



IN PARTNERSHIP WITH:
Université de Bologne (Italie)

Activity Report 2012

Project-Team FOCUS

Foundations of Component-based Ubiquitous Systems

IN COLLABORATION WITH: Dipartimento di Scienze dell'Informazione

RESEARCH CENTER
Sophia Antipolis - Méditerranée

THEME
Distributed Systems and Services

Table of contents

1. Members	1
2. Overall Objectives	2
2.1. Overall Objectives	2
2.2. Highlights of the Year	2
3. Scientific Foundations	2
4. Application Domains	3
4.1. Ubiquitous Systems	3
4.2. Service Oriented Computing and Cloud Computing	3
4.3. Software Product Lines	3
5. Software	3
5.1. Jolie	3
5.2. Others	4
6. New Results	4
6.1. Service-oriented computing	4
6.1.1. Primitives	5
6.1.2. Contracts and sessions	5
6.2. Adaptability and faults	5
6.2.1. Reversibility	5
6.2.2. Primitives for adaptable and evolving components	5
6.2.3. Reconfigurability in the cloud	6
6.2.4. Delta-Oriented Programming and Software Product Lines	6
6.3. Resource Control	6
6.4. Verification of extensional properties	7
6.4.1. Adaptability	7
6.4.2. Object-orientation	7
6.4.3. Synchronization primitives	7
6.4.4. Coinduction	8
6.5. Expressiveness of computational models	8
7. Partnerships and Cooperations	9
7.1. National Initiatives	9
7.2. European Initiatives	9
7.2.1. FP7 Projects	9
7.2.2. Collaborations in European Programs, except FP7	9
7.2.3. Collaborations with Major European Organizations	10
7.3. International Initiatives	11
7.4. International Research Visitors	11
8. Dissemination	11
8.1. Scientific Animation	11
8.2. Popularization	13
8.3. Teaching - Supervision	13
8.3.1. Teaching	13
8.3.2. Supervision	14
8.3.3. Juries	15
9. Bibliography	15

Project-Team FOCUS

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

Creation of the Project-Team: January 01, 2010 .

1. Members

Faculty Members

Mario Bravetti [“Prof. Associato”, Bologna]
Ugo Dal Lago [“Ricercatore”, Bologna]
Maurizio Gabbriellini [“Prof. Ordinario”, Bologna]
Ivan Lanese [“Ricercatore”, Bologna]
Cosimo Laneve [“Prof. Ordinario”, Bologna]
Simone Martini [“Prof. Ordinario”, Bologna]
Davide Sangiorgi [Team leader, “Prof. Ordinario”, Bologna, HdR]
Gianluigi Zavattaro [“Prof. Associato”, Bologna]

External Collaborators

Claudio Guidi [University of Padova, Italy, and ‘Italiana Software’]
Daniel Hirschhoff [ENS Lyon, HdR]
Fabrizio Montesi [PhD student, IT University of Copenhagen, and ‘Italiana Software’]
Marco Pistore [Senior Researcher, FBK Trento (I)]

PhD Students

Ferdinanda Camporesi [Bologna, joint supervision with ENS Paris]
Ornela Dardha [Bologna]
Tudor Alexandru Lascu [Bologna]
Jean-Marie Madiot [ENS Lyon]
Giulio Pellitta [Bologna]
Paolo Parisen Toldin [Bologna]
Michele Alberti [Marseille]
Marco Solieri [Paris Nord]

Post-Doctoral Fellows

Mila Dalla Preda [EU project Hats and Univ. Bologna]
Elena Giachino [ANR “REVER”]
Marco Gaboardi [Marie Curie scholarship]
Michael Lienhardt [EU project Hats]
Jacopo Mauro [ANR “Aeolus”]
Barbara Petit [ARC “Eternal”]

Administrative Assistant

Nathalie Woodward [shared with another team]

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.

2.2. Highlights of the Year

- Jacopo Mauro's PhD thesis, entitled "Constraints meet concurrency", has been awarded a price as the best dissertation for the two years 2010 and 2011 by the Italian Association for Logic Programming (GULP).

3. Scientific Foundations

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.

4.3. Software Product Lines

A Software Product Line is a set of software systems that together address a particular market segment or fulfill a particular mission. Today, Software Product Lines are successfully applied in a range of industries, including telephony, medical imaging, financial services, car electronics, and utility control [54]. Customization and integration are keywords in Software Product Lines: a specific system in the family is constructed by selecting its properties (often technically called “features”), and, following such selection, by customizing and integrating the needed components and deploying them on the required platform.

5. Software

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.

In 2012 the development of Jolie has continued. The main activities have been:

- We have enhanced the correlation mechanism in Jolie to handle multiparty sessions with concurrent interactions with multiple participants.
- We have developed a compiler that projects choreography-based programs in Chor (<http://www.chor-lang.org>) to Jolie.
- We have developed a new website for Jolie (using Jolie itself), significantly updating its documentation.
- We have improved Jolie's compatibility with the Java RMI technology.
- We have developed a first experimental implementation of a monitoring layer for Jolie services.

As last year, so in 2012 Jolie has been used for teaching, in a master course at the IT University of Copenhagen (ITU, Denmark) and in a master course at the Technical University of Denmark (DTU, Denmark).

5.2. Others

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

- *IntML* is a functional programming language guaranteeing sublinear space bounds for all programs [53]. See the Activity Reports of previous years (in particular 2010) for more details. During 2012 no substantial modifications have been made.
- *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.
- We have implemented a technique for the deadlock analysis of a concurrent object oriented language (ABS, designed within the European project HATS). The technique consists of
 - an inference system for contracts to be associated to methods. Contracts are terms that retain information about resource dependencies;
 - a fixpoint algorithm for solving contract definitions, which are recursive and may introduce new resource names.

The release of the software is planned for early 2013.

- *Croll-pi Interpreter* (<http://proton.inrialpes.fr/~mlienhar/croll-pi/implem/>). We have developed an interpreter for croll-pi using Maude. 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 used the interpreter to test the expressive power of croll-pi on various problems, including the 8-queen problem, error handling in an automotive scenario from the EU project Sensoria, and constructs for distributed error handling such as stabilizers.

