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

Activity Report 2011

# Project-Team FOCUS

## Foundations of Component-based Ubiquitous Systems

RESEARCH CENTER
**Sophia Antipolis - Méditerranée**

THEME
**Distributed Systems and Services**

# Table of contents

## Project-Team FOCUS

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

# 1. Members

**Faculty Members**

Mario Bravetti [“Ricercatore”, Bologna]
Ugo Dal Lago [“Ricercatore”, Bologna]
Maurizio Gabbrielli [“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 Hirschkoff [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]
Jacopo Mauro [Bologna]
Giulio Pellitta [Bologna]
Paolo Parisen Toldin [Bologna]

**Post-Doctoral Fellows**

Mila Dalla Preda [EU project Hats and Univ. Bologna]
Elena Giachino [EU project Hats]
Marco Gaboardi [Marie Curie scholarship]
Michael Lienhardt [EU project Hats]
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

- During the period 2006-2011, the Focus members have had 6 papers in the prestigious conference IEEE Symposium on Logic in Computer Science (LICS).
- Marco Gaboardi has obtained a Marie Curie International Outgoing Fellowship for Career Development. Project title: "Practical Light Types For Resource Consumption" (PLATFORM). Partner: University of Pennsylvania. Bologna scientist in charge : Simone Martini. University of Pennsylvania scientist in charge : Benjamin Pierce. Start Date: 2011-07-25 Duration: 36 months Project Funding: around 220KE.
- Fabrizio Montesi won the Confindustria/CINI italian national prize for one of the best six Master’s Theses in Information and Communication Technologies, with the thesis "Jolie: a service-oriented programming language". The contest receives applications from all areas related to ICT, from engineering to computer science.

# 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 $\pi$-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.

## 3.2. Foundational calculi and interaction

Modern distributed systems have witnessed a clear shift towards interaction and conversations as basic building blocks for software architects and programmers. The systems are made by components, that are supposed to interact and carry out dialogues in order to achieve some predefined goal; Web services are a good example of this. Process calculi are models that have been designed precisely with the goal of understanding interaction and composition. The theory and tools that have been developed on top of process calculi can set a basis with which CBUS challenges can be tackled. Indeed industrial proposals of languages for Web services such as BPEL are strongly inspired by process calculi, notably the $\pi$-calculus.

## 3.3. Type systems and logics

Type systems and logics for reasoning on computations are among the most successful outcomes in the history of the research in $\lambda$-calculus and (more recently) in process calculi. Type systems can also represent a powerful means of specifying dialogues among components of CBUS. For instance—again referring to Web services—current languages for specifying interactions only express basic connectivity, ignoring causality and timing aspects (e.g., an intended order on the messages), and the alternative is to use Turing Complete languages that are however undecidable. Types can come at hand here: they can express causality and order information on messages [58], [56], [59], while remaining decidable systems.

## 3.4. Implicit computational complexity

A number of elegant and powerful results have been recently obtained in implicit computational complexity in the $\lambda$-calculus in which ideas from Linear Logics enable a fine-grained control over computations. This experience can be profitable when tackling issues of CBUS related to resource consumption, such as resources allocation, access to resources, certification of bounds on resource consumption (e.g., ensuring that a service will answer to a request in time polynomial with respect to the size of the input data).

# 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

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”. Strictly speaking Service Oriented Computing is constructed around the Web Service technology: it allows web architects to produce new services by assembling existing services available from the shelves (in web development, the term ‘mashup’ is sometimes employed for this). According to the specific task at hand, a search is performed for suitable services to provide the needed capabilities.

## 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 [60]. 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 recently developed Jolie [7] (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 2011 the development of Jolie has continued. The main activities have been:

- A new session message-routing mechanism, based on correlation sets has been implemented. This mechanism makes message routing programmable from inside Jolie code.
- a new primitive for (smart) service aggregation.
- a graphical editor.
- an integrated development environment.

Moreover, this year Jolie has been used for teaching, in a 30-hour master course at IT University of Copenhagen, Denmark.

## 5.2. Others

Below we list some sofwtare that has been developed in Focus in previous years but that during 2011 has not substantially changed. Short descriptions of these items can be found in the Focus activity report for 2010.

- `PiDuce` (see http://www.cs.unibo.it/PiDuce/) is a prototype for experimenting Web services technologies, based on theories of process calculi and XML documents and schemas [3].
- *IntML* is a functional programming language guaranteeing sublinear space bounds for all programs [57].

# 6. New Results

## 6.1. Contracts and sessions

**Participants:** Ugo Dal Lago, Ornela Dardha, Maurizio Gabbrielli, Elena Giachino, Claudio Guidi, Jacopo Mauro, Fabrizio Montesi.

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.

The expected dialogue in a session can be specified by means of types, the *session types*.

We have studied [11] the integration of union and session types in a class-based language for building network applications, which amalgamates sessions and methods in order to flexibly exchange data according to communication protocols. We have established type safety properties guaranteeing that, after a session has started, computation cannot get stuck on a communication deadlock, and studied type inference. On a similar topic is the paper [23].

We have used types [32] to guarantee bounds on sessions. These are polynomial bounds, on both time and space needed by the interacting processes to carry out the interactions in their sessions. This is the first example of a refinement of session types guaranteeing quantitative properties beyond the usual safety property, and builds on earlier work on soft linear logic.

In service-oriented architectures, the mechanism that allows to manage sessions and, in particular, to assign incoming messages to the correct sessions, is critical for efficiency and performance. A well-known solution to this problem, first introduced by BPEL, makes use of *correlation sets*. Intuitively these distinguish different sessions by means of the values for some specific variables which are present also in messages, thus allowing for their routing to sessions on the basis of these values. We have studied [36] a typed language for programming services based on correlation sets, that takes into account key aspects of service-oriented systems, such as distribution, loose coupling, open-endedness and integration. We have provided an implementation of the language as an extension of the Jolie language and applied it to a nontrivial real-world example of a fully-functional distributed user authentication system.

