IN PARTNERSHIP WITH:
CNRS

Institut polytechnique de Grenoble

Université Pierre Mendes-France (Grenoble 2)

Université Joseph Fourier (Grenoble 1)

Activity Report 2011

Project-Team SARDES

System architecture for reflective distributed computing environments

IN COLLABORATION WITH: Laboratoire d'Informatique de Grenoble (LIG)

RESEARCH CENTER
Grenoble - Rhône-Alpes

THEME
Distributed Systems and Services

Table of contents

Project-Team SARDES

Keywords: Software Engineering, Autonomic Computing, Component Programming, Discrete Control Systems, Self-Management

1. Members

Research Scientists

Jean-Bernard Stefani [Team Leader, Senior Researcher, Ingénieur Général des Mines]
Eric Rutten [INRIA Junior Researcher, HdR]
Alan Schmitt [INRIA Junior Researcher, until august 2011, HdR]
Damien Pous [CNRS Junior Researcher]
Vivien Quéma [CNRS Junior Researcher, until sept. 2011, HdR]

Faculty Members

Vivien Quéma [Professor, INPG, from sept. 2011, HdR]
Sara Bouchenak [Associate Professor, Université Joseph Fourier, HdR]
Fabienne Boyer [Associate Professor, Université Joseph Fourier, HdR]
Olivier Gruber [Professor, Université Joseph Fourier, HdR]
Noël De Palma [Professor, Université Joseph Fourier, HdR]
Renaud Lachaize [Associate Professor, Université Joseph Fourier]

Technical Staff

Clément Deschamps [INRIA IJD, until nov. 2011]
Fabien Gaud [ARAVIS project, jan.-sept. 2011]
Fabien Mottet [ARAVIS project, until sept. 2011]

PhD Students

Xin An [INRIA, Famous contract]
Pierre-Louis Aublin [government grant]
Gautier Berthou [INRIA, PLAY project]
Loris Bouzonnet [industrial grant, until sept. 2011]
Thomas Braibant [government grant]
Ludovic Demontes [government grant, until sept. 2011]
Xavier Etchevers [Orange Labs]
Sylvain Genevès [government grant, until sept. 2011]
Baptiste Lepers [CNRS, SocEDA project]
Willy Malvaud [Inria grant (contract)]
Claudio Mezzina [INRIA grant (contract]
Alessio Pace [INRIA CORDI grant]
Quentin Sabah [ST Microelectronics]
Amit Sangroya [INRIA, MyCloud contract, from Jan. 2011]

Post-Doctoral Fellows

Tayeb Bouhadiba [MIND project, until may 2011]
Cinzia Di Giusto [Digiteo grant]
Nicolas Palix [ATER university grant, until sept. 2011]
Dàmian Serrano [INRIA grant]

Administrative Assistant

Diane Courtiol

2. Overall Objectives

2.1. Overall objectives

The overall goal of the SARDES project-team is to develop software engineering and software infrastructure (operating system, virtual machine, middleware) foundations for the construction of provably dependable, self-manageable distributed systems.

To contribute to the above goal, the project-team has three major objectives:

1. To develop component-based software technology, that allows the construction of efficient, dynamically configurable systems, and that relies on a well-defined formal foundation.
2. To develop a “language-based” approach to the construction of configurable, provably dependable operating systems and distributed software infrastructures.
3. To develop algorithms and control techniques required to build scalable, self-manageable distributed systems.

In line with these objectives, the project-team organizes its research along four major areas:

- **Languages and foundations for component systems** Work in this area focuses on language support and semantical foundations for distributed component-based systems, with two main goals: (1) the development of a new generation of reflective software component technology with a formal semantical basis, and extensive language support in the form of architecture description and programming languages for dynamic distributed software architectures; (2) the study of process calculus foundations and coinductive proof techniques for distributed component-based programs.
- **System support for multiscale systems** Work in this area focuses on operating system and middleware services required for the construction of component-based systems at different scales (multicore systems on chip, and peer-to-peer systems), with two main goals: (1) to develop algorithms and operating system functions required for the support of efficient event-based concurrency and component reconfiguration in MPSoCs; (2) to develop algorithms and middleware functions required for the deployment, configuration and operation of applications in realistic peer-to-peer environments, typically exploiting an epidemic approach.
- **Control for adaptive and self-managed systems** Workk in this area focuses on the exploitation and development of discrete and continuous control techniques for the construction of adaptive component-based system. Application domains considered for this theme are, respectively, embedded systems and performance management for application server clusters.
- **Virtual machine for component systems** Work in this area focuses on the development of a component-based virtual machine for embedded systems, with two main goals: (1) to develop an extended instruction set for component support, including support for dynamic configuration, orthogonal component persistence, and isolation; (2) to develop a native implementation of the virtual machine, on resource-constrained hardware.

3. Scientific Foundations

3.1. Components and semantics

The primary foundations of the software component technology developed by SARDES relate to the component-based software engineering [110], and software architecture [108] fields. Nowadays, it is generally recognized that component-based software engineering and software architecture approaches are crucial to the development, deployment, management and maintenance of large, dependable software systems [57]. Several component models and associated architecture description languages have been devised over the past fifteen years: see e.g. [89] for an analysis of recent component models, and [93], [64] for surveys of architecture description languages.

To natively support configurability and adaptability in systems, SARDES component technology also draws from ideas in reflective languages [84], and reflective middleware [87], [62], [69]. Reflection can be used both to increase the separation of concerns in a system architecture, as pioneered by aspect-oriented programming [85], and to provide systematic means for modifying a system implementation.

The semantical foundations of component-based and reflective systems are not yet firmly established, however. Despite much work on formal foundations for component-based systems [90], [52], several questions remain open. For instance, notions of program equivalence when dealing with dynamically configurable capabilities, are far from being understood. To study the formal foundations of component-based technology, we try to model relevant constructs and capabilities in a process calculus, that is simple enough to formally analyze and reason about. This approach has been used successfully for the analysis of concurrency with the π-calculus [96], or the analysis of object-orientation [53]. Relevant developments for SARDES endeavours include behavioral theory and coinductive proof techniques [105], [103], process calculi with localities [65], [67], [70], and higher-order variants of the π-calculus [104], [77].

3.2. Open programming

Part of the language developments in SARDES concern the challenge of providing programming support for computer systems with continuously running services and applications, that operate at multiple physical and logical locations, that are constantly introduced, deployed, and combined, that interact, fail and evolve all the time. Programming such systems – called *open programming* by the designers of the Alice programming language [101] — is challenging because it requires the combination of several features, notably: (i) *modularity*, i.e. the ability to build systems by combining and composing multiple elements; (ii) *security*, i.e. the ability to deal with unknown and untrusted system elements, and to enforce if necessary their isolation from the rest of the system; (iii) *distribution*, i.e. the ability to build systems out of multiple elements executing separately on multiple interconnected machines, which operate at different speed and under different capacity constraints, and which may fail independently; (iv) *concurrency*, i.e. the ability to deal with multiple concurrent events, and non-sequential tasks; and (v) *dynamicity*, i.e. the ability to introduce new systems, as well as to remove, update and modify existing ones, possibly during their execution.

The rigorous study of programming features relate to the study of programming language constructs and semantics [97], [112], in general. Each of the features mentioned above has been, and continues to be, the subject of active research on its own. Combining them into a practical programming language with a well-defined formal semantics, however, is still an open question. Recent languages that provide relevant background for SARDES' research are:

- For their support of dynamic notions of modules and software components: Acute [106], Alice [101], [102], ArchJava [54], Classages [91], Erlang [56], Oz [112], and Scala [98].
- For their security and failure management features: Acute, E [95], Erlang and Oz [68].
- For their support for concurrent and distributed execution, Acute, Alice, JoCaml [73], E, Erlang, Klaim [61], and Oz.

3.3. Software infrastructure

The SARDES approach to software infrastructure is both architecture-based and language-based: architecture-based for it relies on an explicit component structure for runtime reconfiguration, and language-based for it relies on a high-level type safe programming language as a basis for operating system and middleware construction. Exploiting high-level programming languages for operating system construction [109] has a long history, with systems such as Oberon [113], SPIN [60] or JX [74]. More recent and relevant developments for SARDES are:

- The developments around the Singularity project at Microsoft Research [72], [80], which illustrates the use of language-based software isolation for building a secure operating system kernel.

- The seL4 project [75], [86], which developed a formal verification of a modern operating system microkernel using the Isabelle/HOL theorem prover.
- The development of operating system kernels for multicore hardware architectures such as Corey [63] and Barrelfish [59].
- The development of efficient run-time for event-based programming on multicore systems such as libasync [114], [88].