For other software, such as PiDuce, see the activity reports for Focus of previous years.

6. New Results

6.1. Service-oriented computing

Participants: Mila Dalla Preda, Ornella Dardha, Maurizio Gabbriellini, Elena Giachino, Claudio Guidi, Jacopo Mauro, Fabrizio Montesi, Davide Sangiorgi.

6.1.1. Primitives

In the context of Service-Oriented Architectures (SOAs), the integration of services is an important aspect that is usually addressed by using specific tools, such as Enterprise Service Bus (ESB). Although widely used these ad-hoc solutions do not exploit the possibility of using a mechanism of interface extension to foster the rapid prototyping and deployment of in-the-middle entities that compose services abstracting from the order in which they exchange messages. We have proposed [29] a framework to perform service integration, based on the extension of service interfaces, capturing a class of service integrators that are decoupled from the services they integrate in an SOA. We also provide a reference implementation for the primitive of service integration by extending the Jolie language, thus allowing for the experimentation with real SOA scenarios. We have shown [30] how our methodology differs from the standard practice with ESB.

6.1.2. Contracts and sessions

Contracts are descriptions of the functionalities offered by a component or a service, and of the way these functionalities may be accessed by clients. A contract may include a description of the component capabilities, place constraints on their usage, as well as declare preferences, entitlements and credentials. When a client wants to use one of the functionalities offered, it engages a dialogue (e.g., a sequence of interactions) with the servers; this is usually called a session.

Contracts specify the expected dialogue in a session and they can be expressed as types, usually called *session types* in this context.

A session type describes communication by specifying type and direction of data exchanged between two parties. When session types and session primitives are added to the syntax of the types and terms of a language, they give rise to additional separate syntactic categories. As a consequence, there may be duplication of efforts in the theory: the proofs of properties must be checked both on ordinary types and on session types. We have shown [32] that this duplication is not necessary, by exhibiting an encoding of (dyadic) session types into ordinary types. Using the encoding, the properties of session types are derived as straightforward corollaries.

We have also studied the problem of handling unexpected or unwanted conditions in sessions, that may change the default execution of distributed communication protocols. We have proposed [14] a global escape mechanism; it can handle such events while preserving compatibility of multiparty conversations. This flexibility enables us to model complex exceptions such as criss-crossing global interactions and error handling for distributed cooperating threads. Guided by multiparty session types, our semantics is proven to provide a termination algorithm for global escapes, as well as further safety properties, such as progress within the session and atomicity of escapes with respect to the subset of involved participants.

6.2. Adaptability and faults

Participants: Mario Bravetti, Elena Giachino, Ivan Lanese, Michael Lienhardt, Jacopo Mauro, Davide Sangiorgi, Gianluigi Zavattaro.

6.2.1. Reversibility

We have continued the study of reversibility started in the past years, aimed at developing programming abstractions for reliable distributed systems. We have shown [39] preliminary results on the interplay between reversibility and compensations, which are a main ingredient in many existing techniques for reliability, in particular long running transactions.

We have then applied [43] our reversibility theory to μOz , a concurrent programming language defined by a stack-based abstract machine, and we make it reversible. This is a first step towards the definition of reversible variants of more complex languages. As additional result we show that the memory overhead due to reversibility is optimal as an order of magnitude.

6.2.2. Primitives for adaptable and evolving components

We study primitives for adaptable and evolving components both in an abstract algebraic setting and in a more concrete setting based on the ABS object-oriented language.

We have defined [13] adaptable processes, a concurrent higher-order calculus where processes have a location, and are sensible to actions of update at runtime. This allows us to express a wide range of evolvability patterns. We have also defined [24] a temporal logic over adaptable processes, with examples of the expressiveness of the logic and of its significance in relation to the calculus of adaptable processes.

A different direction has focused on ABS, a concurrent object-oriented language based on futures for asynchronous method invocations and on object groups for concurrency control. We have given [38] an overview of the architectural aspects of ABS: a feature-driven development workflow, a formal notion of deployment components for specifying environmental constraints, and a dynamic component model that is integrated into the language. We have employed an industrial case study to demonstrate how the various aspects work together in practice. In [40] we have focused on the component model and studied techniques allowing safe dynamic reconfiguration. Our approach adds to ABS: i) output ports to represent variability points, ii) critical sections to control when updates of the software can be made and iii) hierarchy to model locations and distribution. These different notions work together to allow dynamic and safe update of a system.

6.2.3. Reconfigurability in the cloud

The cloud is a relevant application domain for FOCUS. We have considered [35] the problem of deploying and (re)configuring resources in a cloud setting, where interconnected software components and services can be deployed on clusters of heterogeneous (virtual) machines that can be created and connected on-the-fly. We introduce a component model to capture similar scenarii from realistic cloud deployments, and instrument automated planning of day-to-day activities such as software upgrade planning, service deployment, elastic scaling, etc. We formalize the model and characterize the feasibility and complexity of configuration achievability.

6.2.4. Delta-Oriented Programming and Software Product Lines

Delta-oriented programming (DOP) provides a technique for implementing Software Product Lines based on modifications (add, remove, modify) to a core program. Unfortunately, such modifications can introduce errors into a program, especially when type signatures of classes are modified in a non-monotonic fashion. To deal with this problem in we have designed [42] a type system for delta-oriented programs based on row polymorphism. This exercise elucidates the close correspondence between delta-oriented programs and row polymorphism.

In [41] we have studied the notion of conflict for a variant of DOP without features, separating out the notions of hard and soft conflict. Specifically, we have defined a language for this subset of DOP and give a precise, formal definition of these notions. We have then extended the type system in [42] to ensure that the computation of a well-typed product will always succeed and has an unambiguous result.

6.3. Resource Control

Participants: Michele Alberti, Ugo Dal Lago, Marco Gaboardi, Daniel Hirschhoff, Simone Martini, Paolo Parisen Toldin, Giulio Pellitta, Barbara Petit, Davide Sangiorgi, Marco Solieri.