In current SOC languages based on correlation sets, a message can be assigned to a unique session. In another line of work on correlation sets [35], we have studied the possibility – useful in many practical examples – of broadcasting messages to more than one session. We have investigated a data structure, based on radix trees, and an algorithm for managing a correlation mechanism, that support the broadcast primitive without degrading performances.

## 6.2. Fault Handling, compensations and transactions

**Participants:** Mila Dalla Preda, Maurizio Gabbrielli, Ivan Lanese, Jacopo Mauro, Gianluigi Zavattaro.

One of the predominant properties of CBUS is the loose coupling among the components. In fact, components can dynamically connect/disconnect and can be modified/updated at run time. It is thus important to support unexpected events, called faults.

In [30] we have studied the problem of fault handling in the kind of object-oriented languages developed in the EU Hats project; notably these languages have asynchronous method calls whose results are returned inside futures. We present an extension for those languages where futures are used to return fault notifications and to coordinate error recovery between the caller and callee. This can be exploited to ensure that invariants involving many objects are restored after faults.

Traditional fault handling mechanisms, including those based on try-catch operators, do not seem sufficient to deal with the non-local errors and failures of distributed systems. At the application level, more advanced transactional models and primitives are needed to guarantee integrity and continuity of the whole system. We study approaches based on long running transactions and compensations. A long running transaction is a computation that either successfully terminates, or it aborts. In case of abort, a compensation is executed to take the system to a consistent state. In [53], extending work started last year, we make a thorough comparison among different approaches to the specification of compensations, in particular static forms of recovery where the compensation is statically defined together with the transaction, and dynamic forms where the compensation is progressively built along with a computation.

We have also continued our study on faults and compensations in Service Oriented Computing. The approach to the interplay between bi-directional request-response interaction and faults, proposed in our past works on the Jolie language, supported the idea that the bi-directional pattern should not be interrupted in case of faults. However, this may cause long delays or even deadlocks if the communicating partner disappears. On the contrary, the approach of WS-BPEL causes no delay, but it does not allow to compensate the remote activity. We have investigated [38] an intermediate approach in which it is not necessary for the fault handler to wait for the response, but it is still possible on response arrival to gracefully close the conversation with the remote service.

A related work, but mainly developped in 2010, is [21].

## 6.3. Service orchestration and choreography

**Participants:** Mila Dalla Preda, Maurizio Gabbrielli, Claudio Guidi, Ivan Lanese, Jacopo Mauro, Fabrizio Montesi, Marco Pistore, Gianluigi Zavattaro.

Orchestration has to do with the definition of services that should obey given behaviours. The services may be realised by composing services already available. Orchestration is often discussed in relationship with "choreography", which refers to global descriptions of the intended behaviour of a system of components, stating the role of each participant and the set of coordination requirements.

In [55] we have studied the basic linguistic constructs and a reference implementation for aggregation, a mechanism for composing services that abstracts from the order of their communications. Aggregation is widely used in practice. However, since it is not natively supported by service-oriented languages, it is mostly implemented by means of ad-hoc solutions which typically exploit middleware technology.

A critical aspect for pervasive computing is the possibility to discover and use process knowledge at run time depending on the specific context. This can be achieved by using an underlying service-based application and exploiting its features in terms of dynamic service discovery, selection, and composition. Pervasive process fragments represent a service-based tool that allows to model incomplete and contextual knowledge. In [40] we provide a solution to automatically compose such fragments into complete processes, according to a specific context and specific goals. We compute the solution by encoding process knowledge, domain knowledge and goals into an AI planning problem.

Concerning choreography languages, two main approaches have been followed in their design: the interaction-oriented approach at the basis of WS-CDL and the process-oriented approach of BPEL4Chor. In [52] we investigate the relationships between the two approaches In particular, we point out several possible interpretations for interaction-oriented choreographies: one synchronous and various asynchronous. Under each of these possible interpretations we characterize the class of interaction-oriented choreographies which have a direct process-oriented counterpart, and we formalize the corresponding notion of equivalence between the initial interaction-oriented choreography and the corresponding process-oriented counterpart.

In [50] we study the issue of checking a multiparty choreography against formal protocol specifications, and then projecting it onto a description of the individual service orchestrators. Contributions are also the definition of a multiparty choreography model, and the correctness proof for the projection.

## 6.4. Primitives for adaptable and evolvable components

**Participants:** Mario Bravetti, Ivan Lanese, Michael Lienhardt, Jacopo Mauro, Marco Pistore, Davide Sangiorgi, Gianluigi Zavattaro.

In Focus we study linguistic primitives for components, and models for them following the process calculus approach. A special emphasis is given to the adaptability and evolvability of the components — important issue in complex software systems. Components indeed are often used in contexts that had not been predicted at the time when the components were built. Moreover, the needs and the requirements on a system may change over time: one may find that the original specification was incomplete or ambiguous, or new needs may arise that had not been predicted at design time. As designing and deploying a system is costly, it is important that the system be capable of evolving and adapting itself to changes in the surrounding environment.

Models and linguistic constructs for adaptability and evolvability of components are studied in [34] and [19]. The key features of the component model in [34] are: a hierarchical structure of components; the capacity to move, update, wrap components; method interfaces for components; and capacities to isolate and distribute components. In the model in [19], adaptable processes have a location and can be subject to dynamic update actions at runtime (related to this paper is also [20]).