3.4. System management and control

Management (or *Administration*) is the function that aims at maintaining a system's ability to provide its specified services, with a prescribed quality of service. We approach management as a *control* activity, involving an event-reaction loop: the management system detects events that may alter the ability of the managed system to perform its function, and reacts to these events by trying to restore this ability. The operations performed under system and application administration include observation and monitoring, configuration and deployment, resource management, performance management, and fault management.

Up to now, administration tasks have mainly been performed in an ad-hoc fashion. A great deal of the knowledge needed for administration tasks is not formalized and is part of the administrators' know-how and experience. As the size and complexity of the systems and applications are increasing, the costs related to administration are taking up a major part of the total information processing budgets, and the difficulty of the administration tasks tends to approach the limits of the administrators' skills. For example, an analysis of the causes of failures of Internet services [99] shows that most of the service's downtime may be attributed to management errors (e.g. wrong configuration), and that software failures come second. In the same vein, unexpected variations of the load are difficult to manage, since they require short reaction times, which human administrators are not able to achieve.

The above motivates a new approach, in which a significant part of management-related functions is performed automatically, with minimal human intervention. This is the goal of the so-called *autonomic computing* movement [82]. Several research projects [51] are active in this area. [83], [79] are recent surveys of the main research problems related to autonomic computing. Of particular importance for SARDES' work are the issues associated with configuration, deployment and reconfiguration [71], and techniques for constructing control algorithms in the decision stage of administration feedback loops, including discrete control techniques [66], and continuous ones [76].

Management and control functions built by SARDES require also the development of distributed algorithms [92], [111] at different scales, from algorithms for multiprocessor architectures [78] to algorithms for cloud computing [94] and for dynamic peer-to-peer computing systems [55], [100]. Of particular relevance in the latter contexts are epidemic protocols such as gossip protocols [107] because of their natural resilience to node dynamicity or *churn*, an inherent scalability.

4. Software

4.1. AAC_tactics

Participants: Thomas Braibant, Damien Pous [correspondant].

AAC_tactics is a plugin for the Coq proof-assistant that implements new proof tactics for rewriting modulo associativity and commutativity. It is available at http://sardes.inrialpes.fr/~braibant/aac_tactics and as part of the Coq distribution.

- ACM: D.2.4 Software/Program Verification
- Keywords: Rewriting, rewriting modulo AC, proof tactics, proof assistant

- Software benefit: AAC_tactics provides novel efficient proof tactics for rewriting modulo associativity and commutativity.
- License: LGPL
- Type of human computer interaction: N/A
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Coq

4.2. ATBR

Participants: Thomas Braibant, Damien Pous [correspondant].

ATBR (Algebraic Tools for Binary Relations) is library for the Coq proof assistant that implements new proof tactics for reasoning with binary relations. Its main tactics implements a decision procedure for inequalities in Kleene algebras. It is available at http://sardes.inrialpes.fr/~braibant/atbr and as part of the Coq distribution contributed modules.

- ACM: D.2.4 Software/Program Verification
- Keywords: Binary relations, Kleene algebras, proof tactics, proof assistant
- Software benefit: ATBR provides new proof tactics for reasoning with binary relations.
- License: LGPL
- Type of human computer interaction: N/A
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Coq

4.3. MoKa

Participant: Sara Bouchenak [correspondant].

MoKa is a software framework for the modeling and capacity planning of distributed systems. It first provides a set of tools to build analytical models that describe the behavior of distributed computing systems, in terms of performance, availability, cost. The framework allows to include several model algorithms and to compare them regarding their accuracy and their efficiency. Furthermore, MoKa provides a set of tools to build capacity planning methods. A capacity planning method allows to find a distributed system configuration that guarantee given quality-of-service objectives. MoKa is able to include different capacity planning algorithms and to compare them regarding their efficiency and the optimality of their results. MoKais available at: http://sardes.inrialpes.fr/research/moka.

- ACM: C.2.4 Distributed Systems, C.4 Performance of Systems, D.2.9 Management
- Keywords: Caching, multi-tier systems, consistency, performance
- Software benefit: a novel end-to-end caching protocol for multi-tier services.
- License: TBD
- Type of human computer interaction: command-line interface
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Java

4.4. ConSer

Participant: Sara Bouchenak [correspondant].

CONSER is a software framework for the modeling and the concurrency and admision control of servers systems. It implements a fluid-based model that exhibits the dynamics and behavior of a server system in terms of service performance and availability. CONSER implements various novel admission control laws for servers such as $AM\text{-}C$, $PM\text{-}C$, $AA\text{-}PM\text{-}C$ and $PA\text{-}AM\text{-}C$. A control law produces the server concurrency level that allows to trade-off and meet given service level objectives. CONSER's modeling and control laws algorithms are implemented following a proxy-based approach for more transparency.

- ACM: C.4 Performance of Systems; D.2.9 Management
- Keywords: System management, capacity planning, performance management
- Software benefit: MoKa provides modeling, capacity planning and performance management facilities for application server clusters. Thanks to its model-based capacity planning, MoKa is able to enforce service level objectives while minimizing the service cost.
- License: LGPL
- Type of human computer interaction: web interface
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Java, AspectJ

4.5. e-Caching

Participants: Dàmian Serrano, Sara Bouchenak [correspondant].

E-CACHING is a software framework for higher scalability of multi-tier Internet services through end-to-end caching of dynamic data. It provides a novel caching solution that allows to cache different types of data (e.g. Web content, database query results, etc.), at different locations of multi-tier Internet services. The framework allows to combine different caches and, thus, to provide higher scalability of Internet services. E-CACHING maintains the integrity of the cached data through novel distributed caching algorithms that guarantee the consistency of the underlying data.

- ACM: C.2.4 Distributed Systems, C.4 Performance of Systems
- Keywords: Caching, multi-tier systems, consistency, performance
- Software benefit: a novel end-to-end caching protocol for multi-tier services, consistency management, performance improvement.
- License: TBD
- Type of human computer interaction: command-line interface
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Java

4.6. MRB

Participants: Amit Sangroya, Dàmian Serrano, Sara Bouchenak [correspondant].

MRB is a software framework for benchmarking the performance and dependability of MapReduce distributed systems. It includes five benchmarks covering several application domains and a wide range of execution scenarios such as data-intensive vs. compute-intensive applications, or batch applications vs. interactive applications. MRB allows to characterize application workload, faultload and dataload, and it produces extensive performance and dependability statistics.

- ACM: C.2.4 Distributed Systems, C.4 Performance of Systems
- Keywords: Benchmark, performance, dependability, MapReduce, Hadoop, Cloud Computing
- Software benefit: the first performance and dependability benchmark suite for MapReduce systems.
- License: TBD
- Type of human computer interaction: GUI and command-line interface
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Java, Unix Shell scripts

4.7. BZR

Participants: Eric Rutten [correspondant], Gwenaël Delaval [POP ART team].

BZR is a reactive language, belonging to the synchronous languages family, whose main feature is to include discrete controller synthesis within its compilation. It is equipped with a behavioral contract mechanisms, where assumptions can be described, as well as an "enforce" property part: the semantics of the latter is that the property should be enforced by controlling the behaviour of the node equipped with the contract. This property will be enforced by an automatically built controller, which will act on free controllable variables given by the programmer.

BZR is now further developed with the Pop-Art team, where G. Delaval got a position. It has been designed and developed in the Sardes team in relation with the research topic on Model-based Control of Adaptive and Reconfigurable Systems. It is currently applied in different directions: component-based design and the Fractal framework; real-time control systems and the Orccad design environment; operating systems and administration loops in virtual machines; hardware and reconfigurable architecture (FPGAs).

See also the web page http://bzr.inria.fr.

- ACM: D.3.3 [Programming Languages]: Language Constructs and Features—Control structures; C.3 [Special-purpose and Application-based Systems]: Real-time and embedded systems; D.2.2 [Software Engineering]: Design Tools and Techniques—Computer-aided software engineering, State diagrams; D.2.4 [Software Engineering]: Software / Program Verification—Formal methods, Programming by contract
- Keywords: Discrete controller synthesis, modularity, components, contracts, reactive systems, synchronous programming, adaptive and reconfigurable systems
- Software benefit: the first integration of discrete control synthesis in a compiler, making it usable at the level of the programming language.
- License: TBD
- Type of human computer interaction: programming language and command-line interface
- OS/Middleware: Linux
- Programming language: Caml; generates C or Java or Caml executable code

5. New Results

5.1. Languages and Foundations: Process algebra

Participants: Damien Pous, Alan Schmitt, Jean-Bernard Stefani, Claudio Mezzina, Cinzia di Giusto.

The goal of this work is to study process algebraic foundations for component-based distributed programming. Most of this work takes place in the context of the ANR PiCoq project.