In Focus, we study both foundations and methodologies for controlling the amount of resources programs and processes make use of. The employed techniques mainly come from the world of type theory and proof theory, and as such have been used extensively in the context of sequential computation. Interesting results have been obtained recently indicating that those techniques can be quite useful in the concurrent context too, thus being potentially interesting for CBUS.

During 2012, we have continued our work on intensionally complete techniques for the complexity analysis of functional programs. In [15] a relatively complete type system from which complexity of call-by-name terms has been introduced, while in [25] the same approach is used in a call-by-value setting. The introduced methodology allows us to reduce the problem at hand to the verification of a set of first-order proof-obligations. No information is lost along the reduction.

The interpretation method has been the object of a couple of investigations. On the one hand, we have proved a necessary condition for a first-order program to admit a quasi-interpretation [12]: it must be blind, namely it must be somehow insensible to its argument value, but only sensible to their length. Moreover, we have introduced a new methodology for the complexity analysis of higher-order programs based on an higher-order generalizations of ordinary polynomial interpretations and quasi-interpretations [23].

Among the most foundational works in this area, we should also mention those about invariance results on cost models [16], [22]: we proved that in many different cases, the number of beta-reduction steps is an adequate cost-model for the lambda calculus. These results are potentially useful for complexity analysis, in that they show that a natural and quite intuitive cost model is indeed reasonable, meaning that evaluation can be simulated by finer-grained models of computation within a polynomial overhead.

Finally, some of our works have to do with the semantics of various sorts of lambda calculi with linearity constraints: a non-deterministic extension of the call-by-value lambda calculus, which corresponds to the additive fragment of the linear-algebraic lambda-calculus [36]; a lambda calculus with constructors that decomposes the pattern matching à la ML into some atomic rules [44]; a categorical approach to model the programming language SIPCF that has been conceived in order to program only linear functions between Coherence Spaces [20].

We have also continued our work on techniques for ensuring termination of programs, studying [34] how to transport techniques initially devised for processes onto sequential higher-order languages with imperative features (e.g., λ -calculi with references). The method employed makes it possible to combine term rewriting measure-based techniques for termination of imperative languages with traditional approaches to termination in purely functional languages, such as logical relations.

6.4. Verification of extensional properties

Participants: Mario Bravetti, Daniel Hirschhoff, Cosimo Laneve, Jean-Marie Madiot, Tudor Alexandru Lascu, Davide Sangiorgi, Gianluigi Zavattaro.

Extensional refers to properties that have to do with behavioral 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. A substantial amount of the work carried out this year has to do with the transfer of techniques from the area of concurrency theory to the investigation of properties in adaptable systems, object-oriented concurrent systems, and systems based on specific synchronization mechanisms.

6.4.1. Adaptability

We mentioned earlier the process calculus of adaptable processes [13] and the related temporal logic [24]. In the same papers, we have addressed the (un)decidability of two safety properties related to error occurrence, and we have explained how the proof techniques in [13] can be extended to prove (un)decidability results for the temporal logic.

6.4.2. Object-orientation

We have considered concurrent object-oriented languages with futures and cooperative scheduling. Verification of deadlock in such systems is a nontrivial task due to the dynamic and unbounded creation of futures. We have introduced [45] a technique to prove deadlock freedom for such systems by translating a concrete program to an abstract version of the program, and then encoding such abstract program into a Petri net. Deadlock can be detected on Petri nets via checking the reachability of a distinct marking: absence of deadlocks in the Petri net constitutes deadlock freedom of the concrete system.

6.4.3. Synchronization primitives

We have investigated [33] the impact of node and communication failures on the decidability and complexity of parametric verification of a formal model of ad hoc networks, in which finite state processes communicate via selective broadcast. We have considered three possible kinds of node failures –intermittence, restart, and crash– and three cases of communication failures –nondeterministic message loss, message loss due to conflicting

emissions, and detectable conflicts. Interestingly, we have proved that the considered decision problem (reachability of a control state) is decidable for node intermittence and message loss (either nondeterministic or due to conflicts) while it turns out to be undecidable for node restart/crash, and conflict detection. The conclusion is that verification is decidable only when processes are unaware of the occurrence of a failure.

In another line of work, we have studied the impact of dualities and symmetries in synchronization primitives for message-passing processes [37]. We have shown that in languages where input and output are dualisable (e.g., variants of the π -calculus such as πI and fusion), duality breaks with the addition of ordinary input/output types. We have then considered the minimal symmetrical conservative extension of π -calculus input/output types. We have proved duality properties for it. As example of application of the dualities, we have used this language to relate two encodings of λ -calculus, by Milner and by van Bakel and Vigliotti, syntactically quite different from each other. Thus, results on one encoding can be transferred onto the other one.

6.4.4. Coinduction

Induction is a pervasive tool in Computer Science and Mathematics for defining structures and reasoning on them. Coinduction is the dual of induction, and as such it brings in tools that are quite different from those provided by induction. The best known instance of coinduction is bisimulation, mainly employed to define and prove equalities among potentially infinite objects: processes, streams, non-well-founded sets, and so on. Sangiorgi has completed [48], [51] two comprehensive textbooks on bisimulation and coinduction (in [51], Sangiorgi is an editor, and author of two chapter contributions [49], [47]). The books explain the fundamental concepts and techniques, and the duality with induction. A special emphasis is put on bisimulation as a behavioural equivalence for processes. Thus the books also serve as an introduction to models for expressing processes, and to the associated techniques of operational and algebraic analysis.

6.5. Expressiveness of computational models

Participants: Cosimo Laneve, Maurizio Gabbrielli, Gianluigi Zavattaro.

Expressiveness refers to the study of the expressive power of computational models.

We have studied [46] the expressiveness of an actor-based language similar to the language ABS developed in Hats. We have identified the presence/absence of fields as a relevant 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. Such decidability result holds in actors with states when the number of actors is finite and the state is read-only.

