



IN PARTNERSHIP WITH:  
**Université de Bologne (Italie)**

Activity Report 2013

## **Project-Team FOCUS**

# Foundations of Component-based Ubiquitous Systems

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

RESEARCH CENTER  
**Sophia Antipolis - Méditerranée**

THEME  
**Distributed programming and Software engineering**



## Table of contents

<b>1. Members</b>	<b>1</b>
<b>2. Overall Objectives</b>	<b>2</b>
2.1. Overall Objectives	2
2.2. Highlights of the Year	2
<b>3. Research Program</b>	<b>2</b>
<b>4. Application Domains</b>	<b>3</b>
4.1. Ubiquitous Systems	3
4.2. Service Oriented Computing and Cloud Computing	3
4.3. Software Product Lines	3
<b>5. Software and Platforms</b>	<b>4</b>
5.1. Jolie	4
5.2. Others	4
<b>6. New Results</b>	<b>6</b>
6.1. Service-oriented computing	6
6.1.1. Primitives	6
6.1.2. Choreographies	6
6.2. Models for reliability	6
6.2.1. Reversibility	7
6.2.2. Compensations	7
6.3. Cloud Computing	7
6.3.1. Languages for cloud applications	7
6.3.2. Models for cloud application deployment	7
6.4. Resource Control	7
6.4.1. Expressive type systems for complexity analysis	8
6.4.2. Complexity analysis and process algebras	8
6.4.3. Characterizing probabilistic complexity classes	8
6.4.4. Ensuring differential privacy	8
6.5. Verification of extensional properties	8
6.5.1. Type systems for objects and components	9
6.5.2. Deadlock analysis	9
6.6. Expressiveness of computational models	9
<b>7. Partnerships and Cooperations</b>	<b>10</b>
7.1. National Initiatives	10
7.2. European Initiatives	11
7.2.1. FP7 Projects	11
7.2.2. Collaborations in European Programs, except FP7	11
7.2.3. Collaborations with Major European Organizations	11
7.3. International Initiatives	12
7.4. International Research Visitors	12
7.4.1. Visits of International Scientists	12
7.4.2. Visits to International Teams	13
<b>8. Dissemination</b>	<b>13</b>
8.1. Scientific Animation	13
8.2. Teaching - Supervision - Juries	14
8.2.1. Teaching	14
8.2.2. Supervision	16
8.2.3. Juries	16
8.3. Popularization	16
<b>9. Bibliography</b>	<b>17</b>



## 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]

#### External Collaborators

Daniel Hirschkoﬀ [ENS Lyon]

Fabrizio Montesi [IT University of Copenhagen]

Marco Pistore [Gouvernement étranger, until Dec 2013]

#### 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]

Paolo Parisen Toldin [University Bologna, until March 2013]

Giulio Pellitta [University Bologna]

Marco Solieri [Université Paris 13]

#### Post-Doctoral Fellows

Mila Dalla Preda [University Bologna, until March 2013]

Marco Gaboardi [Universities of Bologna and Philadelphia, until October 2013]

Claudio Guidi [University Bologna, EIT project]

Jacopo Mauro [Inria Sophia, ANR AEOLUS project]

Michael Lienhardt [University Bologna, Envisage project]

#### Visiting Scientist

Xian Xu [East China University of Science and Technology, Shanghai, until May 2013]

#### Administrative Assistant

Christine Riehl

#### Other

Valeria Vignudelli [University Bologna, trainee, until Nov 2013]

## 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 "Constraints Meet Concurrency" has won the 2013 award for Best Italian PhD Theses in Theoretical Computer Science (track Logics, Semantics and Programming Theory). The prize is awarded by the Italian Chapter of EATCS.
- Davide Sangiorgi has received the "LICS Test-of-Time" award (this award recognizes a paper from the proceedings of the conference "Logics in Computer Science" from 20 years prior that has best met the "test of time" in term of impact).
- the EU project Envisage (FP7), on the theme of cloud computing, has been approved and has started its activities.

## 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  $\pi$ -calculus) and  $\lambda$ -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,  $\lambda$ -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 [51]. 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 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.

In 2013 the development of the Jolie language has mainly focused on tools for the programming environment and on the integration of the language with cloud infrastructure. More in detail, we have produced the following software.

- PaaSOSA. This is a prototype for the deployment of Jolie services in a cloud infrastructure. We have also worked on the integration of Jolie with the Drools engine. Drools is used in PaaSOSA for storing and managing monitor events from services.
- JEeye. This is web GUI prototype for designing Jolie graphical workflows which can be deployed into PaaSOSA.
- WSOA. We have developed a SaaS (Software as a Service) layer for the publication of Jolie APIs using different formats (http, JSON, SOAP, XML, and so on). We have also enhanced some libraries for the integration between Jolie and GWT technology.

All the server side code of PaaSOSA, JEeye and WSOA has been developed by using Jolie.

### 5.2. Others

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

- *Deadlock analysis* 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).

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 behavioral descriptions of methods, called contracts, which retain resource dependency information. Then this algorithm is integrated with a number of possible different back-ends that analyze contracts and derive deadlock information. We have prototyped two such back-ends:

1. an evaluator that computes a fixpoint semantics, and
2. an evaluator using abstract model checking.

The evaluator (1) is available at <http://www.cs.unibo.it/~laneve/deadlock/index.html>

The evaluator (2) is available at <http://www.cs.unibo.it/~giachino/siteDat/index.php>

- *CaReDeb* (<http://proton.inrialpes.fr/~mezzina/deb/>).



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 misbehavior, 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.

- *AIOCI* (<http://www.cs.unibo.it/projects/jolie/aioej.html>).

AIOCI is a framework for programming adaptive distributed systems based on message passing. AIOCI comes as a plugin for Eclipse, AIOCI-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.