In [22] we provide an adaptation approach that can automatically adapt business processes to run-time context changes that impede achievement of a business goal. We define a formal framework that adopts planning techniques to automatically derive necessary adaptation activities on demand. The adaptation consists in identifying recovery activities that guarantee that the execution of a business process can be successfully resumed and, as a consequence, the business goals are achieved. The solution proposed is evaluated on a real-world scenario from the logistics domain.

Adaptability and evolvability are major concerns in Software Product Lines. The EU Hats project has developed the idea of delta-oriented programming (DOP) as a technique for implementing Software Product Lines based on modifications (add, remove, modify) to a core object-oriented program. Such modifications can introduce errors into a program, when type signatures of classes are modified. To overcome this problem we have introduced [54] a type system for delta-oriented programs. The system is based on row polymorphism, a well-known method in type systems for records.

## 6.5. Resource Control

**Participants:** Ugo Dal Lago, Marco Gaboardi, Daniel Hirschkoff, Simone Martini, Paolo Parisen Toldin, Giulio Pellitta, Davide Sangiorgi.

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.

We have continued our work on techniques for ensuring termination of programs. On the one hand we have refined [25] previous techniques, enhancing them by taking into account input/output capabilities of channels and subtyping. On the other hand we have studied [28] how to transport techniques initially devised for processes onto sequential higher-order languages with imperative features (e.g., $\lambda$-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.

In [31], a type system of linear and dependent types, called dlPCF, has been proved to be a sound, but also relatively complete, way to prove intensional (but also extensional) properties of PCF programs. In other words, not only all properties of programs proved by typing are operationally valid, but all true properties of programs can be proved so by way of dlPCF. This holds not only for terms of base type, but also for (first-order) functions: this makes the type system more expressive than intersection type disciplines.

A characterization of probabilistic polynomial time classes by way of linear typing systems for a variant of Godel's T called RSLR, has been proposed [51]. Classes like BPP and PP are characterized by RSLR once appropriate constraints on the probability of error are imposed.

A unifying methodology for the study of resource consumption of processes has been presented in [13]: it is a refinement of realizability, in which not only termination but also concrete resource bounds can be obtained by showing a function to be realized by a program.

On the same topic is the paper [26], that polishes previous work.

## 6.6. Verification of extensional properties

**Participants:** Elena Giachino, Cosimo Laneve, 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. We mainly employ techniques based on behavioral equivalences (and preorders), and on types and logics. Type systems offer a good trade-off between expressiveness and efficiency of the techniques. A substantial amount of the work carried out this year has to do with the transfer of techniques between the areas of concurrency theory and object-oriented languages.

We have developed [29] a technique for the deadlock analysis of systems of concurrent object groups. The technique makes use of types in the form of contracts, that is, abstract descriptions of method's behaviours. Object groups are collections of objects that perform collective work. Within a group, there can be only one running thread at a time; the scheduling of threads is cooperative.

We have studied [37] the concept of ownership types, originally introduced for (sequential) object-oriented languages, in the setting of pure message-passing concurrency. Ownership types have the effect of statically preventing certain communications, and can block the accidental or malicious leakage of secrets. Intuitively, a channel defines a boundary and forbids access to its inside from outer channels, thus preserving the secrecy of the inner names from malicious outsiders.

In a different line of work, we have analyzed ad hoc networks, intended as networks of devices connected by wireless links and communicating via broadcast. We have considered [27], [18] models in which the communication topology of a network is represented as a graph. Nodes represent states of individual processes, and adjacent nodes represent single-hop neighbors. Processes are finite state automata that communicate via selective broadcast messages. Reception of a broadcast is restricted to single-hop neighbors. In these systems we have studied various forms of reachability (example: the existence of an initial topology in which the execution of the protocol can lead to a configuration with at least one node in a certain state).

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 [47], [49] two comprehensive textbooks on bisimulation and coinduction (in [49], Sangiorgi is an editor, and author of two chapter contributions [48], [46]). 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.

## 6.7. Tutorial papers on Service-Oriented Computing

**Participants:** Mario Bravetti, Ivan Lanese, Davide Sangiorgi, Gianluigi Zavattaro.

We have contributed to a few tutorial papers that summarise the work on languages and tools for Service-Oriented Computing that has been carried out within the EU project Sensoria in Focus and elsewhere. The papers appear as chapters of a book dedicated to the topic.

The chapters [45] and [43] present and contrast the primitives and the behavioural theories of the main process calculi designed for modeling services.

Languages and models for service-oriented applications usually include primitives and constructs for exception and compensation handling. Exception handling is used to react to unexpected events while compensation handling is used to undo previously completed activities. In [44] we investigate the impact of exception and compensation handling in service-oriented process calculi.

The chapter [42] deals with contracts: descriptions intended to provide support for the automatic on-demand discovery of functionalities offered by a service. The approach followed is to describe such contracts as process calculi expressions. We show how, in certain cases, service contracts can be automatically extracted out of service behaviour, and how they can be used to formally check compliance among the communication protocols followed by the interacting services.

Finally, in [41] we present different tools that have been developed for verifying properties of service implementations with respect to their formal specifications in an automated, or semi-automated, way.

## 6.8. Expressiveness of computational models

**Participants:** Mila Dalla Preda, Ugo Dal Lago, Ivan Lanese, Cosimo Laneve, Davide Sangiorgi.

Expressiveness refers to the study of the expressive power of computational models. In 2011 we have addressed four main aspects.