To develop composable abstractions for programming dependable systems, we investigate concurrent reversible models of computation, where arbitrary executions can be reversed, step by step, in a causally consistent way. This year we have continued the study of the reversible higher-order pi-calculus and obtained a new encoding of it in the higher-order pi-calculus which improves on the result we published in Concur 2010 by proving the faithfulness of the encoding with a much finer equivalence relation. We also developed a reversible variant of the higher-order pi-calculus where reversibility can be controlled by means of an explicit rollback primitive [37]. We have proved that this rollback primitive is sound and complete in that it provides a causally consistent and complete reversal of concurrent computations, and we have developed a low-level semantics for this primitive, closer to an actual distributed implementation, which we have proved equivalent to the high-level one. All these results are presented in detail in Claudio Mezzina's forthcoming PhD thesis, and have been developed in cooperation with the INRIA Focus team at the University of Bologna.

An interesting and expressive component model for embedded systems is the BIP component model [58], developed by J. Sifakis' team at the Verimag Laboratory, which features hierarchical software architectures, explicit constructs for specifying component compositions (*glues*), and multipoint synchronization under priority constraints. We have begun a process calculus analysis of BIP, with a view to combine the reactive features of BIP with the dynamic reconfiguration features of Fractal. Our first result takes the form of new process calculus, called CAB, which we have proved to be a conservative extension of BIP. CAB also enabled us to study the intrinsic expressivity of the BIP model and to prove that priority constraints are essential to BIP expressivity [34].

We have made significant progress on the formalization in the Coq proof assistant of a core higher-order π-calculus, called HOcore [20]. We have in particular adapted a canonical locally nameless representation of binding to handle alpha-conversion in our formalization. Several major theorems of HOcore, in particular the fact that IO-bisimulation is correct in relation to barbed congruence and is decidable. This work has been submitted for publication.

A longer version of our work on untyping theorems and cyclic linear logic has been accepted for publication in LMCS [24], and a book chapter on up-to techniques for bisimulations, written with Davide Sangiorgi from the INRIA Focus team in Bologna, has been published by Cambdridge University Press [46].

Together with Filippo Bonchi (LIP, ENS Lyon), we have worked on a new algorithm for checking the language equivalence of non-deterministic finite automata (NFA). This algorithm improves on the standard Hopcroft and Karp' algorithm, by using up-to techniques. The first empirical tests look really promising [47]

Together with Tom Hirschowitz (LAMA, U. de Chambéry), we have worked on a categorical model of CCS, where innocent strategies are pre-sheaves. This work has been presented at the ICE workshop [36], and a long version has been submitted to SACS.

5.2. Languages and Foundations: Proof tactics

Participants: Damien Pous, Thomas Braibant.

The goal of this work is to develop proof-assistant-based tools for verifying distributed systems and distributed abstract machines. In particular, we aim to support the derivation of fully formal proofs of correctness for abstract machines supporting the component-based languages and programming models we develop.

We have presented our work about tools for rewriting modulo AC in Coq at CPP'11 [32]. An extended version of our work on Kleene algebra (ATBR, first published at ITP'10), was accepted for publication in LMCS [19]. Also on the Coq side, we have developed a library for verifying hardware circuits, which was also presented at CPP'11 [31].

5.3. Control for adaptive systems: Discrete control for adaptive and reconfigurable systems

Participants: Eric Rutten, Noël de Palma, Olivier Gruber, Fabienne Boyer, Tayeb Bouhadiba, Xin An.

The goal of this work is to apply control techniques based on the behavioral model of reactive automata and the algorithmic techniques of discrete controller synthesis. We adopt the synchronous approach to reactive systems, and use an associated effective controller synthesis tool, Sigali, developed at INRIA Rennes. Both are integrated into a programming language, called BZR, and its compiler, as an extension of the Heptagon language. We thus have a complete tool-supported method from control modeling down to concrete execution, considering different execution models, and targetting either software or hardware. We explore control theory for computer science, as an original alernative to computer science for control (as more usually in embedded systems), and to classical discrete control systems (as more usually applied to manufacturing). We are exploring several target application domains, where we expect to find commonalities in the control problems, and variations in the definitions of configurations, and in the criteria motivating adaptation.

We have obtained this year the following results:

- At the programming language level, we are continuing the development of a modelling and controller generation language called BZR, which involves DCS in its compilation. BZR is designed and developed in cooperation with the Pop Art and VerTecs (INRIA Rennes) teams [40].
- We have developed a technique for designing reconfiguration controllers in the Fractal component-based framework, where discrete control loops automatically enforce safety properties on the interactions between components, concerning, e.g., mutual exclusions, forbidden or imposed sequences [29] [48] .
- We have integrated BZR with Orccad, a programming environment for real-time control systems, in cooperation with the NeCS and SED teams at INRIA Grenoble [28].
- We are investigating administration loops in virtual machine-based distributed systems [44], and the coordination of such loops, especially in relation with green computing problems. We are starting a new ANR project called *Ctrl-Green* on this topic in 2012.
- We work on the formal modelling and control of dynamic reconfiguration in FPGA circuits, in cooperation with the DaRT team (INRIA Lille) [43] and the Lab-STICC laboratory in Lorient [42], building upon earlier work related to the MARTE framework.
- In cooperation with GIPSA-Lab and ENSI Tunis, we have adapted the discrete controller synthesis technique to the control of decentralized systems that are composed of several subsystems spread across remote sites [17].
- In cooperation with Orange labs and GIPSA-Lab we are beginning to explore the application of discrete event systems and supervisory control to the domain of Machine to Machine and Internet of Things, with the objective to manage energy aspects; this will start with the CIFRE PhD (U. Grenoble) of Mengxuan Zhao (co-advised with H. Alla, G. Privat).

5.4. System configuration and deployment

Participants: Loris Bouzonnet, Fabienne Boyer, Willy Malvault, Noël de Palma, Vivien Quéma, Jean-Bernard Stefani.

The goal of this work is to study system configuration and software deployment issues in large distributed systems.

System configuration and software deployment in a distributed environment can be greatly aided by the use of a uniform component model to support software assembly, software configuration and deployment, as well as runtime system configuration. We have developed a specialization of the Fractal component model that provides a reference model for heterogeneous software assembly and configuration. In particular, we have shown how this reference model can be used to assemble and configure software architectures built from heterogeneous software packages (e.g. OSGI bundles for Java packages, Debian or RPM packages for Linux modules and applications). The definition of this model, a description of its implementation and its evaluation are documented in Loris Bouzonnet's PhD thesis [13].

As an alternative to current public cloud infrastructures, which rely on large data centers, we have started the study of a cloud infrastructure based on a peer-to-peer (P2P) overlay network built on gossip-based protocols. More precisely, we have studied how to implement a distributed resource allocation service in a P2P environment maintained by a gossip-based peer-sampling protocol [81]. The resulting system, called Salute, provides for the allocation of application-specific overlays out of an underlying P2P network. By combining several P2P services (including peer-sampling, topology maintenance, and node synchronization), and by partitioning available nodes into free nodes (available for the allocation of new application overlays) and reserve nodes (nodes dedicated to the maintenance of allocated overlays), Salute provides a churn-resilient, completely decentralized cloud infrastructure. In addition, we have shown that Salute can provide its allocation service while maintaining fairness and avoiding starvation. The Salute architecture has been validated through simulations using network traces from different real-worl P2P environments. The Salute architecture, algorithms and their validation are documented in Willy Malvault's PhD thesis [14].

In a cloud computing context the complexity of deploying and configuring non-trivial software architectures is exacerbated. In line with our previous work on architecture-based distributed system management, we have proposed a novel algorithm for configuring component-based distributed applications deployed within several virtual machines in an IaaS environment. The algorithm is completely decentralized, relies on a message queuing middleware and exploits the software architecture descriptions of the applications to deploy and configure, written in an extension of the Fractal Architecture Description Language. A first version of this algorithm, that does not take into account potential failures during the configuration process, has been formally specified in collaboration with Gwen Salaün from the INRIA Vasy team in Grenoble, and presented at IEEE Cloud 2011 [35].

5.5. System support: System support for multicore machines

Participants: Vivien Quéma, Renaud Lachaize, Fabien Gaud, Baptiste Lepers, Sylvain Genevès, Fabien Mottet.

Multicore machines with Non-Uniform Memory Accesses (NUMA) are becoming commodity platforms. Efficiently exploiting their resources remains an open research problem. Most of the body of existing work focuses on increasing locality between computations and memory or I/O resources. This is achieved by allocating data items preferably in local memory nodes, by moving computations close to I/O devices or by moving already allocated memory pages close to the applications which use them most. In all these works, researchers always assume that all processors have equal memory performance. Nevertheless, this assumption is not always valid. In 2011, we have studied the performance achieved by a 16-core NUMA exhibiting an irregular connectivity between processors. Some processors are directly connected to all other processors and access memory nodes with a low latency. Other processors have a lower degree of connectivity and need more hops to access some memory nodes and access memory with a higher latency.