Our other study of expressiveness has been made on Constraint Handling Rules (CHR), a committed-choice declarative language originally designed for writing constraint solvers and that is nowadays a general purpose language. The study of CHR is interesting within Focus as this kind of languages, having both constraint solving and concurrency features, allow us to express in a natural way quantitative aspects related to resources. Moreover, constraints may be used to describe dynamic adaptation and evolution of systems. CHR programs consist of multi-headed guarded rules which allow one to rewrite constraints into simpler ones until a solved form is reached. Many empirical evidences suggest that multiple heads augment the expressive power of the language (somehow, it can be considered similar to multiple synchronization gathering n processes simultaneously), however no formal result in this direction had been proved so far. We have proved [18] a number of expressiveness results to support such a claim. First, we have analyzed the Turing completeness of CHR with respect to the underlying constraint theory. If the constraint theory is powerful enough then restricting to single head rules does not affect Turing completeness. On the other hand, differently from the case of the multi-headed language, the single head CHR language is not Turing powerful when the underlying signature (for the constraint theory) does not contain function symbols. Then we have proved that, no matter which constraint theory is considered, under some reasonable assumptions it is not possible to encode the CHR language (with multi-headed rules) into a single headed language while preserving the semantics of the programs. Moreover, under some stronger assumptions, considering an increasing number of atoms in the head of a rule augments the expressive power of the language.

7. Partnerships and Cooperations

7.1. National Initiatives

- AEOLUS (Mastering the Cloud Complexity) is an ANR-ARPEGE project started on 1st December 2010 and with a 40-month duration. 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 distribution and the dependencies of packages, in such a way that *modification plans* can be (semi)automatically synthesized when packages should be updated or the system should be reconfigured. Main persons involved: Gabbrielli, Mauro, Sangiorgi, Zavattaro.
- ETERNAL (Interactive Resource Analysis) is an Inria-ARC project which started on January 1st, 2011 and will end on December 31st, 2012. ETERNAL aims at putting together ideas from Implicit Computational Complexity and Interactive Theorem Proving, in order to develop new methodologies for handling quantitative properties related to program resource consumption, like execution time and space. People involved: Dal Lago, Gaboardi, Martini, Petit.
- 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.
- The ANR project PACE (Processus non-standard: Analyse, Coinduction, et Expressivité) has been recently accepted but will start only 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: Hirschhoff (project coordinator), Dal Lago, Lanese, Sangiorgi, Zavattaro.

7.2. European Initiatives

7.2.1. FP7 Projects

- Hats (Highly Adaptable and Trustworthy Software using Formal Models) is an EU Integrated Project from FP7, started March 2009 and with a 4 year duration. Hats studies formal methods for obtaining high adaptability combined with trustworthiness in the setting of object-oriented languages and software product lines. Most Focus members are involved.
- PLATFORM (Practical Light Types for Resource Consumption) is a Marie Curie IOF project from FP7, started July 2011 with a three-year span. It involves one Focus member (Gaboardi) in research work at University of Pennsylvania and in Bologna. Project aim is the development of a practical programming language with information, in the form of dependent types, about the resources needed by programs during their execution, and where type checking a program will naturally corresponds to exhibit a certification of its resource consumption.

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, will use 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, Lanese, Laneve, Sangiorgi, Zavattaro.
- The EIT activity "Intelligent Services for Digital Cities" in the context of the Digital Cities Action Line, has been approved, with funding to be spent in 2013. Main persons involved: Gabbrielli.

7.2.3. Collaborations with Major European Organizations

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, Gaboardi, Martini, Sangiorgi. Several visit exchanges during the year, in both directions. One joint PhD supervision (J.-M. Madiot).
- Inria EPI Sardes (on models and languages for components, reversibility). Contact person(s) in Focus: Lanese, Sangiorgi. A number of visits in both directions. One joint PhD supervision (C. Mezzina).
- Laboratoire d'Informatique, Université Paris Nord, Villetaneuse (on implicit computational complexity). Contact person(s) in Focus: Dal Lago, Gaboardi, 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, Gaboardi, Martini, Sangiorgi. Various 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: Zavattaro. Various short visits in both directions during the year.
- University of Innsbruck (on resource control and termination). Contact person(s) in Focus: Dal Lago. A few short visits during 2012.
- Inria EPI Indes, (on orchestration and programming languages). A common meeting was organised in Bologna, January 2012, where 4 people from Indes and almost everybody from Focus participated.
- EPI Carte, Inria-Nancy Grand Est and LORIA (on implicit computational complexity). Contact person(s) in Focus: Dal Lago, Gaboardi. A few short visits during 2012.
- 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.
- Facultad de Informatica, Universidad Complutense de Madrid (on web services). Contact person(s) in Focus: Bravetti. Bravetti is an external collaborator in the Spanish Ministry of Science and Education project TESIS (advanced methodologies and tools for TESTing and web servIceS).

7.3. International Initiatives

7.3.1. Inria International Partners

- Department of Computer and Information Science, University of Pennsylvania. There has been several collaborations in the past. Presently M. Gaboardi is a long-term visiting researcher in the programming language group, working on resource control and programming languages.

7.4. International Research Visitors

7.4.1. Visits of International Scientists

Among the visits below, we note 2 long-term visits: the 6-month visit of Matias Lee (paid by a EADIC II - Erasmus Mundus scholarship); and the sabbatical year of Xian Xu, from East China University of Science and Technology, Shanghai (paid a scholarship from the Chinese Science Foundation).

- Peter Hofner, Nicta, Sydney, Australia. 1 week end of January. Topic: Formal Methods for Wireless Networks.
- Martin Avanzini, Institute of Compute Science, University of Innsbruck, Austria. 23 to 27 April, 2012. Topic: order-theoretic approaches to complexity analysis of functional programs.
- Marco Carbone, IT University of Copenhagen. A week in May/June 2012. Topic: faults and compensations in choreography languages.
- Raju Halder, Macquarie University, Australia. 2 days in May 2012. Topic: Abstract Interpretation and concurrency.
- Luca Fossati, Imperial College London. A week in May 2012. Topic: Liveness properties by means of types in the π -calculus.
- Xian Xu has began in May a year sabbatical in Focus. He is lecturer at the East China University of Science and Technology in Shanghai.
- Matias David Lee. Currently PhD student at Universidad Nacional de Córdoba, Argentina, has spent 5 months during 2012, from January to May (having arrived in November 2011), for a long-term visit.
- Bertrand Meyer, ETH Zurich. One day in September 2012. Topic: Concurrent Object-Oriented Programming.
- Mariangiola Dezani, Univ. Turin, Italy. 4 days October 2012. Topic: Session Types.
- Clement Aubert, LIPN Université Paris 13. From 5 to 16 November 2012. Topic: implicit complexity and the geometry of interaction.
- Matthieu Perrinel, LIP ENS Lyon. From 17 to 21 December 2012. Topic: context semantic and decideable criteria for complexity analysis.