First, we have continued our investigation of reversible computations. Reversibility is a main ingredient in the study of programming abstractions for reliable systems, e.g. for exception handling. In fact, reversibility can be used for going back to some consistent state after an exception has occurred. In previous years we had defined $\rho\pi$, a higher-order calculus where processes can both go forward and backward in the computation. This year we have studied [33] fine-grained rollback primitives to control reversibility. The definition of a proper semantics for such a primitive is a surprisingly delicate matter because of the potential interferences between concurrent rollbacks. We have also considered lower-level distributed semantics, which are closer to an actual implementation of the rollback primitives, and their relationship with the high-level semantics.

A thread of research close to that of $\rho\pi$ is the study of the properties and the expressive power of a simple calculus with reversible transitions, called *reversible structures*. In [24], we have demonstrated a standardization theorem for these structures. When terms in reversible structures have unique id, the standardization theorem may be strengthened in a form that bears a quadratic algorithm for reachability, a problem that is EXPSPACE-complete for generic structures (as in Petri Nets). The expressive power of reversible structures has been studied in [17], [12] and a compilation of asynchronous Reversible CCS has been provided.

A second aspect has been been motivated by the analogy between malicious software and biological infections [39]. In the paper, we have used a formalism originally developed for the analysis of biological systems — the kappa calculus by Danos and Laneve — for the formalization and analysis of malicious software. In particular we have modeled the different actors involved in a malicious code attack in the kappa-calculus. Then, by simulating the behavior, we have shown how to extract relevant information that can drive the choice of the defense technique to apply.

A third aspect has been the refinement [14] of some previous work on the expressiveness and decidability of higher-order concurrent languages. — formalisms for concurrency in which processes can be passed around in communications.

A fourth aspect has been the study of properties of a simple calculus for quantum computation. In [16], we have demonstrated a confluence property both for finite and infinite computations using a novel technique.

# 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: Zavattaro, Sangiorgi.
- 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.

This project has been presented during a poster session at the “journées scientifiques de l’INRIA” in Paris, November 2011.

- S. Martini, U. Dal Lago, M. Gaboardi, and D. Sangiorgi are involved in the CNRS PICS 2010 (“International Projects for Scientific Cooperation”) project “Linear Logic and applications”.
- REVER (Programming Reversible Recoverable Systems) is an ANR project starting 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: Lanese, Laneve, Zavattaro.

## 7.2. European Initiatives

- 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.

## 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.

### *7.3.2. Visits of International Scientists*

Ferret’s visit below has also been used to give a short intensive course on abstract interpretation for PhD students in Bologna.

- Beniamino Accattoli, LIX, Polytechnique Paris. June 6-9 and October 24-28. Topic: the complexity of beta reduction in the lambda calculus.
- Matteo Cimini, Icelanding Center of Excellence in Theoretical Computer Science, Reykjavik, 1 week in December. Topic: logics for concurrent languages.
- Giorgio Delzanno, University of Genova. 2-day visit. Topic: Verification of protocols for Mobile Ad Hoc Networks.
- Claudia Faggian, PPS Paris 7, July 18-21 and October 3-7. Topic: linear logic and quantum computation
- Jerome Ferret, ENS Paris and INRIA, 1 week. Topic: abstract interpretation.
- Paolo di Giamberardino, University of Rome "La Sapienza", September 19-23. Topic: soft session types.
- Harry Mairson, Brandeis University, USA. February 4-16. Topic: the complexity of evaluation in the simply typed lambda calculus.
- Matias David Lee. Currently PhD student at Universidad Nacional de Córdoba, Argentina. Arrived in November 2001 for a long-term visit (6 months).

- Damiano Mazza, CNRS and Paris 13. October 17-20. Topic: computational complexity in a concurrency scenario.
- Claudio Mezzina, Inria Grenoble, a few visits throughout the year, two months in total. Topic: constructs for reversible computations.
- Jean-Bernard Stefani, Inria Grenoble, two 2-day visits. Topic: models for components and reversibility.

### *7.3.3. Other cooperations*

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

- Inria EPI Indes, (on orchestration and programming languages). A common meeting was organised in Sophia Antipolis, May 2011, where 8 people from Focus and almost everybody from Indes participated.

  Focus and Indes have moreover significantly contributed to the "Programming language day", May 31, 2011, Amphi Morgenstern, INRIA Sophia Antipolis – Méditerranée.
- 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).
- 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, starting in September 2011).
- Laboratoire d'Informatique, Université Paris Nord, Villetaneuse (on implicit computational complexity). Contact person(s) in Focus: Dal Lago, Gaboardi, Martini. Several visit exchanges during the year, in both directions. An Italian PhD student (Marco Solieri) will soon start a PhD thesis with joint supervision. Gaboardi has made a 2-month visit.
- Team PPS, University of Paris-Diderot Paris 7 (on logics for processes, resource control). Contact person(s) in Focus: Dal Lago, Gaboardi, Martini, Sangiorgi, Zavattaro. Various short visits in both directions during the year.
- Research Institute for the Mathematical Sciences – RIMS – University of Kyoto, Japan (on typing and resource control). Contact person(s) in Focus: Dal Lago. A 2-week exchange (Dal Lago) in 2011.
- Computer Science Department, Brandeis University, USA (on complexity of evaluation in functional programming languages). Contact person(s) in Focus: Dal Lago, Martini. A 2-week visit exchange (Mairson) in 2011.
- 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).
- EPI Carte, INRIA-Nancy Grand Est and LORIA (on implicit computational complexity). Contact person(s) in Focus: Gaboardi. A few short visits during 2011.
- 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) is starting at the end of 2011.
- Inria EPI Signes, Inria Bordeaux Sud-Ouest (on lambda-calculi, linear logic and semantics). Contact person(s) in Focus: Dal Lago, Martini. Martini visited Signes for a few days in December 2010. One joint PhD supervision (Ivano Ciardelli).