Current operating systems are not aware of such performance characteristics. We have shown that the completion time of applications taken from the PARSEC benchmark suite can vary by up to 15% depending on the processor they are scheduled on. We have thus proposed a new OS scheduler that takes this asymmetry into account in order to make efficient decisions. This scheduler relies on a new metric, called MAPI (number of main Memory Accesses Per retired Instruction), to predict the impact of processor interconnect asymmetry on the performance of applications. We have empirically evaluated the relevance of this metric on applications taken from the PARSEC benchmark suite. We have shown that this metric helps estimating the performance gap between running an application on a "well-interconnected” processor and on a "weakly-interconnected” one. Using this metric, the scheduler we proposed makes efficient decisions. More precisely, we have observed that the scheduler always performed within 3% of the best possible scheduling decision. This work is currently under submission.

5.6. System support: Protocols for resilient systems

Participants: Vivien Quéma, Alessio Pace.

We have worked on replication protocols for P2P systems. In particular, we have worked on replication in so called Distributed Hash-Tables (DHTs). DHTs provide a simple high-level put/get abstraction that can be used to build efficient distributed storage systems. DHTs gained wide popularity in the last decade, fostering a large amount of interest in the academia, and inspiring the design of key/value distributed storage systems deployed in production.

DHTs provide a way to deterministically map objects to nodes and allow efficiently retrieving objects in a distributed fashion. Nodes and objects are logically arranged in a large numeric key-space, according to a given variant of consistent hashing. Typically, the node in charge of an object is the one whose position immediately follows the object in the key-space.

To guarantee that objects are reliably stored, DHTs rely on replication. A replication protocol is in charge of ensuring that, at any time, each object is replicated on a sufficiently large number of replicas. Several replication strategies have been proposed in the last years. The most efficient ones use predictions about the availability of nodes to reduce the number of object migrations that need to be performed: objects are preferably stored on highly available nodes.

We have proposed an alternative replication strategy. Rather than exploiting highly available nodes, we have designed a protocol that leverages nodes that exhibit regularity in their connection pattern. Roughly speaking, the strategy consists in replicating each object on a set of nodes that is built in such a way that, with high probability, at any time, there are always at least k nodes in the set that are available. We have evaluated this new replication strategy using traces of two real-world systems: eDonkey and Skype. Our evaluation showed that our regularity-based replication strategy induces a systematically lower network usage than existing state of the art replication strategies. This work has been published at the International Symposium on Reliable Distributed Systems, in October 2011.

5.7. System support: End-to-end caching

Participants: Sara Bouchenak, Dàmian Serrano.

Cloud Computing is a paradigm for enabling remote, on-demand access to a virtually infinite set of configurable computing resources. This model aims to provide hardware and software services to customers, while minimizing human efforts in terms of service installation, configuration and maintenance, for both cloud provider and cloud customer. A cloud may have the form of an Infrastructure as a Service (SaaS), a Platform as a Service (PaaS) or a Software as a Service (SaaS). Clouds pose significant challenges to the full elasticity of clouds, their scalibility and their dependability in large scale data management and large scale computing resources. Caching is a means for high performance and scalability of distributed systems. Although caching solutions have been successfully studied for individual systems such as database systems or web servers, if collectively applied, these solutions violate the coherence of cached data. We precisely studied this issue in e-Caching, a novel end-to-end caching system.

The contribution of this work is twofold: guaranteeing the coherence of cached data at multiple locations of a distributed system, while improving the overall performance of the system. In collaboration with Marta Patino and Ricardo Jimenez from Universidad Politecnica de Madrid, we proposed a novel distributed caching protocol, implemented it and evaluated it with real online services. The experiments showed that e-Caching was successfully able to improve service performance by two orders of magnitude.

This work has been presented at CFSE, the French Chapter of ACM-SIGOPS in May 2011. An extended version has been submitted for publication in a journal.

5.8. System support: Performance and dependability benchmaking

Participants: Amit Sangroya, Dàmian Serrano, Sara Bouchenak [correspondant].

MapReduce has become a popular programming model and runtime environment for developing and executing distributed data-intensive and compute-intensive applications. It offers developers a means to transparently handle data partitioning, replication, task scheduling and fault tolerance on a cluster of commodity computers. MapReduce allows a wide range of applications such as log analysis, data mining, Web search engines, scientific computing, bioinformatics, decision support and business intelligence.

There has been a large amount of work on MapReduce towards improving its performance and reliability. Several efforts have explored task scheduling policies in MapReduce, cost-based optimization techniques, replication and partitioning policies. There has also been a considerable interest in extending MapReduce with other fault tolerance models, or with techniques from database systems. However, there has been very little in the way of empiric evaluation for the comparison of the different systems. Most evaluations of these systems have relied on microbenchmarks based on simple MapReduce programs. While microbenchmarks may be useful in targetting specific system features, they are not representative of full distributed applications, and they do not provide multi-user realistic workloads. Furthermore, as far as we know, no studies have investigated dependability benchmarking of MapReduce.

Thus, we provide *MapReduce Benchmarking (MRB)*, a novel MapReduce benchmark suite to enable a thorough analysis of a wide range of features of MapReduce systems. MRB has the following features. First, it enables the *empirical evaluation of the performance and dependability of MapReduce systems*. This provides a means to analyze the effectiveness of scalability and fault tolerance, two key features of MapReduce. Second, it covers a *variety of application domains, workload and faultload characteristics*, ranging from compute-oriented to data-oriented applications, batch applications to online real-time applications. While MapReduce farmewsorks were originally limited to offline batch processing, recent works are exploring the extension of MapReduce beyond batch processing. Moreover, in order to stress MapReduce dependability and performance, the benchmark suite enables different fault injection rates, workloads and concurrency levels. Finally, the benchmark suite is *portable and easy to use on a wide range of platforms*, covering different MapReduce frameworks and cloud infrastructures. This work has been submitted for publication.

5.9. System support: Self-adaptive Internet services

Participant: Sara Bouchenak.

Although distributed services provide a means for supporting scalable Internet applications, their ad-hoc provisioning and configuration pose a difficult tradeoff between service performance and availability. This is made harder as Internet service workloads tend to be heterogeneous, and vary over time in amount of concurrent clients and in mixture of client interactions. This work proposes an approach for building self-adaptive Internet services through utility-aware capacity planning and provisioning. First, an analytic model is presented to predict Internet service performance, availability and cost. Second, a utility function is defined and a utility-aware capacity planning method is proposed to calculate the optimal service configuration which guarantees SLA performance and availability objectives while minimizing functioning costs. Third, an adaptive control method is proposed to automatically apply the optimal configuration to the Internet service. Finally, the proposed model, capacity planning and control methods are implemented and applied to an online bookstore. Experimental evaluations show that the service successfully self-adapts to both workload mix and workload amount variations, and present significant benefits in terms of performance and availability, with a saving of resources underlying the Internet service.

This work is part of the MyCloud ANR project. It has been described in a chapter of the book titled *Performance and Dependability in Service Computing*, 2011. There has been an industrial transfer of the MoKa software prototype.

5.10. Self-Configuration of distributed system in the Cloud

Participants: Fabienne Boyer, Noël de Palma.

Cloud computing environments fall under three main kinds of offers according to the resources they provide. The Infrastructure as a Service (IaaS) level enables the access to virtualized hardware resources (processing, storage and network). The Software as a Service (SaaS) layer aims at providing the end-users with software applications. The intermediary layer, called Platform as a Service (PaaS), offers a set of tools and runtime environments that allow managing the applications life-cycle. This life-cycle includes the phases related to the design, the development, the deployment of applications, and generally speaking all their management stages (workload, fault tolerance, security). This article focuses on the deployment of distributed applications in virtualized environments such as cloud computing. Such deployments require to generate the virtual images that will be instantiated as virtual machines, thus ensuring the execution of the application on an IaaS platform. Each image embeds technical elements (operating system, middleware pieces) and functional ones (data and applicative software entities). Once it has been instantiated, each virtual machine is subjected to a stage of dynamic settings, which finalizes the global configuration of the distributed application.

On the whole, the deployment solutions currently available do not take into account these different configuration parameters, which are mostly managed by dedicated scripts. Moreover these solutions are not able to automate the images generation, their instantiation as virtual machines and their configuration independently from the kind of distributed application to be deployed. For instance, Google App Engine solution only deals with Web services organized into precisely defined tiers. In our opinion, the absence of general solutions results essentially from a lack of formalism for describing the distributed application architecture with its configuration constraints in a virtualized infrastructure such as cloud computing. Pour work focused on a general solution, for Virtual Applications Management Platform, that automates the deployment of any distributed applications in the cloud. The suggested approach is architectural, meaning that it is based upon an explicit representation of the applications' distributed architecture. We offer, on the one hand, a formalism for describing an application as a set of interconnected virtual machines and, on the other hand, an engine for interpreting this formalism and automating the application deployment on an IaaS platform. Specifically, we study three contributions:

- A formalism that offers a global view of the application to be deployed in terms of components with the associated configuration- and interconnection constraints and with their distribution within virtual machines. This formalism extends OVF language, dedicated to virtual machines description, with an architecture description language (ADL) that allows describing a distributed application software architecture;
- A deployment engine, i.e. a runtime support able to deploy automatically an application described with this formalism. This engine is based on a decentralized protocol for self-configuring the instantiated virtual machines. In our opinion it can ease the scalability of the dynamic configuration stage;
- A performance evaluation of the proposed solution on an industrial IaaS platform.