8. Dissemination

8.1. Scientific Animation

M. Bravetti: Program Committee member of the following conferences: ESOC 2012: European Conference on Service-Oriented and Cloud Computing; SITIS 2012 (IBCS): The 8th International Conference On Signal-Image Technology and Internet-Based Systems, Track on Internet-Based Computing and Systems; WS-FMDS 12: 2nd Workshop on Formal Methods in the Development of Software; CLOUD 2012: 5th IEEE International Conference on Cloud Computing; ICST 2012: 5th IEEE International Conference on Software Testing, Verification, and Validation; IADIS: International Conference on Applied Computing 2012; PLACES 2012: 5th Int. Workshop on Programming Language Approaches to Concurrency and Communication-centric Software; WS-FM 2012: 9th International Workshop on Web Services and Formal Methods.

Member of the editorial board of the journal J.UCS Journal of Universal Computer Science.

M. Dalla Preda: Program Committee member of the following conferences: 5th IEEE international conference Malware 2012; Software Security and Protection Workshop 2012; Workshop on Program Protection and Reverse Engineering 2012.

U. Dal Lago: Program Committee member of the following conferences: 39th International Colloquium on Automata, Languages and Programming (ICALP 2012); Third International Workshop on Developments in Implicit Complexity (DICE 2012).

Invited speaker at Second International Workshop on Linearity, Tallin, Estonia, 1 April, 2012. Guest Editors of a special number of Theoretical Computer Science devoted to DICE 2012.

Member of the organising committee of the “Logic and interactions 2012” weeks, Marseille 30 January – 2 March 2012.

M. Gaboardi: Program Committee member for the Third Workshop on Developments in Implicit Computational Complexity

Invited speaker at the ASL meeting special session on Structural Proof Theory and Computing April 2-3, 2012, Madison, USA.

I. Lanese: Program Committee member of the following conferences: 9th International Workshop on Web Services and Formal Methods (WS-FM 2012); 5th Interaction and Concurrency Experience (ICE 2012).

Co-chair of the Service Oriented Architectures and Programming track of the 27th Annual ACM Symposium on Applied Computing (SAC 2012). Workshop Co-Chair of ESOC 2012, the European Conference on Service-Oriented and Cloud Computing 19-21 September 2012, Bertinoro, Italy.

Member of the organising committee of the 5th Interaction and Concurrency Experience (ICE 2012).

C. Laneve: Program Committee member of the following conferences: 23rd International Conference on Concurrency Theory (CONCUR); 22nd European Symposium on Programming (ESOP).

Invited speaker at 7th International Symposium on Trustworthy Global Computing Newcastle upon Tyne, UK, 7-8 September 2012.

M. Gabbrielli: Member of the editorial board of the Int’l Journal Theory and Practice of Logic Programming. Member of the program committee of the 9th International Workshop on Constraint Handling Rules; Fifteenth International Symposium on Practical Aspects of Declarative Languages (PADL’13).

Expert Committee member for 2012 EAPLS Best Ph.D. Dissertation award.

Organizer of the 2012 Bertinoro school for Ph.D. students in computer science (BISS 2012).

S. Martini: Program Committee member of the Second International Workshop on Linearity (FOPARA), Tallinn, Estonia.

Invited speaker at Pisa Summer Workshop on Proof Theory, Pisa, Italy.

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

Program Committee member of the conferences: 38th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM); Twenty-Seventh Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2012); 39th International Colloquium on Automata, Languages and Programming (ICALP 2012),

Invited speaker at the Milner Symposium, 15-18 April 2012, Edimburgh UK, 2012 Microsoft Summer School on Concurrency, August 22-29 2012, St. Petersburg, Russia.

Co-director of the International School HATS-FMCO on Formal Models for Components and Objects, 24-28 September 2012 University Residential Center of Bertinoro, Italy.

G. Zavattaro: Program Committee member of the following conferences: 6th International Workshop on Reachability Problems; 9th International Symposium on Formal Aspects of Component Software (FACS); 4th Fourth International Conferences on Advanced Service Computing (SERVICE COMPUTATION 2012); 6th IEEE International Symposium on Theoretical Aspects of Software Engineering (TASE); 5th IEEE International Conference on Cloud Computing (CLOUD 2012).

Invited speaker at 3rd International Workshop on Interactions between Computer Science and Biology (CS2Bio'12), 16 June 2012, Stockholm, Sweden.

Member of the Editorial Board of the Journal of Software (JSW), Academy Publisher.

General Chair of ESOCC 2012, the European Conference on Service-Oriented and Cloud Computing 19-21 September 2012, Bertinoro, Italy.

8.2. 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);
- consultant of the Minister of Education on admission tests for high school professorship admission;
- the talks “Quello che i calcolatori possono e non possono fare”, MassaScienza, Massa Carrara; Lingua universalis: Calcolo, informatica e linguaggio, workshop T4T, Turin; Alan Turing pioniere dell’era digitale, ScienzaGiovane, Bologna; Calcolo e simboli: la scienza digitale, Accademia delle Scienze di Bologna and Accademia di Agricoltura Scienze e Lettere di Verona;
- the paper [21].

Ugo Dal Lago and Simone Martini have given short presentations at “Scienzagiovane”, a forum of science popularisation in Bologna mainly targeting high-school students.

8.3. Teaching - Supervision

8.3.1. Teaching

Below are the teaching activities of the Focus members during 2011.