# 8. Dissemination

## 8.1. Animation of the scientific community

**M. Bravetti:** Program Committee member of the following conferences: 8th European Performance Engineering Workshop (EPEW); 4th International Conference on Fundamentals of Software Engineering (FSEN); 4th IEEE International Conference on Software Testing, Verification and Validation (ICST); 1st International Workshop on Trustworthy Service-Oriented Computing; Process Algebra and Coordination (PACO); 1st Workshop on Formal Methods in the Development of Software (WS-FMDS); 11th International Conference On Quality Software (QSIC); 4th IEEE International Conference on Cloud Computing (CLOUD). Invited lecturer at 7th TAROT Summer School on Software Testing, Verification and Validation (TAROT) Saint Petersburg, Russia.

**M. Gaboardi:** Invited speaker at Workshop on Linear Logic, Kyoto, Japan. Organizing Committee member of the following events: Oregon Programming Language Summer School (OPLSS 2011), Eugene, Oregon; June 16-July 1, 2011. XXIV Incontro di Logica (AILA 2011), Bologna, February 2-4, 2011.

**U. Dal Lago:** Program Committee member of the following conferences: Tenth International Conference on Typed Lambda Calculi and Applications (TLCA), Novisad, Serbia; International Workshop on Games in Logic and Programming (GaLoP), Saarbrucken, Germany. An invited talk at the Workshop on Geometry of Interaction, Traced Monoidal Categories and Implicit Computational Complexity, Kyoto, Japan, Novembre 2011.

**M. Gabbrielli:** Organizer in March 2011 of the third International School on Computational logic. Member of the editorial board of the Int'l Journal Theory and Practice of Logic Programming. Member of the program committee of the 4th International Conference on Fundamentals of Software Engineering (FSEN).

**I. Lanese:** Program Committee co-Chair of the Service Oriented Architectures and Programming track of the 26th Annual ACM Symposium on Applied Computing (SAC 2011), Taichung, Taiwan Program Committee member of the following conferences: 1st Conference on Electronics, Telecommunications and Computers (CETC) Lisboa, Portugal; 1st International Workshop on TRUstworthy Service Oriented Computing (INTRUSO), Copenaghen, Denmark; 4th Interaction and Concurrency Experience (ICE), Reykjavik, Iceland. Invited to give a talk at IMT Lucca, Italy, on "Control of Reversibility", October 2011.

**C. Laneve:** Program Committee member of: 18th International Workshop on Expressiveness in Concurrency (EXPRESS), Aachen, Germany, September 2011; 8th International Conference on Quantitative Evaluation of SysTems (QUEST), Aachen, Germany. Invited speaker at the 5th Workshop on Membrane Computing and Biologically Inspired Process Calculi, Versailles, September 2011.

**S. Martini:** Program Committee member of: Foundational and Practical Aspects of Resource Analysis (FOPARA), Madrid, Spain; Second Workshop on Developments in Implicit Computational Complexity (DICE), Saarbrucken, Germany; Member of the final jury for the PhD of Alberto Carraro, Universitá Ca' Foscari, Venezia, Italy. Rapporteur of the PhD thesis of Beniamino Accattoli, Universitá di Roma "La Sapienza", Italy. Organizing Committee member of the XXIV Incontro di Logica (AILA 2011), Bologna, February 2-4, 2011.

**D. Sangiorgi:** Member of the editorial board of the journals: Logical Methods in Computer Science, Acta Informatica, and Distributed Computing. Chairman of the IFIP Working Group 2.2 (Formal description of programming concepts). Program Committee member of the conferences: 4th International Conference on Algebraic Informatics, Linz, Austria; Twenty-Sixth Annual IEEE Symposium on Logic in Computer Science (LICS), Toronto, Ontario, Canada; the 38th International Colloquium on Automata, Languages and Programming (ICALP), Zurich, Switzerland; the Twenty-seventh Conference on the Mathematical Foundations of Programming Semantics (MFPS), Pittsburgh; workshop Process Algebra and Coordination (PaCo), Reykjavik, Iceland. Member of the evaluation committee of the "Habilitation a Diriger des Recherche" (HDR) of Alan Schmitt, INRIA Grenoble, France. Invited lecturer at the school 'Software Engineering and Formal Methods' November 7-11, 2011 Montevideo, Uruguay.

**G. Zavattaro:** Program Committee Chair of 9th IEEE European Conference on Web Services (ECOWS). Program Committee member of the following conferences: 5th International Workshop on Reachability Problems (RP); 1st International Workshop on Verification, Analysis, and Testing of Service Systems (VATSS); 8th International Workshop on Web Services and Formal Methods (WS-FM); 5th Membrane Computing and Biologically Inspired Process Calculi workshop (MeCBIC); 4th IEEE International Conference on Cloud Computing (CLOUD); 2nd Int. Workshop on Interactions between Computer Science and Biology (CS2Bio); Twenty-Seventh Conference on the Mathematical Foundations of Programming Semantics (MFPS). Invited speaker at 1st International Workshop on Process Algebra and Coordination (PaCo). Member of the Editorial Board of the Journal of Software (JSW). Member of the evaluation committee of the "Habilitation a Diriger des Recherche" (HDR) of Gwen Salaun, INRIA Grenoble, France.

## 8.2. Organisation of events

- Third International Spring School on Computational Logic, 10-15 April 2011, Bertinoro, Italy
- The Oregon Programming Languages Summer School (OPLSS) has been held at the University of Oregon each summer since 2002. The focus of the 2011 edition has been in the mix or interplay of theory and practice in program verification. The 2011 edition featured 14 lecturers and 94 students. Marco Gaboardi has been involved in the organisation.
- XXIV Incontro di Logica (AILA 2011), Bologna; February 2-4, 2011. The series of "Incontri di Logica" offers researchers in mathematics, philosophy of science and computer science an arena to discuss all aspects of mathematical logic which relate to their research. The 2011 edition held in Bologna has welcomed about 100 participants mainly from Italy.