We published in this context two journal articles (TPDS [26] and TAAS [18]) and three conference papers (Cloud11 [35], UCC11 [39] and SAC12).

5.11. Virtual Machine

Participants: Olivier Gruber, Fabienne Boyer, Damien Pous, Ludovic Demontes, Clément Deschamps.

A core aspect of the Synergy virtual machine is its ability to reconfigure component-based applications at execution time. We have focused on the reconfiguration protocol with the intent of verifying and proving its robustness.

In a first step, we have formalized and verified that any correct and complex reconfiguration through our reconfiguration protocol can be processed as a sequence of elementary reconfiguration operations and always results in a component assembly that is architecturally consistent. This aspect has been verified using model-checking techniques. This work has been done in collaboration with Gwen Salaün from the VASY team (Inria Rhône-Alpes). It lead to a publication in the Formal Method (FM'11) conference [30].

In a second step, we have considered software failures that may occur during a reconfiguration. Although the protocol is trusted code, it invokes components to reconfigure them, thereby executing unsafe code that may fail. This work with Damien Pous produced a high-level formalisation of our reconfiguration protocol and a completely certified modelisation of these algorithms in Coq [50]. This work resulted in a submitted publication.

Finally, we have also investigated the control of complex reconfiguration through using discrete synchronous control techniques with Eric Rutten and Gwenael Delaval [44].

6. Contracts and Grants with Industry

6.1. Contracts with Industry

- PhD grant Loris Bouzonnet, funded by Bull.
- PhD grant Quentin Sabah, funded by STMicroelectronics.
- PhD grant Xavier Etchevers, funded by Orange Labs.

7. Partnerships and Cooperations

7.1. Regional Initiatives

7.1.1. Aravis (ANR-Minalogic)

Participants: Vivien Quéma, Renaud Lachaize, Fabien Gaud, Sylvain Genevès, Fabien Mottet, Baptiste Lepers.

The ARAVIS project aims at addressing the challenges raised, both at the hardware and software levels, by the production of highly integrated multiprocessor systems on chip (MPSoCs) designed for demanding applications such as video encoding/decoding and software-defined radio communications. Due to the complexity of the manufacturing process, the latest generations of chips exhibit peculiar features that must be taken into account : (i) massively parrallel processing units, (ii) irregular behavior and aging of the processing units due to unavoidable defects of the manufacturing process. The ARAVIS project strives to provide a hardware and software platform suited to the adaptation requirements raised by the needs of such emerging hardware technologies and applications. The proposed approach encompasses three contributions: (i) a symmetric hardware architecture based on an asynchronous interconnect with integrated voltage/frequency scaling, (ii) a set of regulation algorithms based on control theory to optimize quality of service and energy consumption, (iii) a component-based runtime environment and related software tools to ease the dynamic management of applications and execution resources.

The project partners are STMicroelectronics, CEA-LETI, TIMA and INRIA (Necs and Sardes project teams). The project runs from October 2007 to September 2011.

7.1.2. MIND (ANR-Minalogic)

Participants: Eric Rutten, Jean-Bernard Stefani, Tayeb Bouhadiba, Cinzia di Giusto.

The MIND project aims to develop an industrial technology for component-based construction of embedded systems, based on the Fractal component model.

This includes the development of programming languages (extended C, ADL, IDL), a chain for compiling software architecture descriptions and generating code, and a graphical IDE integrated to Eclipse. In addition, the project aims to study extensions and refinements to the Fractal model suitable for dealing with non-functional aspects such as real-time and priority constraints, the model-based control of dynamic reconfiguration of components, and its integration with the BIP component model developed at the Verimag laboratory.

The project partners include STMicroelectronics, CEA, INRIA (Adam and Sardes prokect teams), Schneider. The project runs from October 2008 to may 2011.

7.2. National Initiatives

7.2.1. ASR Network

The Sardes team is a member of the CNRS research network GDR 725 ASR "Architecture, Système et Réseau". See http://asr.univ-perp.fr/.

7.2.2. Automatique pour l'informatique autonomique (CNRS PEPS)

Participant: Eric Rutten.

This project is lead by Eric Rutten and funded by CNRS in the *programme Projet Exploratoire-Premier(s) Soutien(s) PEPS Rupture de l'INS2I 2011*. It concerns Control Techniques for Autonomic Computing, and intends to group researchers of different backgrounds (Architectures and FPGA, distributed systems and adaptative software, programming languages for reconfiguration, and control theory) to gather experiences and points of view on this multi-disciplinary topic.

http://sardes.inrialpes.fr/~rutten/peps-api/

7.2.3. Cogip (CNRS PEPS)

Participant: Damien Pous.

This project is lead by Filippo Bonchi (LIP, Lyon), and it includes two researchers from Paris: Samuel Mimram (CEA), and Paul-André Melliès (PPS). This project focuses on semantics of concurrent programming languages, by working at the interface between coalgebraic methods and game semantics.

http://perso.ens-lyon.fr/daniel.hirschkoff/cogip/

7.2.4. SocEDA (ANR Arpege project)

Participants: Vivien Quéma, Baptiste Lepers.

The goal of SocEDA is to develop and validate an elastic and reliable federated SOA architecture for dynamic and complex event-driven interaction in large highly distributed and heterogeneous service systems. Such architecture will enable exchange of contextual information between heterogeneous services, providing the possibilities to optimize/personalize the execution of them, according to social network information.

The main outcome of the SocEDA project will be a platform for event-driven interaction between services, that scales at the Internet level based on the proposed architecture and that addresses Quality of Service (QoS) requirements.

The project partners are INRIA (ADAM in Lilles), EBM WebSourcing (FR), ActiveEon (FR), ARMINES (FR), France Telecom R&D (FR), CNRS (I3S and LIG), INSA Lyon, Thales Communications.

The project runs from October 2010 to September 2013.

7.2.5. PiCoq (ANR project)

Participants: Damien Pous, Alan Schmitt, Jean-Bernard Stefani, Thomas Braibant.

The goal of the PiCoq project is to develop an environment for the formal verification of properties of distributed, component-based programs. The project's approach approach lies at the interface between two research areas: concurrency theory and proof assistants. Achieving this goal relies on three scientific advances, which the project intends to address:

- Finding mathematical frameworks that ease modular reasoning about concurrent and distributed systems: due to their large size and complex interactions, distributed systems cannot be analysed in a global way. They have to be decomposed into modular components, whose individual behaviour can be understood.
- Improving existing proof techniques for distributed/modular systems: while behavioural theories of first-order concurrent languages are well understood, this is not the case for higher-order ones. We also need to generalise well-known modular techniques that have been developed for first-order languages to facilitate formalisation in a proof assistant, where source code redundancies should be avoided.
- Defining core calculi that both reflect concrete practice in distributed component programming and enjoy nice properties w.r.t. behavioural equivalences.

The project partners include INRIA (Sardes), LIP (Plume team), and Université de Savoie. the project runs from November 2010 to October 2014.

The ANR PiCoq is in the programme ANR 2010 BLAN 0305 01: http://sardes.inrialpes.fr/collaborations/PiCoq/.

7.2.6. Project MyCloud (ANR project)

Participants: Amit Sangroya, Sara Bouchenak, Dàmian Serrano.

The objective of the MyCloud project is to define and implement a novel cloud model: *SLAaaS* (*SLA aw*are *S*ervice). The SLAaaS model enriches the general paradigm of Cloud Computing, and enables systematic and transparent integration of service levels and SLA to the cloud. SLAaaS is orthogonal to IaaS, PaaS and SaaS clouds and may apply to any of them. The MyCloud project takes into account both the cloud provider and cloud customer points of view. From cloud provider's point of view, MyCloud proposes autonomic SLA management to handle performance, availability, energy and cost issues in the cloud. An innovative approach combines control theory techniques with distributed algorithms and language support in order to build autonomic elastic clouds. Novel models, control laws, distributed algorithms and languages will be proposed for automated provisioning, configuration and deployment of cloud services to meet SLA requirements, while tackling scalability and dynamics issues. On the other hand from cloud customer's point of view, the MyCloud project provides SLA governance. It allows cloud customers to be part of the loop and to be automatically notified about the state of the cloud, such as SLA violation and cloud energy consumption. The former provides more transparecy about SLA guaranties, and the latter aims to raise customers' awareness about cloud's energy footprint.