- Mario Bravetti
 - Undergraduate: “Tecnologie Web”, 46 hours, 3rd year, University of Bologna, Italy.
 - Master: “Tecnologie Web Avanzate”, 46 hours, 1st or 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.
- Marco Gaboardi
 - Master: “Advanced Programming”, 40 hours. 1st or 2nd year. University of Pennsylvania, USA.
- Ivan Lanese
 - Undergraduate: “Programmazione”, 40 hours, 1st year, University of Bologna, Italy.

- Cosimo Laneve
 - Undergraduate: “Programmazione”, 90 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: “Types and programming languages”, 40 hours, 1st year, University of Bologna, Italy.
 - Master: “Basic Computability”, 12 hours, 1st year and 2nd year, University of Bologna, Italy.
 - Master: “Logical foundations of computer science”, 40 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.
- Gianluigi Zavattaro
 - Master: “Informatica Teorica”, 72 hours, 1st year, University of Bologna, Italy.
 - Master: “Linguaggi di Programmazione”, 48 hours, 2nd year, 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”, 40 hours, 2nd year, University of Bologna, Italy.
 - Undergraduate: “Attività di supporto alla didattica universitaria di Informatica nell’ambito della Piattaforma di e-learning A3”, 130 hours, 1st year, University of Bologna, Italy.
 - Undergraduate: “Laboratorio Informatica 1”, 60 hours, 1st year, University of Bologna, Italy.
- Elena Giachino
 - Undergraduate: “Programmazione”, 40 hours, 1st year, University of Bologna, Italy.

8.3.2. Supervision

Completed Phd thesis:

- Jacopo Mauro, “Constraints meet concurrency”. Dottorato di Ricerca in Informatica, University of Bologna, April 2012 Supervisors: M. Gabbrielli. [10]
- Claudio Mezzina, “Réversibilité dans le pi calcul d’ordre supérieur”, April 2012, University of Grenoble and University of Bologna, Supervisors: Jean-Bernard Stefani and Davide Sangiorgi. [11]

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.

- Ornella Dardha, january 2011, Types for processes, D. Sangiorgi.
- Ferdinanda Camporesi, january 2009, Analysis of system biology, R. Cousot (Ecole Normale, Paris) and M. Gabbrielli
- Tudor Alexandru Lascu, january 2011, Formal methods for modelling and analyzing concurrent systems, C. Laneve and G. Zavattaro
- Jean-Marie Madiot, september 2011, Types, proofs and proof techniques for process calculi, D. Hirschhoff and D. Sangiorgi.
- Giulio Pellitta, january 2011, Implementations of geometry of interaction, S. Martini.
- Paolo Parisen Toldin, january 2010, Implicit computational complexity for probabilistic classes, S. Martini.
- Michele Alberti, december 2011; Separation results in the algebraic lambda-calculus; L. Regnier (Marseille) and S. Martini
- Marco Solieri, december 2011; Complexity of reduction in sharing graphs, S. Guerrini (Paris Nord) and S. Martini.

A few other PhD students are (co)-supervised by members of Focus, but currently are not in the Focus team (some of them might be added in 2013):

- Sara Zuppiroli (Bologna), january 2010; models for quantum computing; U. Dal Lago.
- Roberto Amadini (Bologna), january 2012, Constraint Programming, M. Gabbrielli.
- Alessandro Rioli (Bologna), january 2012; bisimulation equivalence in quantum lambda calculi; U. Dal Lago.

8.3.3. *Juries*

D. Sangiorgi has been rapporteur and member of the evaluation committee of the “Habilitation a Diriger des Recherches” (HDR) of Ludovic Henrio, Université de Nice, and Daniele Varacca, Université Paris 7.

U. Dal Lago has been rapporteur and member of the PhD jury for Antoine Madet, Université Paris 7.

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, p. 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, p. 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, p. 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, p. 133-142, <http://www.cs.unibo.it/~gaboardi/publication/DalLagoGaboardi11lics.pdf>.
- [7] I. LANESE, C. A. MEZZINA, J.-B. STEFANI. *Reversing Higher-Order Pi*, in "CONCUR", P. GASTIN, F. LAROUSSINIE (editors), Lecture Notes in Computer Science, Springer, 2010, vol. 6269, p. 478–493, <http://www.cs.unibo.it/~lanese/publications/fulltext/concur2010.pdf.gz>.
- [8] F. MONTESI, C. GUIDI, G. ZAVATTARO. *Composing Services with JOLIE*, in "Fifth IEEE European Conference on Web Services (ECOWS 2007)", 2007, p. 13-22.
- [9] D. SANGIORGI. *An introduction to Bisimulation and Coinduction*, Cambridge University Press, 2011.

Publications of the year

Doctoral Dissertations and Habilitation Theses

- [10] J. MAURO. *Constraints meet concurrency*, University of Bologna, 2012.
- [11] C. A. MEZZINA. *Réversibilité dans le pi calcul d'ordre supérieur*, University of Grenoble and University of Bologna (joint degree), 2012.