## 8.3. 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 Java", 20 hours, 1st 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

  PhD: "Semantics of Linear Logic", 8 hours, Universita' degli Studi di Milano, Italy.

  PhD: "Category Theory and Programming Languages", 10 hours, University of Oregon, USA.

- Ivan Lanese

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

  Master: "Data Bases and Information Systems", 24 hours, 1st year, University of Bologna, Italy.

- Cosimo Laneve

Undergraduate: “Programmazione”, 90 hours, 1st year, University of Bologna, Italy.

Master: “Analisi di Programmi”, 80 hours, 1st year, University of Bologna, Italy.

- Simone Martini

  Undergraduate: “Programming languages”, 40 hours, 2nd 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: “Classical Logic”, 12 hours, 1st and 2nd 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.

- Elena Giachino

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

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.

Ornela 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.

Jean-Marie Madiot, september 2011, Types, proofs and proof techniques for process calculi, D. Hirschkoff and D. Sangiorgi.

Jacopo Mauro, january 2009, Constraints and concurrency, M. Gabbrielli.

Giulio Pellitta, january 2011, Implementations of geometry of interaction, S. Martini.

Paolo Parisen Toldin, january 2010, Implicit computational complexity for probabilistic classes, S. Martini.

A few other PhD students are (co)-supervised by members of Focus, but are not in the Focus team:

- Michele Alberti - Marseille; 1/12/2011; Separation results in the algebraic lambda-calculus; L. Regnier/S. Martini
- Ivano Ciardelli - Bordeaux; 1/11/2010; Models of first order linear logic and their use in linguistics; C. Retoré/S. Martini
- Claudio Mezzina - Grenoble; 1/9/2008; Reversible computations in processes; Stefani-Sangiorgi.
- Marco Solieri - Paris Nord; 1/12/2011; Complexity of reduction in sharing graphs; S. Guerrini/S. Martini
- Sara Zuppiroli, Bologna; january 2010; models for quantum computing; U. Dal Lago.

# 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] S. Carpineti, C. Laneve, L. Padovani. *PiDuce - a project for experimenting Web services technologies*, in "Science of Computer Programming", 2009, vol. 74, n$^{o}$ 10, p. 777–811.

[4] 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.

[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. Hofmann. *Quantitative Models and Implicit Complexity*, in "FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science, 25th International Conference", R. Ramanujam, S. Sen (editors), Lecture Notes in Computer Science, Springer, 2005, vol. 3821, p. 189-200.

[7] F. Montesi, C. Guidi, G. Zavattaro. *Composing Services with JOLIE*, in "Fifth IEEE European Conference on Web Services (ECOWS 2007)", 2007, p. 13-22.

[8] B. Pierce, D. Sangiorgi. *Behavioral equivalence in the polymorphic pi-calculus*, in "Journal of the ACM", 2000, vol. 47, n$^{o}$ 3, p. 531–584.

[9] D. Sangiorgi, D. Walker. *The $\pi$-calculus: a Theory of Mobile Processes*, Cambridge University Press, 2001.

## Publications of the year

### Articles in International Peer-Reviewed Journal

[10] P. Baillot, P. Coppola, U. D. Lago. *Light logics and optimal reduction: Completeness and complexity*, in "Inf. Comput.", 2011, vol. 209, n$^{o}$ 2, p. 118-142, http://dx.doi.org/10.1016/j.ic.2010.10.002.

[11] L. BETTINI, S. CAPECCHI, M. DEZANI-CIANCAGLINI, E. GIACHINO, B. VENNERI. *Deriving Session and Union Types for Objects*, in "Mathematical Structures in Computer Science", 2011, To appear.

[12] L. CARDELLI, C. LANEVE. *Reversibility in Massive Concurrent Systems*, in "Scientific Annals of Computer Science", 2011, vol. XXI(2), p. 175-198, http://www.infoiasi.ro/bin/download/Annals/XXI2/XXI2_0.pdf.

[13] U. D. LAGO, M. HOFMANN. *Realizability models and implicit complexity*, in "Theor. Comput. Sci.", 2011, vol. 412, n$^{o}$ 20, p. 2029-2047, http://dx.doi.org/10.1016/j.tcs.2010.12.025.

[14] I. LANESE, J. A. PÉREZ, D. SANGIORGI, A. SCHMITT. *On the expressiveness and decidability of higher-order process calculi*, in "Inf. Comput.", 2011, vol. 209, n$^{o}$ 2, p. 198–226.

[15] D. SANGIORGI, N. KOBAYASHI, E. SUMII. *Environmental bisimulations for higher-order languages*, in "ACM Trans. Program. Lang. Syst.", 2011, vol. 33, n$^{o}$ 1, 5.

### Articles in Non Peer-Reviewed Journal

[16] U. D. LAGO, A. MASINI, M. ZORZI. *Confluence Results for a Quantum Lambda Calculus with Measurements*, in "Electr. Notes Theor. Comput. Sci.", 2011, vol. 270, n$^{o}$ 2, p. 251-261, http://dx.doi.org/10.1016/j.entcs.2011.01.035.

### Invited Conferences

[17] L. CARDELLI, C. LANEVE. *Reversibility in Massive Concurrent Systems*, in "Proceedings of 5th Workshop on Membrane Computing and Biologically Inspired Process Calculi (MeCBIC 2011)", 2011, vol. abs/1108.3419, p. 3-4, http://arxiv.org/abs/1108.3419.