The project partners are INRIA (Sardes is the project coordinator), Grenoble; LIP6, Paris; EMN, Nantes; We Are Cloud, Montpellier; Elastic Grid LLC, USA.

The project runs from November 2010 to October 2013.

7.2.7. Famous (ANR project)

Participants: Eric Rutten, Xin An.

The FAMOUS project (FAst Modeling and Design FlOw for Dynamically ReconfigUrable Systems) intends to make reconfigurable hardware systems design easier and faster, by introducing a complete methodology that takes the reconfigurability of the hardware as an essential design concept and proposes the necessary mechanisms to fully exploit those capabilities at runtime. The tool under development in this project is expected to be used by both industrial designers and academic researchers, especially for modern application system specific design such as smart cameras, image and video processing, etc.

The project partners are INRIA (Sardes in Grenoble and DaRT in Lille), Université de Bretagne Sud, Université de Bourgogne, Sodius.

The project runs from December 2009 to November 2013.

7.3. European Initiatives

7.3.1. FP7 ICT Project

7.3.1.1. PLAY

Title: Pushing dynamic and ubiquitous interaction between services Leveraged in the Future Internet by ApplYing complex event processing

Type: COOPERATION (ICT)

Defi: Internet of Services, Software & Virtualisation

Instrument: Specific Targeted Research Project (STREP)

Duration: October 2010 - September 2013

Coordinator: FZI (Germany)

Others partners: INRIA (Oasis in Sophia), FZI (Germany), ICCS (Greece), EBM WebSourcing (FR), ARMINES (FR), France Telecom R&D (FR), CIM Grupa DOO (RS)

See also: http://www.play-project.eu/

Abstract: The goal of PLAY is to develop and validate an elastic and reliable federated SOA architecture for dynamic and complex, event-driven interaction in large highly distributed and heterogeneous service systems. Such architecture will enable exchange of contextual information between heterogeneous services, providing the possibilities to optimize/personalize the execution of them, resulting in the so called situational-driven adaptivity.

The main outcome will be a FOT (federated open trusted) Platform for event-driven interaction between services, that scales at the Internet level based on the proposed architecture and that addresses Quality of Service (QoS) requirements. The platform consists of:

Federated middleware layer: a peer-to-peer overlay network combined with a publish/subscribe mechanism, that has the task to collect events coming from the heterogeneous and distributed services,

Distributed complex event processor: an elastic, distributed computing cloud based engine for complex processing/combining of events coming from different services in order to detect interesting situations a service should react on

Situational-aware business adapter: a recommender engine for proposing adaptation and changes in running business processes and services in a non-predefined (ad-hoc) way, by ensuring the consistency of the whole instance

The system will be tested in two use cases: crisis management and telecom industry, showing the advantages of such an architecture for Future Internet. Indeed, PLAY aims to revolutionize the Future Internet by making it situational-aware, which leads to the event-aware services that are able to proactively adapt themselves to the changes in the environment.

8. Dissemination

8.1. Animation of the scientific community

- J.B. Stefani is a member of the Programme Committee of the ACM SAC 2012 international conference, and a member of the editorial board of the international journal Annals of Telecommunications.
- J.B. Stefani is a member of the Technology Council of STMicroelectronics.
- V. Quḿa was the Organizer of the fourth edition of the Winter School on Hot Topics in Distributed Computing.
- V. Quéma was a PC member for SRDS, AlgoTel, CFSE and CSE.
- A. Schmitt is a member of the steering committee of the Journées Françaises des Langages Applicatifs (JFLA).
- D. Pous is on the program committee of the Journées Françaises des Langages Applicatifs (JFLA).
- N. de Palma organized the first workshop of green computing middleware that was associate to the middleware conference MW 2011.
- N. de Palma was a PC member for the CFSE conference and ICAS conference.
- S. Bouchenak is a member of the program committee of the International Workshop on Cloud Computing Platforms, CloudCP 2011, of INFOCOMP 2011 and of CFSE 2011.
- S. Bouchenak is a co-General Chair of CFSE/RenPar/SympA conferences, 2013.
- S. Bouchenak and E. Rutten co-organize Seminars on the application of control theory and mathematical optimization to distributed system control, LIG laboratory, June and September 2011. http://sardes.inrialpes.fr/%7Ebouchena/seminars/2011-06-07-LIG/index.html

- E. Rutten organized a one-day workshop on *Automatique pour l'informatique autonomique*, satellite of MSR 2011, 15 nov. 2011 http://proton.inrialpes.fr/~rutten/peps-api/peps-api-MSR11.html
- E. Rutten is on the program committees of : APRES 2012: 4th Workshop on Adaptive and Reconfigurable Embedded Systems; ICSSEA'11: 23rd International Conference on Software & Systems Engineering and their Applications; ESLsyn 2011, The 2011 Electronic System Level Synthesis Conference; MSR'11 : Modélisation des Systèmes Réactifs; 3rd Workshop on Adaptive and Reconfigurable Embedded Systems (APRES 2011); Sixth International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks in 2011 (FeBID 2011); 2nd workshop on Distributed Architecture modeling for Novel Component based Embedded systems (DANCE 2011).
- Fabienne Boyer is in the program commitee of the Green Computing Middleware (GCM'2011) workshop and of the Workshop on Middleware and Architectures for Autonomic and Sustainable Computing (MAASC'2011).

8.2. Teaching

V. Quéma is a full-time professor at Grenoble INP, since sept. 2011.

Noël de Palma is a full-time professor at Grenoble University.

Olivier Gruber is a full-time professor at Grenoble University.

Olivier Gruber is the head of the Parallel, Distributed, and Embedded Systems track in the international Master of Science in Informatics at Grenoble (MOSIG).

Renaud Lachaize is a full-time lecturer at Grenoble University.

Sara Bouchenak is a full-time lecturer at Grenoble University.

Fabienne Boyer is a full-time lecturer at Grenoble University.

Fabienne Boyer is the head of the M2PGI (Master Professionnel Génie Informatique) Alternance at Grenoble.

A. Schmitt is teaching a Bachelor level course on Computational Models at U. Joseph Fourier.

D. Pous and A. Schmitt are teaching the Proof Assistant: from Theory to Practice course at the MSTII Doctoral School (U. Grenoble).

D. Pous is teaching a Bachelor level course on Natural Deduction at U. Joseph Fourier.

HdR : Alan Schmitt, Static Analyses for Manipulations of Hierarchically Structured Data, may 2011.

PhD : Alessio Pace, "Gossiping in the wild – Tackling practical problems faced by gossip protocols when deployed on the Internet", Grenoble University, October 2011, V. Quéma, JB Stefani.

PhD : Willy Malvault, "Vers une architecture pair-à-pair pour l'informatique dans le nuage", Grenoble University, October 2011, V. Quéma, JB Stefani.

PhD in progress : Claudio Mezzina, "Reversibility in the higher-order π-calculus", Feb. 2012, J. B. Stefani.

PhD in progress : Thomas Braibant, ”Kleen algebras and circuits in Coq”, Grenoble University, started in sept. 2008, D. Pous, J.B. Stefani.

PhD in progress : Quentin Sabah, "Isolated actors for multi-core programming", Grenoble University, exp. May 2013, J. B. Stefani..

PhD in progress : Sylvain Geneves, "Etude de modèles d'exécutions dans les architectures multi-coeurs", beg Sept. 2008, R. Lachaize, V. Quéma.

PhD in progress : Baptiste Lepers, "MemProf: a memory profiler for multicore systems", beg Jan. 2011, R. Lachaize, V. Quéma.

PhD in progress : Pierre-Louis Aublin, "Robust BFT protocols", beg Sept. 2010, S. Ben Mokhtar, V. Quéma.

PhD in progress : Gautier Berthou, "Accountable Anonymous Communication Protocols", started in Sept 2010, V. Quéma.

PhD in progress : Xin An (U. Grenoble), ”Discrete control and design space exploration for reconfigurable architectures”, started in sept. 2010, adv. E. Rutten and A. Gamatié (INRIA Lille).

PhD in progress : Sébastien Guillet (UBS, Lorient),”Control and decision for dynamically partially reconfigurable FPGA architectures”, started in dec. 2009, adv. F. de la Motte (Lorient) and E. Rutten.

9. Bibliography

Major publications by the team in recent years

[1] É. BRUNETON, T. COUPAYE, M. LECLERCQ, V. QUÉMA, J.-B. STEFANI. *The Fractal Component Model and its Support in Java*, in "Software – Practice and Experience (SP&E)", 2006, vol. 36, n^{o} 11-12.