- *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. Experimental results are encouraging as METIS looks quite effective in practice by handling problem instances with hundreds of components in less than a minute. This is a key ingredient in the solution to the automation problem addressed by the Aeolus project. The paper [48] is dedicated to the description of the tool, while [41] addresses the formal aspects of the technique.

The software below have not undergone substantial modifications during 2013

- *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.

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.

- *IntML* is a functional programming language guaranteeing sublinear space bounds for all programs [50]. 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.

## 6. New Results

### 6.1. Service-oriented computing

**Participants:** Mario Bravetti, Ivan Lanese, Fabrizio Montesi, Gianluigi Zavattaro.

#### 6.1.1. Primitives

We have studied primitives used in the context of service-oriented computing, at different levels of abstraction and in different contexts. At the abstract level, we considered both standard web services and Internet of Things, where computational and communication capabilities are attached to real-world objects such as smartphones or alarm clocks. For web services, we defined SSCC (Stream-based Service-Centered Calculus) [17], a calculus allowing to describe both service composition (orchestration) via streams and the protocols that services follow when invoked (conversation). We assessed the expressive power of SSCC by modeling van der Aalst's workflow patterns and an automotive case study from the European project Sensoria. For analysis, we presented a simple type system ensuring compatibility of client and service protocols. We also studied the behavioral theory of the calculus, highlighting some axioms that capture the behavior of the different primitives. As a final application of the theory, we defined and proved correct some program transformations. For Internet of Things, a main contribution [37] has been the definition of a calculus and of an equivalence allowing to capture the behavior of the system as seen by the human end-user. Since this equivalence is not compositional we defined also a finer equivalence which is compositional. We showed how our equivalences can be applied to reason on simple Internet of Things examples.

At a more concrete level, we have continued to study and extend the Jolie language. In [44] we present a detailed description of the Jolie language. We put our emphasis on how Jolie can deal with heterogeneous services. On the one hand, Jolie combines computation and composition primitives in an intuitive and concise syntax. On the other hand, the behavior and deployment of a Jolie program are orthogonal: they can be independently defined and recombined as long as they have compatible typing. In [42] we extended Jolie to model process-aware web information systems. Our major contribution is to offer a unifying approach for the programming of distributed architectures based on HTTP that support typical features of the process-oriented paradigm, such as structured communication flows and multiparty sessions.

#### 6.1.2. 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 are deadlock-free by construction. From a choreography one can automatically derive the behavior of each participant using a notion of projection. Under suitable conditions on the structure of a choreography, the correctness of its projection can be established in terms of a trace-based semantics. In [24] we have proposed a purely-global programming model. The idea is to define abstract choreographies – called protocol specifications – and use them to type a more concrete choreography. This more concrete choreography is used to generate executable code for the different participants. The approach is based on a novel interpretation of asynchrony and parallelism. We evaluated the approach by providing a prototype implementation for a concrete programming language and by applying it to some examples from multicore and service-oriented programming [49]. In [43] we tackled one of the main limitations of choreographies, namely the fact that they model closed systems. To this end we proposed a notion of composable choreographies. The key of our approach is the introduction of partial choreographies, which can mix global descriptions with communications among external peers. We prove that if two choreographies are composable, then the endpoints independently generated from each choreography are also composable, preserving their typability and deadlock-freedom. In [39] we showed how to transform choreographies which do not satisfy the conditions for their projection into choreographies that satisfy them preserving their behavior and enabling a correct projection.

### 6.2. Models for reliability

**Participants:** Ivan Lanese, Michael Lienhardt, 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. In [38] we present *croll-pi*, a concurrent calculus extending *roll-pi* – an higher-order pi-calculus featuring a rollback operator – allowing the specification of alternatives to a computation to be used upon rollback. Alternatives in *croll-pi* are attached to messages. We show the robustness of this mechanism by encoding more complex idioms for specifying alternatives. We illustrate the expressiveness of our approach by encoding a calculus of communicating transactions and by modeling the 8-queens problem. We also formally prove that *croll-pi* is strictly more expressive than *roll-pi*.

### 6.2.2. Compensations

We have continued the study of the expressive power of primitives for specifying compensations in long running transactions. Dynamic compensation installation allows for easier specification of fault handling in complex interactive systems since it enables to update the compensation policies according to run-time information. In [40] we show that in a simple pi-like calculus with static compensations the termination of a process is decidable, but it is undecidable in one with dynamic compensations. We then consider three commonly used patterns for dynamic compensations: parallel compensations, where new compensation items can only be added in parallel, replacing compensations, where old compensations are replaced, and nested compensations, where old compensations can be used (linearly) to build new ones. We show that termination is decidable in the first two cases and undecidable in the last one.

## 6.3. Cloud Computing

**Participants:** Elena Giachino, Michael Lienhardt, Tudor Alexandru Lascu, Jacopo Mauro, Gianluigi Zavattaro.

### 6.3.1. Languages for cloud applications

To foster the industrial adoption of virtualized services, it is necessary to address two important problems: (1) the efficient analysis, dynamic composition of services with qualitative and quantitative service levels and (2) the dynamic control of resources such as storage and processing capacities according to the internal policies of the services. Current technologies for cloud computing, addresses these problems at deployment and run time. The ENVISAGE project and the position paper [20] proposes, on the contrary, to overcome these problems by leveraging service-level agreements into software models and resource management into early phases of service design.