[18] G. DELZANNO, A. SANGNIER, G. ZAVATTARO. *Parameterized Verification of Safety Properties in Ad Hoc Network Protocols*, in "PACO", EPTCS, 2011, vol. 60, p. 56-65.

### International Conferences with Proceedings

[19] M. BRAVETTI, C. D. GIUSTO, J. A. PÉREZ, G. ZAVATTARO. *Adaptable Processes (Extended Abstract)*, in "FMOODS/FORTE Conf. on Formal Techniques for Distributed Systems", Lecture Notes in Computer Science, Springer, 2011, vol. 6722, p. 90-105.

[20] M. BRAVETTI, C. D. GIUSTO, JORGE A. PÉREZ, G. ZAVATTARO. *Steps on The Road to Component Evolvability*, in "Formal Aspects of Component Software (FACS 2010)", Lecture Notes in Computer Science, Springer, 2011, vol. 6921, p. 295-3299.

[21] R. BRUNI, A. KERSTEN, I. LANESE, G. SPAGNOLO. *A New Strategy for Distributed Compensations with Interruption in Long-Running Transactions*, in "Proc. of WADT 2010", Lecture Notes in Computer Science, Springer, 2011, vol. 7137, p. 42-60, to appear.

[22] A. BUCCHIARONE, M. PISTORE, H. RAIK, R. KAZHAMIAKIN. *Adaptation of Service-based Business Processes by Context-Aware Replanning*, in "IEEE International Conference on Service Oriented Computing & Applications (SOCA 2011)", 2011, to appear.

[23] S. CAPECCHI, E. GIACHINO, N. YOSHIDA. *Global Escape in Multiparty Sessions*, in "Proceedings of the 30th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'10)", LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011, vol. 8, p. 338–351.

[24] L. CARDELLI, C. LANEVE. *Reversible Structures*, in "CMSB '11: Proceedings of the 9th International Conference on Computational Methods in Systems Biology", New York, NY, USA, ACM, 2011, p. 131-140, http://www.cs.unibo.it/~laneve/papers/ReversibleStructures.pdf.

[25] I. CRISTESCU, D. HIRSCHKOFF. *Termination in a Pi-calculus with Subtyping*, in "Proceedings 18th International Workshop on Expressiveness in Concurrency", EPTCS, 2011, vol. 64, p. 44–58.

[26] U. DAL LAGO, S. MARTINI, D. SANGIORGI. *Light Logics and Higher-Order Processes*, in "17th International Workshop on Expressiveness in Concurrency, EXPRESS 2010", EPTCS, 2011, p. 46–60.

[27] G. DELZANNO, A. SANGNIER, G. ZAVATTARO. *On the Power of Cliques in the Parameterized Verification of Ad Hoc Networks*, in "FOSSACS", Lecture Notes in Computer Science, Springer, 2011, vol. 6604, p. 441-455.

[28] R. DEMANGEON, D. HIRSCHKOFF, D. SANGIORGI. *Termination in impure lambda-calculus*, in "Fundamentals of Software Engineering", Lecture Notes in Computer Science, Springer, 2011, to appear, http://perso.ens-lyon.fr/romain.demangeon/Recherche/impfun2.pdf.

[29] E. GIACHINO, C. LANEVE. *Analysis of Deadlocks in Object Groups*, in "FMOODS/FORTE", 2011, p. 168–182.

[30] E. B. JOHNSEN, I. LANESE, G. ZAVATTARO. *Fault in the Future*, in "Proc. of COORDINATION 2011", W. D. MEUTER, G.-C. ROMAN (editors), Lecture Notes in Computer Science, Springer, 2011, vol. 6721, p. 1-15, http://www.cs.unibo.it/~lanese/publications/fulltext/coordination2011.pdf.gz.

[31] 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.

[32] U. D. LAGO, P. D. GIAMBERARDINO. *Soft Session Types*, in "Proceedings 18th International Workshop on Expressiveness in Concurrency", EPTCS, 2011, vol. 64, p. 59-73, http://dx.doi.org/10.4204/EPTCS.64.5.

[33] 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, p. 297-311, http://www.cs.unibo.it/~lanese/publications/fulltext/concur2011.pdf.gz.

[34] M. LIENHARDT, I. LANESE, M. BRAVETTI, D. SANGIORGI, G. ZAVATTARO, Y. WELSCH, J. SCHAFER, A. POETZSCH-HEFFTER. *A Component Model for the ABS Language*, in "Formal Methods for Components and Objects (FMCO) 2010", Lecture Notes in Computer Science, Springer, 2011, vol. 6957, p. 165-185, http://www.cs.unibo.it/~lanese/publications/fulltext/fmco2010.pdf.gz.

[35] J. MAURO, M. GABBRIELLI, C. GUIDI, F. MONTESI. *An Efficient Management of Correlation Sets with Broadcast*, in "COORDINATION", 2011, p. 80-94.

[36] F. MONTESI, M. CARBONE. *Programming services with correlation sets*, in "ICSOC.", 2011, to appear, http://www.itu.dk/~fabr/icsoc2011/.

[37] M. PATRIGNANI, D. CLARKE, D. SANGIORGI. *Ownership Types for the Join Calculus*, in "FMOODS/FORTE Conf. on Formal Techniques for Distributed Systems", R. BRUNI, J. DINGEL (editors), Lecture Notes in Computer Science, Springer, 2011, vol. 6722, p. 289–303.

[38] M. D. PREDA, M. GABBRIELLI, I. LANESE, J. MAURO, G. ZAVATTARO. *Graceful Interruption of Request-response Service Interactions*, in "Proc. of ICSOC 2011", Lecture Notes in Computer Science, Springer, 2011, vol. 7084, http://www.cs.unibo.it/~lanese/publications/fulltext/icsoc2011.pdf.gz.