[2] G. DELAVAL, E. RUTTEN. *Reactive Model-Based Control of Reconfiguration in the Fractal Component-Based Model*, in "13th International Symposium on Component-Based Software Engineering (CBSE)", Lecture Notes in Computer Science, Springer, 2010, vol. 6092.

[3] J. N. FOSTER, M. B. GREENWALD, J. T. MOORE, B. C. PIERCE, A. SCHMITT. *Combinators for bidirectional tree transformations: A linguistic approach to the view-update problem*, in "ACM Trans. Program. Lang. Syst.", 2007, vol. 29, n^{o} 3.

[4] P. GENEVÈS, N. LAYAÏDA, A. SCHMITT. *Efficient Static Analysis of XML Paths and Types*, in "PLDI ’07: Proceedings of the 2007 ACM SIGPLAN conference on Programming language design and implementation", ACM Press, 2007.

[5] R. GUERRAOUI, N. KNEZEVIC, V. QUÉMA, M. VUKOLIC. *The next 700 BFT protocols*, in "5th European Conference on Computer systems (Eurosys)", ACM, 2010.

[6] R. GUERRAOUI, R. R. LEVY, B. POCHON, V. QUÉMA. *Throughput optimal total order broadcast for cluster environments*, in "ACM Trans. Comput. Syst.", 2010, vol. 28, n^{o} 2.

[7] M. LECLERCQ, A. E. ÖZCAN, V. QUÉMA, J.-B. STEFANI. *Supporting Heterogeneous Architecture Descriptions in an Extensible Toolset*, in "29th International Conference on Software Engineering (ICSE)", IEEE Computer Society, 2007.

[8] S. LENGLET, A. SCHMITT, J.-B. STEFANI. *Characterizing Contextual Equivalence in Calculi with Passivation*, in "Information and Computation", November 2011, vol. 209, n^{o} 11, p. 1390–1433.

[9] L. MALRAIT, S. BOUCHENAK, N. MARCHAND. *Fluid modeling and control for server system performance and availability*, in "39th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)", 2009.

[10] J. PHILIPPE, N. DE PALMA, F. BOYER, O. GRUBER. *Self-adaptation of service level in distributed systems*, in "Softw., Pract. Exper.", 2010, vol. 40, n^{o} 3.

[11] D. POUS, D. SANGIORGI. *"Enhancements of the coinductive proof method"*, in "Advanced Topics in Bisimulation and Coinduction", Cambridge University Press, 2011, http://www.cambridge.org/gb/knowledge/isbn/item6542021.

[12] S. SICARD, F. BOYER, N. DE PALMA. *Using Components for Architecture-Based Management: The Self-Repair Case*, in "30th International Conference on Software Engineering (ICSE)", ACM, 2008.

Publications of the year

Doctoral Dissertations and Habilitation Theses

[13] L. BOUZONNET. *Apports des architectures à composants pour le déploiement d'applications à la juste taille*, Université de Grenoble, September 2011, http://hal.inria.fr/tel-00629760/en.

[14] W. MALVAULT. *Vers une architecture pair-à-pair pour l'informatique dans le nuage*, Université de Grenoble, Oct 2011.

[15] A. PACE. *Gossiping in the wild – Tackling practical problems faced by gossip protocols when deployed on the Internet*, Université de Grenoble, Oct 2011.

[16] A. SCHMITT. *Static Analyses for Manipulations of Hierarchically Structured Data*, Université de Grenoble, May 2011, Habilitation à Diriger les Recherches.

Articles in International Peer-Reviewed Journal

[17] M. A. BELHAJ SEBOUI, N. BEN HADJ ALOUANE, G. DELAVAL, E. RUTTEN, M. YEDDES. *An approach for the Synthesis of Decentralized Supervisors for Distributed Adaptive Systems*, in "International Journal of Critical Computer-Based Systems", 2011, vol. 2, n^{o} 3/4, p. 246-265, http://dx.doi.org/10.1504/IJCCBS.2011.042328.

[18] S. BOUCHENAK, F. BOYER, B. CLAUDEL, N. DE PALMA, O. GRUBER, S. SICARD. *From Autonomic to Self-Self Behaviors: The JADE Experience*, in "ACM Transactions on Autonomous and Adaptive Systems", 2011.

[19] T. BRAIBANT, D. POUS. *Deciding Kleene Algebras in Coq*, in "LMCS", 2011, Accepted, to appear, http://fr.arxiv.org/abs/1105.4537v1.

[20] I. LANESE, J. A. PÉREZ, D. SANGIORGI, A. SCHMITT. *On the Expressiveness and Decidability of Higher-Order Process Calculi*, in "Information and Computation", February 2011, vol. 209, n^{o} 2, p. 198–226.

[21] S. LENGLET, A. SCHMITT, J.-B. STEFANI. *Characterizing Contextual Equivalence in Calculi with Passivation*, in "Information and Computation", November 2011, vol. 209, n^{o} 11, p. 1390–1433.

[22] L. MALRAIT, S. BOUCHENAK, N. MARCHAND. *Experience with ConSer: A System for Server Control Through Fluid Modeling*, in "IEEE Transactions on Computers", 2011.

[23] L. MALRAIT, S. BOUCHENAK, N. MARCHAND. *Experience with ConSer: A System for Server Control Through Fluid Modeling*, in "IEEE Transactions on Computers", 2011, vol. 60, n^{o} 7, p. 951-963 [*DOI :* 10.1109/TC.2010.164], http://hal.inria.fr/hal-00480859/en.

[24] D. POUS. *Untyping Typed Algebras and Colouring Cyclic Linear Logic*, in "LMCS", 2011, Accepted, to appear.

[25] L. SEINTURIER, P. MERLE, R. ROUVOY, D. ROMERO, V. SCHIAVONI, J.-B. STEFANI. *A Component-Based Middleware Platform for Reconfigurable Service-Oriented Architectures*, in "Software: Practice and Experience", 2011, http://hal.inria.fr/inria-00567442/en.

[26] N. DE PALMA, D. HAGIMONT, F. BOYER, L. BROTO. *Self-Protection in a Clustered Distributed System*, in "IEEE TPDS", 2011, p. 685-695.

Articles in National Peer-Reviewed Journal

[27] J. ARNAUD, S. BOUCHENAK. *MOKA: Modélisation et planification de capacité pour les systèmes multi-étagés*, in "Revue Technique et Science Informatique (TSI)", 2011.

International Conferences with Proceedings

[28] S. ABOUBEKR, G. DELAVAL, R. PISSARD-GIBOLLET, E. RUTTEN, D. SIMON. *Automatic generation of discrete handlers of real-time continuous control tasks*, in "Proceedings of the 18th World Congress of the International Federation of Automatic Control (IFAC)", Milano, Italy, aug. 28 – sep. 2 2011, p. 786–793, http://dx.doi.org/10.3182/20110828-6-IT-1002.01548.

[29] T. BOUHADIBA, Q. SABAH, G. DELAVAL, E. RUTTEN. *Synchronous Control of Reconfiguration in Fractal Component-based Systems – a Case Study*, in "Proceedings of the International Conference on Embedded Software. EMSOFT 2011, Taipei, Taiwan. October 9-14", 2011, p. 309–318, http://dx.doi.org/10.1145/2038642.2038690.

[30] F. BOYER, O. GRUBER, G. SALAÜN. *Specifying and verifying the SYNERGY reconfiguration protocol with LOTOS NT and CADP*, in "Proceedings of the 17th international conference on Formal methods", Berlin, Heidelberg, FM'11, Springer-Verlag, 2011, p. 103–117, http://dl.acm.org/citation.cfm?id=2021296.2021310.

[31] T. BRAIBANT. *Coquet: A Coq Library for Verifying Hardware*, in "Proc. 1st CPP", LNCS, Springer, 2011, vol. 7086, p. 330-345, http://dx.doi.org/10.1007/978-3-642-25379-9_24.

[32] T. BRAIBANT, D. POUS. *Tactics for Reasoning modulo AC in Coq*, in "Proc. 1st CPP", LNCS, Springer, 2011, vol. 7086, p. 167-182, http://dx.doi.org/10.1007/978-3-642-25379-9_14.

[33] E. BÁRCENAS, P. GENEVÈS, N. LAYAÏDA, A. SCHMITT. *Query Reasoning on Trees with Types, Interleaving, and Counting*, in "Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI '11)", Barcelona, Spain, T. WALSH (editor), AAAI Press/International Joint Conferences on Artificial Intelligence, July 2011, p. 718–723, http://hal.inria.fr/inria-00609556/en.

[34] C. DI GIUSTO, J. STEFANI. *Revisiting glues for component-based systems*, in "13th International Conference on Coordination Models and Languages (COORDINATION)", Lecture Notes in Computer Science, Springer, 2011, vol. 6721.