### 6.3.2. Models for cloud application deployment

Cloud computing offers the possibility to build sophisticated software systems on virtualized infrastructures at a fraction of the cost necessary just few years ago, but deploying/maintaining/reconfiguring such software systems is a serious challenge. The AEOLUS project, aims to tackle the scientific problems that need to be solved in order to ease the problem of efficient and cost-effective deployment and administration of the complex distributed architectures which are at the heart of cloud applications [25]. In particular, it is necessary to define appropriate models for the representation of the interdependencies among the software components of a cloud application as well as declarative languages for the specification of the desired application configuration. We have proposed [31] a model for the representation of the component lifecycle and of its dependencies/conflicts with the other components. Based on such model, we have defined a sound and complete algorithm that efficiently computes a deployment plan (i.e. a sequence of low-level component deployment actions) capable of reaching a final configuration including at least some predefined basic components [48] and we have realized a prototypical implementation of such algorithm which was proved to be effective on case-studies of realistic size (i.e. hundreds of components) [41].

## 6.4. Resource Control

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

### 6.4.1. Expressive type systems for complexity analysis

Along 2013, our work on expressive methodologies for complexity analysis of higher-order languages has proceeded. In particular, we have focused our attention on extending linear dependent types to languages with control operators in the style of `callcc` [27]. This has taken the form of a generalization of bounded linear logic towards Laurent’s polarized linear logic, which is then turned into a type system for the lambda-mu-calculus (in which the aforementioned control operator can indeed be implemented). In the introduced type system, all typable terms can be reduced in polynomial time. We also worked on the linear dependent type inference and on its implementation (though the work has not yet been transferred onto the *Lideal* tool implementing type inference for dependently linear type systems, see <http://lideal.cs.unibo.it/>); more specifically, we showed that type inference can in this context be reduced to a form of constraint amenable to be solved by SMT solvers [28]. Finally, a call-by-value version of  $d\ell$ PCF has been defined and proved sound but also relatively complete as a tool for complexity analysis of programs [16].

### 6.4.2. Complexity analysis and process algebras

Complexity analysis methodologies drawn from linear logic have been adapted to higher-order process algebras, obtaining linear versions of the higher-order  $\pi$ -calculus in which reduction sequences are guaranteed to have a length bounded by a polynomial [14]. This is done by following the exponential discipline Lafont’s Soft Linear logic suggests.

### 6.4.3. Characterizing probabilistic complexity classes

We have also been looking [10] (papers extracted from the thesis should appear soon) at probabilistic computation and at whether probabilistic complexity classes like **BPP**, **ZPP** and **PP** can be characterized by logics and  $\lambda$ -calculi. We encountered some problems in doing the above for **BPP** and **ZPP**, which are semantic classes and which, as a consequence, cannot be easily enumerated (and captured by ICC systems). On the other hand, probabilistic classes like **PP** can indeed be characterized by  $\lambda$ -calculi, as shown by our recent work on RSLR, a system derived from Hofmann’s SLR that captures the (deterministic) polytime computable functions.

### 6.4.4. Ensuring differential privacy

Differential privacy offers a strong guaranteed bound on loss of private information due to release of query results, even under worst-case assumptions. One of the challenges in proving queries differentially private is to prove an upper bound on the query’s sensitivity, i.e., the maximum change in the query’s output that can result from changing the data of a single individual. Reed and Pierce have recently proposed a type analysis using numerical annotations in types to describe bounds on the sensitivity of the queries. A first delicate aspect of this approach is that in order to verify if a program is typable or not one needs to come up with numerical annotations and verify their consistency. Finding a “small” annotation is crucial, since the privacy depends on it. For this reason we designed a sensitivity inference tool [26] that combined with the Z3 SMT solver is able to verify and infer a minimal sensitivity bound in an automatic and efficient way. Another delicate aspect of this approach is the *expressivity* of the type analysis. Reed and Pierce’s type system offers only a very limited form of numerical annotations. These numerical annotations are not enough to provide a bound for programs whose sensitivity depends on data available only at runtime. To recover this problem we introduced *Dfuzz* [32], a language combining linear types and lightweight dependent types.

## 6.5. Verification of extensional properties

**Participants:** Ornella Dardha, Elena Giachino, Michael Lienhardt, Cosimo Laneve, Fabrizio Montesi.

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. Most work carried out this year has to do with type systems for concurrent objects and components ensuring safe and reliable interactions, and on deadlock analysis for systems of concurrent objects or within process sessions.

### 6.5.1. Type systems for objects and components

In previous work, we had developed an integration of session types, for specifying and validating structured communication sequences (sessions) into a class-based core object language for building network applications. We have defined [12] a constraint-based type system that reconstructs the appropriate session types of session declarations instead of assuming that session types are explicitly given by the programmer, and used static analysis via types to ensure that, once a session has started, computation cannot get stuck on a communication deadlock.

In previous papers, we had proposed a component layer for object-oriented language ABS (studied in the EU project Hats), that allows one to perform updates on objects by means of communication ports and their rebinding. We have now [29] introduced a type system for this component model that statically enforces that no object will attempt illegal rebinding.

### 6.5.2. Deadlock analysis

Deadlock represents an insidious and recurring threat when systems also exhibit a high degree of resource and data sharing. We address deadlock analysis of two such systems: (1) concurrent object-oriented languages; (2) protocol specifications.

For (1), we have developed a framework for statically detecting deadlocks in a concurrent object-oriented language with asynchronous method calls and cooperative scheduling of method activations. 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 behavioral descriptions of methods, called contracts, which retain resource dependency information [33]. This component is integrated with a number of possible different back-ends that analyze contracts and derive deadlock information. As a proof-of-concept, we discuss two such back-ends: (i) an evaluator that computes a fixpoint semantics [33] and (ii) an evaluator using abstract model checking [34].

For (2), in [24], we develop a typing discipline that verifies choreographies against protocol specifications, based on multiparty sessions. Exploiting the nature of global descriptions, our type system defines a new class of deadlock-free concurrent systems (deadlock-freedom-by-design), provides type inference, and supports session mobility. We give a notion of Endpoint Projection (EPP) which generates correct entity code (as pi-calculus terms) from a choreography. Finally, we evaluate our approach by providing a prototype implementation for a concrete programming language and by applying it to some examples from multicore and service-oriented programming.