Articles in International Peer-Reviewed Journals

- [12] P. BAILLOT, U. DAL LAGO, J.-Y. MOYEN. *On quasi-interpretations, blind abstractions and implicit complexity*, in "Mathematical Structures in Computer Science", 2012, vol. 22, n^o 4, p. 549–580, <http://journals.cambridge.org/action/displayAbstract?aid=8635368>.
- [13] M. BRAVETTI, C. DI GIUSTO, J. A. PÉREZ, G. ZAVATTARO. *Adaptable Processes*, in "Logical Methods in Computer Science", 2012, vol. 8, n^o 4:13, p. 1–71.
- [14] S. CAPECCHI, E. GIACHINO, N. YOSHIDA. *Global Escape in Multiparty Sessions*, in "Mathematical Structures in Computer Science", 2012, to appear, <http://www.cs.unibo.it/~giachino/bib/GExc-MSCS.pdf>.
- [15] U. DAL LAGO, M. GABOARDI. *Linear Dependent Types and Relative Completeness*, in "Logical Methods in Computer Science", 2012, vol. 8, n^o 4, <http://arxiv.org/pdf/1104.0193.pdf>.
- [16] U. DAL LAGO, S. MARTINI. *On Constructor Rewrite Systems and the Lambda Calculus*, in "Logical Methods in Computer Science", 2012, vol. 8, n^o 3, <http://arxiv.org/pdf/1208.0515.pdf>.
- [17] U. DAL LAGO, M. ZORZI. *Probabilistic operational semantics for the lambda calculus*, in "RAIRO - Theor. Inf. and Applic.", 2012, vol. 46, n^o 3, p. 413–450, <http://journals.cambridge.org/action/displayAbstract?aid=8661072>.
- [18] C. DI GIUSTO, M. GABBRIELLI, M. C. MEO. *On the Expressive Power of Multiple Heads in CHR*, in "ACM Trans. Comput. Log.", 2012, vol. 13, n^o 1, <http://doi.acm.org/10.1145/2071368.2071374>.
- [19] M. GABOARDI, J.-Y. MARION, S. R. D. ROCCA. *An Implicit Characterization of PSPACE*, in "Transactions on Computational Logic", 2012, vol. 13, n^o 2, 38, <http://doi.acm.org/10.1145/2159531.2159540>.

- [20] M. GABOARDI, M. PICCOLO. *What is a Model for a Semantically Linear Lambda-calculus?*, in "Journal of Logic and Computation", 2012.

Articles in National Peer-Reviewed Journals

- [21] S. MARTINI. *Lingua Universalis*, in "Annali della Pubblica Istruzione", 2012, vol. 4-5, p. 65–70.

International Conferences with Proceedings

- [22] B. ACCATTOLI, U. DAL LAGO. *On the Invariance of the Unitary Cost Model for Head Reduction*, in "23rd International Conference on Rewriting Techniques and Applications (RTA'12)", LIPIcs, 2012, vol. 15, p. 22–37, <http://dx.doi.org/10.4230/LIPIcs.RTA.2012.22>.
- [23] P. BAILLOT, U. DAL LAGO. *Higher-Order Interpretations and Program Complexity*, in "Computer Science Logic (CSL'12) - 26th International Workshop", LIPIcs, 2012, vol. 16, p. 62–76, <http://dx.doi.org/10.4230/LIPIcs.CSL.2012.62>.
- [24] M. BRAVETTI, C. DI GIUSTO, J. A. PÉREZ, G. ZAVATTARO. *Towards the Verification of Adaptable Processes*, in "Leveraging Applications of Formal Methods, Verification and Validation. Technologies for Mastering Change - 5th International Symposium (ISoLA 2012), Proceedings, Part I", Lecture Notes in Computer Science, Springer, 2012, vol. 7609, p. 269–283.
- [25] U. DAL LAGO, B. PETIT. *Linear dependent types in a call-by-value scenario*, in "Principles and Practice of Declarative Programming, (PPDP'12)", ACM, 2012, p. 115–126, <http://doi.acm.org/10.1145/2370776.2370792>.
- [26] U. DAL LAGO, P. PARISEN TOLDIN. *A Higher-Order Characterization of Probabilistic Polynomial Time*, in "Foundational and Practical Aspects of Resource Analysis", R. PENA, M. VAN EEKELEN, O. SHKARAVSKA (editors), Lecture Notes in Computer Science, Springer, 2012, vol. 7177, p. 1–18, 10.1007/978-3-642-32495-6_1, http://dx.doi.org/10.1007/978-3-642-32495-6_1.
- [27] M. DALLA PREDÀ. *The Grand Challenge in Metamorphic Analysis*, in "Information Systems, Technology and Management, 6th International Conference, ICISTM", Communications in Computer and Information Science, Springer, 2012, vol. 285, p. 439-444.
- [28] M. DALLA PREDÀ, W. FENG, R. GIACOBACCI, R. GREECHIE, A. LAKHOTIA. *Twisting Additivity in Program Obfuscation*, in "Information Systems, Technology and Management - 6th International Conference, ICISTM", Communications in Computer and Information Science, Springer, 2012, vol. 285, p. 336-347.
- [29] M. DALLA PREDÀ, M. GABBRIELLI, C. GUIDI, J. MAURO, F. MONTESI. *Interface-Based Service Composition with Aggregation*, in "First European Conference on Service-Oriented and Cloud Computing, ESOC 2012", F. DE PAOLI, E. PIMENTEL, G. ZAVATTARO (editors), Lecture Notes in Computer Science, Springer, 2012, vol. 7592, p. 48–63.
- [30] M. DALLA PREDÀ, M. GABBRIELLI, C. GUIDI, J. MAURO, F. MONTESI. *Service integration via target-transparent mediation*, in "SOCA", 2012, to appear.
- [31] M. DALLA PREDÀ, M. GABBRIELLI, C. GUIDI, J. MAURO, F. MONTESI. *Service integration via target-transparent mediation*, in "5th conference on service-oriented computing and applications, SOCA", 2012, to appear.

