U. DAL LAGO, U. SCHÖEPP. *Functional Programming in Sublinear Space*, in "Programming Languages and Systems, 19th European Symposium, Proceedings", Lecture Notes in Computer Science, Springer, 2010, vol. 6012, pp. 205-225