Finally, en passant we remind [23], that studies deadlock analysis of concurrent object-oriented languages via encoding into Petri nets, which had already been discussed in last year's report.

## 6.6. Expressiveness of computational models

**Participants:** Roberto Amadini, Ornela Dardha, Maurizio Gabbrielli, Daniel Hirschhoff, Jean-Marie Madiot, Jacopo Mauro, Davide Sangiorgi, Gianluigi Zavattaro.

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

The fusion calculi are a simplification of the  $\pi$ -calculus in which input and output are symmetric and restriction is the only binder. We show [35] a major difference between these calculi and the  $\pi$ -calculus from the point of view of types, proving some impossibility results for subtyping in fusion calculi. We propose a modification of fusion calculi in which the name equivalences produced by fusions are replaced by name preorders, so to be able to import subtype systems, and related results, from the  $\pi$ -calculus. We have studied the consequences of the modification on behavioural equivalence and expressiveness.

In Focus we use notions of constraint to define in a succinct way models of computation and current constraint solving technologies to solve problems modeled using constraints. For this reason we have studied the expressive power of various computational models involving constraints and their practical impact in terms of solving/execution performances. In [18] we have investigated how the notion of constraint augments the expressive power of a concurrent language if priorities are introduced. The chosen language is Constraint Handling Rules, a committed-choice declarative language originally designed for writing constraint solvers and that is nowadays a general purpose language. The result has been obtained by first formalising the meaning of language encodings and language embedding, widely used in concurrency theory. Different ways to model and define disaster scenarios are analyzed and compared in [11], where we study a model expressive enough to define a disaster scenario that, at the same time, can be used to find plans to save the victims of a disaster using modern constraint solving technology. Similarly, different computation models are considered in [22] where we study how machine learning techniques can be used to boost the performances of constraint solvers. A technique dubbed “portfolio approach” is used to combine the different performances of constraint solvers to obtain a globally better solver using, as a starting point, a simple low-level constraint language.

In [30] we propose an integration of structural sub-typing with boolean connectives and semantic sub-typing to define a Java-like programming language that exploits the benefits of both techniques. The resulting language has a more expressive set of types that comes from the use of boolean constructs, negation types, and the integration of structural and nominal sub-typing in an object-oriented setting. By implementing traditional Java-language constructs we show that the proposed language is also expressive enough w.r.t. the Java language.

## 7. Partnerships and Cooperations

### 7.1. National Initiatives

- AEOLUS (Mastering the Cloud Complexity) is an ANR-ARPEGE project started on December 2010 that will finish 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, Lascu, Mauro, Sangiorgi, 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.
- The ANR project PACE (Processus non-standard: Analyse, Coinduction, et Expressivité) has 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: Hirschhoff (project coordinator), Dal Lago, Lanese, Sangiorgi, Zavattaro.

## 7.2. European Initiatives

### 7.2.1. FP7 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.
- 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 correspond to exhibiting a certification of its resource consumption.

Gaboardi has been in Philadelphia till July 2013, and then in Bologna till October 2013. (He has then moved to Dundee, where he has obtained a tenure-track position.)

### 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, Mauro, Sangiorgi, Zavattaro.
- In the context of the EIT activity "HCI Technologies for the Digital World", funded with 23000 euro, we have worked to the technical part of the EIT ICT Labs Summer School, Intelligent Services for Digital Cities (ISDC-2013), which has been held in Trento in September 2013 (<http://www.trentorise.eu/education/intelligent-services-digital-cities-isdc-2013-summer-school>).

In particular we developed a SaaS component, called WSOA, to our cloud oriented framework JSOA. WSOA can publish a set of APIs for external usage which are selected from those deployed into the PaaS layer (PaaSOSA). APIs are collected by exploiting the aggregation mechanism of Jolie and then deployed into a Jolie web server (Leonardo). They can be published by defining different protocols/formats such as http/soap, http/json, http/POST/xml, and so on. Thanks to WSOA we also developed a plugin for the SATIN project where the published APIs can be automatically imported into SATIN console and used for creating web based mobile applications.

To provide a comprehensive tool for the students of the smart city summer school, we developed a wrapper for the CKAN platform in order to import all the CKAN APIs into PaaSOSA. In such a way, it is now possible to perform calls to the CKAN platform by exploiting Jolie services deployed into PaaSOSA.

Main persons involved: Gabbrielli, Guidi.

### 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. 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.
- 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. 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: Lascau, 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, Gaboardi. Some visits during 2013.
- 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. Some visits during 2003.
- 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 Initiatives

### 7.3.1. Inria International Partners

#### 7.3.1.1. Informal 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 the sabbatical year of Xian Xu, from East China University of Science and Technology, Shanghai (paid a scholarship from the Chinese Science Foundation).

- Xian Xu has completed in May a year sabbatical in Focus. He is lecturer at the East China University of Science and Technology in Shanghai.
- Marco Bernardo, Universita' Urbino. A couple of 2 or 3 day visits during the year. Topic: probabilistic process models.
- Simon Gay, University Glasgow. A 1-month visit, March-April. Topic: session types.
- Andrei Dorman, Paris 13. One 3-day visit in May. Topic: Concurrent Interaction Nets and Graph Rewriting
- Patrick Baillot, ENS Lyon, and Gilles Barthe, IMDEA Madrid. One joint 4-day visit in May. Topic: Security and Cryptography



- Matthew Hennessy, University College Dublin. One 3-day visit in June. Topic: probabilistic bisimulations.
- Ramyaa Ramyaa, LMU Munich. One 1-week visit in June.
- Lionel Vaux and Emmanuel Beffara, Marseille. One 3-day visit in October. Topic: Linear Logic.
- Ludovic Henrio, Inria Sophia Antipolis. One 2-day visit in December. Topic: deadlock-analysis and asynchronous object-oriented languages.
- Stefano Zacchiroli, from Paris 7, has visited us various times for a few days during the year.