- [32] O. DARDHA, E. GIACHINO, D. SANGIORGI. *Session types revisited*, in "Principles and Practice of Declarative Programming, PDP'12", D. DE SCHREYE, G. JANSSENS, A. KING (editors), ACM, 2012, p. 139–150.
- [33] G. DELZANNO, A. SANGNIER, G. ZAVATTARO. *Verification of Ad Hoc Networks with Node and Communication Failures*, in "Formal Techniques for Distributed Systems - Joint 14th IFIP WG 6.1 International Conference (FMOODS 2012) and 32nd IFIP WG 6.1 International Conference (FORTE 2012)", Lecture Notes in Computer Science, Springer, 2012, vol. 7273, p. 235–250.
- [34] R. DEMANGEON, D. HIRSCHKOFF, D. SANGIORGI. *Termination in impure lambda-calculus*, in "Fundamentals of Software Engineering", Lecture Notes in Computer Science, Springer, 2012, vol. 7141, p. 128–142.
- [35] R. DI COSMO, S. ZACCHIROLI, G. ZAVATTARO. *Towards a Formal Component Model for the Cloud*, in "10th International Conference Software Engineering and Formal Methods (SEFM 2012)", Lecture Notes in Computer Science, 2012, vol. 7504, p. 156–171.
- [36] A. DÍAZ-CARO, B. PETIT. *Linearity in the Non-deterministic Call-by-Value Setting*, in "WoLLIC", 2012, p. 216–231, http://dx.doi.org/10.1007/978-3-642-32621-9_16.
- [37] D. HIRSCHKOFF, J.-M. MADIOT, D. SANGIORGI. *Duality and i/o-Types in the pi-calculus*, in "23rd International Conference on Concurrency Theory (CONCUR 2012)", M. KOUTNY, I. ULIDOWSKI (editors), Lecture Notes in Computer Science, Springer, 2012, vol. 7454, p. 302–316, <http://perso.ens-lyon.fr/jeanmarie.madiot/files/concur12.pdf>.
- [38] R. HÄHNLE, M. HELVENSTIJN, E. B. JOHNSEN, M. LIENHARDT, D. SANGIORGI, I. SCHAEFER, P. WONG. *HATS Abstract Behavioral Specification: The Architectural View*, in "Formal Methods for Components and Objects (FMCO) 2011", Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2012, vol. 7542, p. 165–185, <http://proton.inrialpes.fr/~mlienhar/articles/2012%20-%20Hahnle.all%20-%20HATS%20Abstract%20Behavioral%20Specification:%20The%20Architectural%20View.pdf>.
- [39] I. LANESE, C. A. MEZZINA, J.-B. STEFANI. *Controlled Reversibility and Compensations*, in "Proceedings of RC 2012, 4th Workshop on Reversible Computation", R. GLUCK, T. YOKOYAMA (editors), LNCS, Springer Verlag, 2012, vol. 7581, p. 233–240.
- [40] M. LIENHARDT, M. BRAVETTI, D. SANGIORGI. *An Object Group-Based Component Model*, in "Leveraging Applications of Formal Methods, Verification and Validation. Technologies for Mastering Change - 5th International Symposium (ISoLA 2012), Proceedings, Part I", Lecture Notes in Computer Science, Springer, 2012, vol. 7609, p. 64–78, http://dx.doi.org/10.1007/978-3-642-34026-0_6.
- [41] M. LIENHARDT, D. CLARKE. *Conflict Detection in Delta-Oriented Programming*, in "Leveraging Applications of Formal Methods, Verification and Validation. Technologies for Mastering Change - 5th International Symposium (ISoLA 2012), Proceedings, Part I", Lecture Notes in Computer Science, Springer, 2012, vol. 7609, p. 178–192, http://dx.doi.org/10.1007/978-3-642-34026-0_14.
- [42] M. LIENHARDT, D. CLARKE. *Row types for delta-oriented programming*, in "Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems", New York, NY, USA, VaMoS '12, ACM, 2012, p. 121–128 [DOI : 10.1145/2110147.2110161], <http://proton.inrialpes.fr/~mlienhar/articles/2012%20-%20Lienhardt,Clarke%20-%20Row%20Types%20for%20Delta-Oriented%20Programming.pdf>.

- [43] M. LIENHARDT, I. LANESE, C. A. MEZZINA, J.-B. STEFANI. *A Reversible Abstract Machine and Its Space Overhead*, in "FMOODS/FORTE", H. GIESE, G. ROSU (editors), Lecture Notes in Computer Science, Springer, 2012, vol. 7273, p. 1–17, <http://www.cs.unibo.it/~lanese/publications/fulltext/fmoods2012.pdf.gz>.
- [44] B. PETIT. *Continuation Models for the Lambda Calculus With Constructors*, in "Proceedings of the 28th Conference on the Mathematical Foundations of Programming Semantics (MFPS XXVIII)", 2012, vol. 286, p. 337–350, <http://dx.doi.org/10.1016/j.entcs.2012.10.001>.
- [45] F. S. DE BOER, M. BRAVETTI, I. GRABE, M. LEE, M. STEFFEN, G. ZAVATTARO. *A Petri Net based Analysis of Deadlocks for Active Objects and Futures*, in "Proceedings 9th International Symposium on Formal Aspects of Component Software (FACS' 12)", Lecture Notes in Computer Science, Springer, 2012, vol. 7684, p. 110–127.
- [46] F. S. DE BOER, M. M. JAGHOORI, C. LANEVE, G. ZAVATTARO. *Decidability Problems for Actor Systems*, in "23rd International Conference on Concurrency Theory (CONCUR 2012)", Lecture Notes in Computer Science, Springer, 2012, vol. 7454, p. 562–577.

Scientific Books (or Scientific Book chapters)

- [47] D. POUS, D. SANGIORGI. *Enhancements of the bisimulation proof method*, in "Advanced Topics in Bisimulation and Coinduction", D. SANGIORGI, J. RUTTEN (editors), Cambridge University Press, 2012.
- [48] D. SANGIORGI. *Introduction to Bisimulation and Coinduction*, Cambridge University Press, 2012.
- [49] D. SANGIORGI. *Origins of Bisimulation and Coinduction*, in "Advanced Topics in Bisimulation and Coinduction", D. SANGIORGI, J. RUTTEN (editors), Cambridge University Press, 2012.

Books or Proceedings Editing

- [50] M. CARBONE, I. LANESE, A. SILVA, A. SOKOLOVA (editors). *Proceedings of Fifth Interaction and Concurrency Experience (ICE 2012)*, Electronic Proceedings in Theoretical Computer Science, 2012, vol. 104.
- [51] D. SANGIORGI, J. RUTTEN (editors). *Advanced Topics in Bisimulation and Coinduction*, Cambridge University Press, 2012.

Scientific Popularization

- [52] S. MARTINI. *Lingua Universalis*, in "Annali della Pubblica Istruzione", 2012, vol. 4-5, p. 65–70, http://www.annalipubblicaistruzione.it/riviste/annali/pdf/api4_5_2011.pdf.

References in notes

- [53] 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, p. 205–225.
- [54] K. POHL, G. BÖCKLE, F. V. DER LINDEN. *Software Product Line Engineering: Foundations, Principles, and Techniques Objects*, Springer Verlag, 2005.