[39] M. D. PREDA, C. D. GIUSTO. *Hunting Distributed Malware with the $\kappa$-Calculus*, in "Fundamentals of Computation Theory - 18th International Symposium, FCT 2011", Lecture Notes in Computer Science, Springer, 2011, vol. 6914, p. 102-113.

[40] A. SIRBU, A. MARCONI, M. PISTORE, H. EBERLE, F. LEYMANN, T. UNGER. *Dynamic Composition of Pervasive Process Fragments*, in "IEEE International Conference on Web Services (ICWS 2011)", 2011, p. 73-80.

### Scientific Books (or Scientific Book chapters)

[41] M. BARTOLETTI, L. CAIRES, I. LANESE, F. MAZZANTI, D. SANGIORGI, H. T. VIEIRA, R. ZUNINO. *Tools and Verification*, in "Rigorous Software Engineering for Service-Oriented Systems - Results of the SENSORIA project on Software Engineering for Service-Oriented Computing", M. WIRSING, M. HOLZL (editors), Lecture Notes in Computer Science, Springer, 2011, vol. 6582, chap. 5.3, p. 408-427, http://www.cs.unibo.it/~lanese/publications/fulltext/bookSens-tools.pdf.gz.

[42] M. BOREALE, M. BRAVETTI. *Advanced mechanisms for service composition, query and discovery*, in "Rigorous Software Engineering for Service-Oriented Systems - Results of the SENSORIA project on Software Engineering for Service-Oriented Computing", M. WIRSING, M. HOLZL (editors), Lecture Notes in Computer Science, Springer, 2011, vol. 6582.

[43] L. CAIRES, R. D. NICOLA, R. PUGLIESE, V. T. VASCONCELOS, G. ZAVATTARO. *Core Calculi for Service-Oriented Computing*, in "Rigorous Software Engineering for Service-Oriented Systems - Results of the SENSORIA project on Software Engineering for Service-Oriented Computing", M. WIRSING, M. HOLZL (editors), Lecture Notes in Computer Science, Springer, 2011, vol. 6582.

[44] C. FERREIRA, I. LANESE, A. RAVARA, H. T. VIEIRA, G. ZAVATTARO. *Advanced Mechanisms for Service Combination and Transactions*, in "Rigorous Software Engineering for Service-Oriented Systems - Results of the SENSORIA project on Software Engineering for Service-Oriented Computing", M. WIRSING, M. HOLZL (editors), Lecture Notes in Computer Science, Springer, 2011, vol. 6582, chap. 4.3, p. 302-325, http://www.cs.unibo.it/~lanese/publications/fulltext/bookSens-trans.pdf.gz.

[45] I. LANESE, A. RAVARA, H. T. VIEIRA. *Behavioural Theory for Session-oriented Calculi*, in "Rigorous Software Engineering for Service-Oriented Systems - Results of the SENSORIA project on Software Engineering for Service-Oriented Computing", M. WIRSING, M. HOLZL (editors), Lecture Notes in Computer Science, Springer, 2011, vol. 6582, chap. 3.2, p. 189-213, http://www.cs.unibo.it/~lanese/publications/fulltext/bookSens-behav.pdf.gz.

[46] 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, To appear.

[47] D. Sangiorgi. *Introduction to Bisimulation and Coinduction*, Cambridge University Press, 2012, To appear.

[48] D. Sangiorgi. *Origins of Bisimulation and Coinduction*, in "Advanced Topics in Bisimulation and Coinduction", D. Sangiorgi, J. Rutten (editors), Cambridge University Press, 2012, To appear.

### Books or Proceedings Editing

[49] D. Sangiorgi, J. Rutten (editors). *Advanced Topics in Bisimulation and Coinduction*, Cambridge University Press, 2012, To appear.

### Other Publications

[50] M. Carbone, F. Montesi. *Typed Multiparty Global Programming*, 2011, Submitted.

[51] U. D. Lago, P. P. Toldin. *A Higher-Order Characterization of Probabilistic Polynomial Time*, 2011, Submitted.

[52] I. Lanese, F. Montesi, G. Zavattaro. *Classifying Relationships between Interaction- and Process-Oriented Choreographies*, 2011, submitted.

[53] I. Lanese, C. Vaz, C. Ferreira. *On the Expressive Power of Primitives for Compensation Handling*, 2011, submitted, http://www.cs.unibo.it/~lanese/publications/fulltext/ic2011-comp.pdf.gz.

[54] M. Lienhardt, D. Clarke. *Row Types for Delta-Oriented Programming*, 2011, Submitted.

[55] M. D. Preda, M. Gabbrielli, C. Guidi, J. Mauro, F. Montesi. *Theory and practice of interface-based service composition*, 2011, submitted.

## References in notes

[56] M. Carbone, K. Honda, N. Yoshida. *A Calculus of Global Interaction based on Session Types*, in "Electr. Notes Theor. Comput. Sci.", 2007, vol. 171, n$^o$ 3, p. 127–151.

[57] 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.

[58] A. Igarashi, N. Kobayashi. *Resource usage analysis*, in "POPL conference", ACM Press, 2002, p. 331–342.

[59] N. Kobayashi, D. Sangiorgi. *A hybrid type system for lock-freedom of mobile processes*, in "ACM Trans. Program. Lang. Syst.", 2010, vol. 32, n$^o$ 5.

[60] K. Pohl, G. Böckle, F. van der Linden. *Software Product Line Engineering: Foundations, Principles, and Techniques Objects*, Springer Verlag, 2005.