#### 7.4.2. Visits to International Teams

We only report visits that were longer than 1 month.

- Roberto Amadini: 6 months at the Optimisation Research Group of NICTA in Melbourne (<http://optimisation.nicta.com.au/>).
- Ornela Dardha: 1 year (1 November 2012 - 1 November 2013) at IT University of Copenhagen, Denmark, visiting Prof. Marco Carbone (topic: type systems for processes)
- Giulio Pellitta: 6 months at Laboratoire d'Informatique, Université Paris Nord (November 2012 - May 2013). Topic: abstract machines for resource control in the  $\lambda$ -calculus.
- Tudor Alexandru Lascu: 3 months at Paris 7 (PPS team). Topic: cloud computing.

## 8. Dissemination

### 8.1. Scientific Animation

**M. Bravetti:** Program Committee co-chair for: SEFM 2013 11th International Conference on Software Engineering and Formal Methods.

Program Committee member of the following conferences: SITIS 2013 (WeCA): 9th Conference on Signal-Image Technology and Internet-Based Systems (Web Computing and Application track); CLOUD 2013: 6th IEEE Conference on Cloud Computing; WS-FM 2013: 10th Workshop on Web Services and Formal Methods: Formal Aspects of Service-Oriented and Cloud Computing; ESOC 2013: 2nd European Conference on Service-Oriented and Cloud Computing; FSEN 2013: 5th Conference on Fundamentals of Software Engineering; WS-FMDS 2013: 3rd Workshop on Formal Methods in the Development of Software.

**O. Dardha:** Program Committee member 6th Interaction and Concurrency Experience (ICE 2013)

**U. Dal Lago:** Program Committee co-chair and main organiser 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 following conferences: FOSSACS 2013: 16th Conference on Foundations of Software Science and Computation Structures; CSL 2013: Computer Science Logic 2013; FSTTCS 2013: Foundations of Software Technology and Theoretical Computer Science; LOLA 2013: Syntax and Semantics of Low-Level Languages; GaLoP VIII: Games for Logic and Programming Languages.

**I. Lanese:** Program Committee co-chair of the following conferences: ICE 2013: 6th Interaction and Concurrency Experience; Service Oriented Architectures and Programming track of the 28th Annual ACM Symposium on Applied Computing (SAC 2013).

Program Committee member of 9th Workshop on Automated Specification and Verification of Web Systems (WWV 2013).

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

**C. Laneve:** Invited speaker at IFM 2013, 10th International Conference on Integrated Formal Methods

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

**S. Martini:** Program Committee member of the following conferences: 14th Italian Conference on Theoretical Computer Science (ICTCS 2013); 8th International Computer Science Symposium in Russia (CSR 2013); Workshop Logic and Computational Complexity (LCC 13); conference on the History and Philosophy of Computing (HaPoC 2013).

**F. Montesi:** Track program chair for 27th ACM Symposium on Applied Computing (SAC) track on Service-Oriented Applications and Programming track (SOAP).

Program Committee member of 2013 World Conference on Information Systems and Technologies (WorldCIST).

**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.

Invited speaker at the events: the SIGPLAN Programming Languages Mentoring Workshop; 6th Interaction and Concurrency Experience (ICE 2013)

Program Committee member of the conferences: 7th Conference on Language and Automata Theory and Applications (LATA 2013); 5th IPM International Conference on Fundamentals of Software Engineering (FSEN 2013); 24th International Conference on Concurrency Theory (CONCUR); Fourth International Symposium on Games, Automata, Logics and Formal Verification (GandALF)

Lecturer at the schools: 18th Estonian Winter School in Computer Science (EWSCS), Tallin, February 2013, and Scuola Estiva di Logica 2013, Garignano sul Garda, August 2013.

**G. Zavattaro:** Invited speaker at the 2013 European Conference on Service-oriented and Cloud Computing (ESOCC'13).

Program Committee member of the following conferences: 3rd Workshop on Adaptive Services for the Future Internet (WAS4FI'13); 10th Workshop on Web Services and Formal Methods: Formal Aspects of Service-Oriented and Cloud Computing (WS-FM'13); 7th Workshop on Reachability Problems (RP'13); European Conference on Service-oriented and Cloud Computing (ESOCC'13); 2nd Workshop on Quality Assurance for Service-based applications (QASBA 2013); 6th IEEE Conference on Cloud Computing (CLOUD 2013); IFIP Joint International Conference on Formal Techniques for Distributed Systems (33rd FORTE / 15th FMOODS); 10th Symposium on Formal Aspects of Component Software (FACS'13); the 10th Conference on Integrating Formal Methods (iFM'13).

## 8.2. Teaching - Supervision - Juries

### 8.2.1. Teaching

Below are the teaching activities of the Focus members during 2013.

- 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.
- 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.
  - Undergraduate: “Ingegneria del Software”, 20 hours, 2nd year, University of Bologna, Italy.
  - PhD: “Reversible Computing”, 10 hours, Department of Mathematics and Computer Science, University of Cagliari, 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: “Basic Complexity”, 12 hours, 1st year and 2nd year, University of Bologna, Italy.
  - PhD: “Principles of Computer Science: Computing, Machines, and Programming Languages” Joint International Doctoral (Ph.D.) Degree in Law, Science and Technology, University of Torino, 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
  - Undergraduate: “Introduction aux structures de données linéaires”, 6h hours, 1st year, Institut Galilé, Université Paris 13, France
  - Master: “Administration Système”, 15 hours, 2nd year, école d’ingénieurs, Institut Galilé, Université Paris 13, France
  - Master: “Intranet, Internet et Web”, 44 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.
- Roberto Amadini
  - 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.
- Elena Giachino
  - Undergraduate: “Programmazione”, 40 hours, 1st year, University of Bologna, Italy.

### 8.2.2. Supervision

Phd thesis completed in 2013:

- Paolo Parisen Toldin, Implicit computational complexity and probabilistic classes, Dottorato di Ricerca in Informatica, University of Bologna, April 2013 Supervisors: 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.

- Michele Alberti, July 2011; Separation results in the algebraic lambda-calculus; L. Regnier (Marseille) and S. Martini
- 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.
- Ornela Dardha, January 2011, Type Systems for Distributed Programs: Components and Sessions, D. Sangiorgi.
- Saverio Giallorenzo, January 2013, Adaptation and choreography languages, M. Gabbrielli.
- Tudor Alexandru Lascu, January 2011, Automatic planning of the deployment of cloud-based applications, G. Zavattaro
- Jean-Marie Madiot, September 2011, Types, proofs and proof techniques for process calculi, D. Hirschhoff and D. Sangiorgi.
- Giulio Pellitta, January 2011, Extending Implicit Computational Complexity and Abstract Machines to Languages with Control, S. Martini.
- Alessandro Rioli, January 2012; Bisimulation equivalence in quantum lambda calculi, U. Dal Lago.
- Marco Solieri, December 2011; Géométrie de l'interaction, réseaux, et lambda-calcul, S. Guerrini (Paris Nord) and S. Martini.

The following PhD student is supervised by a member of Focus, but currently not in the Focus team:

- Sara Zuppiroli (Bologna), January 2010; models for quantum computing; U. Dal Lago.

### 8.2.3. Juries

U. Dal Lago has been rapporteur and member of the PhD evaluation committee for Clement Aubert, Paris 13, Stephane Zimmermann, Paris 7, and Martin Avanzini, University of Innsbruck.

S. Martini has been rapporteur and member of the evaluation committee of the "Habilitation a Diriger des Recherches" (HDR) of Michele Pagani, Paris 13, December 2013.

G. Zavattaro has been rapporteur and member of the PhD evaluation committee for Huu Nghia Nguyen (supervisors Prof. Pascal Poizat and Fatiha Zaidi), Université Paris Sud, Paris, France; and he has been member of the Ph.D. evaluation committee of Luca Panziera (supervisor Prof. Flavio De Paoli), universita' Milano Bicocca, Milano, 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>;
- the talks "Calcolo e simboli: la scienza digitale. A.M. Turing a cent'anni dalla nascita", Accademia di Agricoltura Scienze e Lettere di Verona; "Problem solving tra linguaggio e competizione", Workshop 'Olimpiadi e Informatica', Verona, 22 novembre 2013.

In addition, **S. Martini** and **U. Dal Lago** gave a series of lectures (20 hours altogether) to the students of "Collegio Superiore", a college of excellence at the University of Bologna (general students, not limited to Computer Science). The course was about algorithms, computational complexity, and cryptography.

## 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<sup>o</sup> 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<sup>o</sup> 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<sup>o</sup> 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<sup>o</sup> 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] P. PARISEN TOLDIN. , *ICC and Probabilistic Classes*, Università degli studi di Bologna, April 2013, <http://hal.inria.fr/tel-00909410>

#### Articles in International Peer-Reviewed Journals

- [11] R. AMADINI, I. SEFRIQUI, J. MAURO, M. GABBRIELLI. *A Constraint-Based Model for Fast Post-Disaster Emergency Vehicle Routing*, in "international Journal of Interactive Multimedia and Artificial Intelligence", 2013, vol. 2, n<sup>o</sup> 4, pp. 67-75, <http://hal.inria.fr/hal-00909296>

- [12] L. BETTINI, S. CAPECCHI, M. DEZANI-CIANCAGLINI, E. GIACHINO, B. VENNERI. *Deriving session and union types for objects*, in "Mathematical Structures in Computer Science", 2013, vol. 23, pp. 1163–1219, <http://hal.inria.fr/hal-00909312>
- [13] S. CAPECCHI, E. GIACHINO, N. YOSHIDA. *Global Escape in Multiparty Sessions*, in "Mathematical Structures in Computer Science", 2013, to appear, <http://hal.inria.fr/hal-00909314>
- [14] U. DAL LAGO, S. MARTINI, D. SANGIORGI. *Light Logics and Higher-Order Processes*, in "Mathematical Structures in Computer Science", 2013, to appear, <http://hal.inria.fr/hal-00909315>
- [15] U. DAL LAGO, P. PARISEN TOLDIN. *An Higher-Order Characterization of Probabilistic Polynomial Time*, in "Information and Computation", 2013, to appear, <http://hal.inria.fr/hal-00909316>
- [16] U. DAL LAGO, B. PETIT. *Linear Dependent Types in a Call-by-Value Scenario*, in "Science of Computer Programming", 2013, to appear [DOI : 10.1016/J.SCICO.2013.07.010], <http://hal.inria.fr/hal-00909317>
- [17] L. C. FILIPE, I. LANESE, F. MARTINS, A. RAVARA, V. VASCONCELOS. *The Stream-based Service-Centered Calculus: a Foundation for Service-Oriented Programming*, in "Formal Aspects of Computing", 2013, 60 p., To appear, <http://hal.inria.fr/hal-00909308>
- [18] M. GABBRIELLI, J. MAURO, M. C. MEO. *The expressive power of CHR with priorities*, in "Information and Computation", 2013, vol. 228-229, pp. 62-82 [DOI : 10.1016/J.IC.2013.05.001], <http://hal.inria.fr/hal-00909295>
- [19] C. VERSARI, G. ZAVATTARO. *Complex Functional Rates in the Modeling of Nano Devices*, in "Electronic Notes in Theoretical Computer Science", March 2013, vol. 293, pp. 3-15, <http://hal.inria.fr/hal-00825147>

### International Conferences with Proceedings

- [20] E. ALBERT, F. BOER, R. HAHNLE, E. B. JOHNSEN, C. LANEVE. *Engineering Virtualized Services*, in "Proc. 2nd Nordic Symposium on Cloud Computing and Internet Technologies (NordiCloud 2013)", Oslo, Norway, ACM, 2013, pp. 59–63 [DOI : 10.1145/2513534.2513545], <http://hal.inria.fr/hal-00909294>
- [21] M. ALBERTI. *Normal Forms for the Algebraic Lambda-Calculus*, in "JFLA - Journées francophones des langages applicatifs", Aussois, France, D. POUS, C. TASSON (editors), Damien Pous and Christine Tasson, February 2013, <http://hal.inria.fr/hal-00779911>
- [22] R. AMADINI, M. GABBRIELLI, J. MAURO. *An Empirical Evaluation of Portfolios Approaches for Solving CSPs*, in "CPAIOR - 10th International Conference on the Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming - 2013", Yorktown Heights, United States, C. GOMES, M. SELLMANN (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 7874, pp. 316-324 [DOI : 10.1007/978-3-642-38171-3\_21], <http://hal.inria.fr/hal-00909297>
- [23] F. S. BOER, M. BRAVETTI, I. GRABE, M. D. LEE, M. STEFFEN, G. ZAVATTARO. *A Petri Net Based Analysis of Deadlocks for Active Objects and Futures*, in "Formal Aspects of Component Software, 9th International Symposium, FACS 2012", Mountain View, United States, C. S. PASAREANU, G. SALAÜN (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 7684, pp. 110-127, <http://hal.inria.fr/hal-00909292>

- [24] M. CARBONE, F. MONTESI. *Deadlock-freedom-by-design: multiparty asynchronous global programming*, in "POPL - 40th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages - 2013", Rome, Italy, ACM, 2013, pp. 263-274, <http://hal.inria.fr/hal-00909320>
- [25] M. CATAN, R. D. COSMO, A. EICHE, T. A. LASCU, M. LIENHARDT, J. MAURO, R. TREINEN, S. ZACCHIROLI, G. ZAVATTARO, J. ZWOLAKOWSKI. *Aeolus: Mastering the Complexity of Cloud Application Deployment*, in "ESOCC - European Conference on Service-Oriented and Cloud Computing - 2013", Malaga, Spain, K.-K. LAU, W. LAMERSDORF, E. PIMENTEL (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 8135, pp. 1-3 [DOI : 10.1007/978-3-642-40651-5\_1], <http://hal.inria.fr/hal-00909298>
- [26] L. D'ANTONI, M. GABOARDI, A. E. J. GALLEGRO, A. HAEBERLEN, B. PIERCE. *Sensitivity analysis using type-based constraints*, in "FPCDSL - 1st annual workshop on Functional programming concepts in domain-specific languages - 2013", Boston, BA, USA, United States, ACM, 2013, pp. 43-50 [DOI : 10.1145/2505351.2505353], <http://hal.inria.fr/hal-00909341>
- [27] U. DAL LAGO, G. PELLITTA. *Complexity Analysis in Presence of Control Operators and Higher-Order Functions*, in "LPAR-19 - Logic for Programming, Artificial Intelligence, and Reasoning - 2013", Stellenbosch, South Africa, K. MCMILLAN, A. MIDDELDORP, A. VORONKOV (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 8312, pp. 258-273 [DOI : 10.1007/978-3-642-45221-5\_19], <http://hal.inria.fr/hal-00909319>
- [28] U. DAL LAGO, B. PETIT. *The geometry of types*, in "The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '13, Proceedings", Rome, Italy, ACM, 2013, pp. 167-178, <http://hal.inria.fr/hal-00909318>
- [29] O. DARDHA, E. GIACHINO, M. LIENHARDT. *A Type System for Components*, in "SEFM - International Conference on Software Engineering and Formal Methods - 2013", Madrid, Spain, R. M. HIERONS, M. G. MERAYO, M. BRAVETTI (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 8137, pp. 167-181 [DOI : 10.1007/978-3-642-40561-7\_12], <http://hal.inria.fr/hal-00909310>
- [30] O. DARDHA, D. GORLA, D. VARACCA. *Semantic Subtyping for Objects and Classes*, in "FMOODS/FORTE - IFIP Joint International Conference on Formal Techniques for Distributed Systems - 2013", Firenze, Italy, D. BEYER, M. BOREALE (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 7892, pp. 66-82 [DOI : 10.1007/978-3-642-38592-6\_6], <http://hal.inria.fr/hal-00909299>
- [31] R. DI COSMO, J. MAURO, S. ZACCHIROLI, G. ZAVATTARO. *Component Reconfiguration in the Presence of Conflicts*, in "ICALP - International Colloquium on Automata, Languages and Programming - 2013", Riga, Latvia, F. V. FOMIN, R. FREIVALDS, M. KWIATKOWSKA, D. PELEG (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 7966, pp. 187-198 [DOI : 10.1007/978-3-642-39212-2\_19], <http://hal.inria.fr/hal-00909336>
- [32] M. GABOARDI, A. HAEBERLEN, J. HSU, A. NARAYAN, B. PIERCE. *Linear Dependent Types for Differential Privacy*, in "40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages", Rome, Italy, Italy, POPL '13, ACM, 2013, pp. 357-370 [DOI : 10.1145/2429069.2429113], <http://hal.inria.fr/hal-00909340>
- [33] E. GIACHINO, C. A. GRAZIA, C. LANEVE, M. LIENHARDT, P. WONG. *DeadLock Analysis of Concurrent Objects – Theory and Practice*, in "IFM - 10th International Conference on integrated Formal Methods -

- 2013", Turku, Finland, E. B. JOHNSEN, L. PETR (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 7940, pp. 394-411 [DOI : 10.1007/978-3-642-38613-8\_27], <http://hal.inria.fr/hal-00909311>
- [34] E. GIACHINO, C. LANEVE. *A Beginner's Guide to the DeadLock Analysis Model*, in "TGC", Newcastle, United Kingdom, Lecture Notes in Computer Science, Springer, 2013, vol. 8191, pp. 49-63, <http://hal.inria.fr/hal-00909309>
- [35] D. HIRSCHKOFF, J.-M. MADIOT, D. SANGIORGI. *Name-passing calculi: from fusions to preorders and types*, in "LICS - 28th Annual ACM/IEEE Symposium on Logic in Computer Science - 2013", New Orleans, United States, IEEE, 2013, pp. 378-387 [DOI : 10.1109/LICS.2013.44], <http://hal.inria.fr/hal-00904138>
- [36] R. HÄHNLE, M. HELVENSTIJN, E. B. JOHNSEN, M. LIENHARDT, D. SANGIORGI, I. SCHAEFER, P. WONG. *HATS Abstract Behavioral Specification: The Architectural View*, in "FMCO - Formal Methods for Components and Objects - 2011", Turin, Italy, B. BECKERT, F. DAMIANI, F. S. DE BOER, M. M. BONSANGUE (editors), Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2013, vol. 7542, pp. 165-185 [DOI : 10.1007/978-3-642-35887-6\_6], <http://hal.inria.fr/hal-00909383>
- [37] I. LANESE, L. BEDOGNI, M. D. FELICE. *Internet of things: a process calculus approach*, in "SAC", Coimbra, Portugal, S. Y. SHIN, J. C. MALDONADO (editors), ACM, 2013, pp. 1339-1346, <http://hal.inria.fr/hal-00909303>
- [38] I. LANESE, M. LIENHARDT, C. MEZZINA, A. SCHMITT, J.-B. STEFANI. *Concurrent Flexible Reversibility*, in "22nd European Symposium on Programming, ESOP 2013", Rome, Italy, M. FELLEISEN, P. GARDNER (editors), Lecture Notes in Computer Science (LNCS), Springer, March 2013, vol. 7792, pp. 370-390 [DOI : 10.1007/978-3-642-37036-6\_21], <http://hal.inria.fr/hal-00811629>
- [39] I. LANESE, F. MONTESI, G. ZAVATTARO. *Amending Choreographies*, in "WWV - 9th International Workshop on Automated Specification and Verification of Web Systems - 2013", Florence, Italy, A. RAVARA, J. SILVA (editors), EPTCS, Open Publishing Association, 2013, vol. 123, pp. 34-48 [DOI : 10.4204/EPTCS.123.5], <http://hal.inria.fr/hal-00909305>
- [40] I. LANESE, G. ZAVATTARO. *Decidability Results for Dynamic Installation of Compensation Handlers*, in "COORDINATION", Florence, Italy, R. D. NICOLA, C. JULIEN (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 7890, pp. 136-150, <http://hal.inria.fr/hal-00909301>
- [41] T. A. LASCU, J. MAURO, G. ZAVATTARO. *Automatic Component Deployment in the Presence of Circular Dependencies*, in "10th International Symposium on Formal Aspects of Component Software, FACS 2013", Nanchang, China, 2013, <http://hal.inria.fr/hal-00909300>
- [42] F. MONTESI. *Process-aware web programming with Jolie*, in "SAC - Symposium on Applied Computing - 2013", Coimbra, Portugal, ACM, 2013, pp. 761-763 [DOI : 10.1145/2480362.2480507], <http://hal.inria.fr/hal-00909322>
- [43] F. MONTESI, N. YOSHIDA. *Compositional Choreographies*, in "CONCUR - Concurrency Theory - 2013", Buenos Aires, Argentina, P. R. D'ARGENIO, H. MELGRATTI (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 8052, pp. 425-439 [DOI : 10.1007/978-3-642-40184-8\_30], <http://hal.inria.fr/hal-00909321>

### Scientific Books (or Scientific Book chapters)



- [44] 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, <http://hal.inria.fr/hal-00909323>

### Research Reports

- [45] R. DI COSMO, M. LIENHARDT, R. TREINEN, S. ZACCHIROLI, J. ZWOLAKOWSKI. , *Optimal Provisioning in the Cloud. Technical report of the Aeolus project*, March 2013, <http://hal.inria.fr/hal-00831455>
- [46] R. DI COSMO, J. MAURO, S. ZACCHIROLI, G. ZAVATTARO. , *Component reconfiguration in the presence of conflicts. Technical report of the Aeolus project*, 2013, <http://hal.inria.fr/hal-00816468>

### Other Publications

- [47] D. HIRSCHKOFF, J.-M. MADIOT, D. SANGIORGI. , *Name-passing calculi: from fusions to preorders and types*, May 2013, <http://hal.inria.fr/hal-00818068>
- [48] T. A. LASCU, J. MAURO, G. ZAVATTARO. , *A Planning Tool Supporting the Deployment of Cloud Applications*, June 2013, Other publication, <http://hal.inria.fr/hal-00843925>

### References in notes

- [49] CHOR. , *Programming Language*, <http://www.chor-lang.org/>
- [50] 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
- [51] K. POHL, G. BÖCKLE, F. V. DER LINDEN. , *Software Product Line Engineering: Foundations, Principles, and Techniques Objects*, Springer Verlag, 2005