[35] X. ETCHEVERS, T. COUPAYE, F. BOYER, N. DE PALMA. *Self-Configuration of Distributed Applications in the Cloud*, in "IEEE International Conference on Cloud Computing, CLOUD 2011", IEEE, 2011, p. 668-675.

[36] T. HIRSCHOWITZ, D. POUS. *Innocent strategies as presheaves and interactive equivalences for CCS*, in "Proc. ICE", EPTCS, 2011, vol. 59, http://dx.doi.org/10.4204/EPTCS.59.2.

[37] I. LANESE, C. A. MEZZINA, A. SCHMITT, J.-B. STEFANI. *Controlling Reversibility in Higher-Order Pi*, in "Proceedings of the 22nd International Conference on Concurrency Theory (CONCUR 2011)", Aachen, Germany, Lecture Notes in Computer Science, Springer, September 2011, vol. 6901, p. 297–311.

[38] A. PACE, V. QUÉMA, V. SCHIAVONI. *Exploiting Node Connection Regularity for DHT Replication*, in "30th IEEE Symposium on Reliable Distributed Systems (SRDS)", IEEE, 2011.

[39] G. SALAÜN, X. ETCHEVERS, T. COUPAYE, F. BOYER, N. DE PALMA. *Automated Configuration of Legacy Applications in the Cloud*, in "IEEE UCC", 2011, p. 685-695.

National Conferences with Proceeding

[40] G. DELAVAL, E. RUTTEN, H. MARCHAND. *Intégration de la synthèse de contrôleurs discrets dans un langage de programmation*, in "Modélisation des Systèmes Réactifs (MSR'11)", Lille, France, 2011, http://hal.inria.fr/inria-00629104/en.

[41] F. GAUD, R. LACHAIZE, B. LEPERS, G. MULLER, V. QUÉMA. *Optimisations applicatives pour multi-cœurs NUMA : un cas d'étude avec le serveur web Apache*, in "Conférence Française en Systèmes d'Exploitation (CFSE'8)", May 2011.

[42] S. GUILLET, F. DE LAMOTTE, E. RUTTEN, G. GOGNIAT, J.-P. DIGUET. *Modélisation et contrôle de reconfiguration partielle dynamique*, in "Actes du Symposium en Architecture de Machines, SympA'14", Saint-Malo, France, 10-13 mai 2011 2011.

Conferences without Proceedings

[43] X. AN, A. GAMATIÉ, E. RUTTEN. *Safe Design of Dynamically Reconfigurable Embedded Systems*, in "Proceedings of the 2nd Workshop on Model Based Engineering for Embedded Systems Design (M-BED 2011)", Grenoble, France, March 18 2011.

[44] F. BOYER, G. DELAVAL, N. DE PALMA, O. GRUBER, E. RUTTEN. *Case studies in discrete control of autonomic computing systems*, in "Handouts of the Sixth International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks, FeBID 2011", Karlsruhe, Germany, June 14 2011, http://controlofsystems.org/febid2011/.

Scientific Books (or Scientific Book chapters)

[45] J. ARNAUD, S. BOUCHENAK. *Performance, Availability and Cost of Self-Adaptive Internet Services*, in "Performance and Dependability in Service Computing", IGI Global, 2011.

[46] D. POUS, D. SANGIORGI. *“Enhancements of the coinductive proof method”*, in "Advanced Topics in Bisimulation and Coinduction", Cambridge University Press, 2011, http://www.cambridge.org/gb/knowledge/isbn/item6542021.

Research Reports

[47] F. BONCHI, D. POUS. *Hopcroft and Karp's algorithm for Non-deterministic Finite Automata*, HAL, 2011, http://hal.inria.fr/hal-00639716/.

[48] T. BOUHADIBA, Q. SABAH, G. DELAVAL, E. RUTTEN. *Synchronous Control of Reconfiguration in Fractal Component-based Systems – a Case Study*, INRIA, May 2011, n^{o} RR-7631, http://hal.inria.fr/inria-00596883/en.

[49] F. GAUD, R. LACHAIZE, B. LEPERS, G. MULLER, V. QUÉMA. *Application-Level Optimizations on NUMA Multicore Architectures: the Apache Case Study*, LIG, Grenoble, France, 2011, n^{o} RR-LIG-011, http://rr.liglab.fr/research_report/RR-LIG-011.pdf.

Other Publications

[50] F. BOYER, O. GRUBER, D. POUS. *Robust reconfiguration of component assemblies*, 2011, Coq formalisation available, http://sardes.inrialpes.fr/~pous/rrca/.

References in notes

[51] *Research Projects in Autonomic Computing*, 2003, IBM Research, http://www.research.ibm.com/autonomic/research/projects.html.

[52] Z. LIU, H. JIFENG (editors). *Mathematical Frameworks for Component Software - Models for Analysis and Synthesis*, World Scientific, 2006.

[53] M. ABADI, L. CARDELLI. *A theory of objects*, Springer, 1996.

[54] J. ALDRICH, C. CHAMBERS, D. NOTKIN. *Architectural Reasoning in ArchJava*, in "Proceedings 16th European Conference on Object-Oriented Programming (ECOOP)", 2002.

[55] S. ANDROUTSELLIS-THEOTOKIS, D. SPINELLIS. *A survey of peer-to-peer content distribution technologies*, in "ACM Comput. Surv.", 2004, vol. 36, n^{o} 4.

[56] J. ARMSTRONG, M. WILLIAMS, C. WIKSTROM, R. VIRDING. *Concurrent Programming in Erlang*, Prentice Hall, 1996.

[57] L. BASS, P. CLEMENTS, R. KAZMAN. *Software Architecture in Practice*, SEI Series in Software Engineering, 2nd edition, Addison-Wesley, 2003.

[58] A. BASU, S. BENSALEM, M. BOZGA, J. COMBAZ, M. JABER, T.-H. NGUYEN, J. SIFAKIS. *Rigorous Component-Based System Design Using the BIP Framework*, in "IEEE Software", 2011, vol. 28, n^{o} 3.

[59] A. BAUMANN, P. BARHAM, P. DAGAND, T. L. HARRIS, R. ISAACS, S. PETER, T. ROSCOE, A. SCHÜPBACH, A. SINGHANIA. *The multikernel: a new OS architecture for scalable multicore systems*, in "22nd ACM Symposium on Operating Systems Principles (SOSP)", ACM, 2009.

[60] B. BERSHAD, S. SAVAGE, P. PRZEMYSLAW, E. SIRER, M. FIUCZYNSKI, D. BECKER, C. CHAMBERS, S. EGGERS. *Extensibility, Safety and Performance in the SPIN Operating System*, in "Proceedings 15th ACM Symposium on Operating Systems Principles", 1995.

[61] L. BETTINI, V. BONO, R. DE NICOLA, G. FERRARI, D. GORLA, M. LORETI, E. MOGGI, R. PUGLIESE, E. TUOSTO, B. VENNERI. *The KLAIM Project: Theory and Practice*, in "Global Computing: Programming Environments, Languages, Security and Analysis of Systems", Lecture Notes in Computer Science, Springer-Verlag, 2003, n^o 2874.

[62] G. S. BLAIR, G. COULSON, A. ANDERSEN, L. BLAIR, M. CLARKE, F. COSTA, H. DURAN-LIMON, T. FITZPATRICK, L. JOHNSTON, R. MOREIRA, N. PARLAVANTZAS, K. SAIKOSKI. *The Design and Implementation of OpenORB v2*, in "IEEE Distributed Systems Online, vol. 2 no 6, Special Issue on Reflective Middleware", 2001.

[63] S. BOYD-WICKIZER, H. CHEN, R. CHEN, Y. MAO, M. F. KAASHOEK, R. MORRIS, A. PESTEREV, L. STEIN, M. WU, Y. DAI, Y. ZHANG, Z. ZHANG. *Corey: an operating system for many cores*, in "8th Usenix Symposium on Operating System Design and Implementation (OSDI '08)", 2008.

[64] J. BRADBURY, J. CORDY, J. DINGEL, M. WERMELINGER. *A Survey of Self-Management in Dynamic Software Architecture Specifications*, in "Proceedings of the 1st ACM SIGSOFT Workshop on Self-Managed Systems, WOSS 2004", ACM, 2004.

[65] M. BUGLIESI, G. CASTAGNA, S. CRAFA. *Access control for mobile agents: the calculus of boxed ambients*, in "ACM. Trans. Prog. Languages and Systems, vol. 26, no 1", 2004.

[66] C. G. CASSANDRAS, S. LAFORTUNE. *Introduction to Discrete Event Systems*, 2nd edition, Springer, 2008.

[67] G. CASTAGNA, J. VITEK, F. Z. NARDELLI. *The Seal Calculus*, in "Inf. Comput.", 2005, vol. 201, n^o 1.

[68] R. COLLET. *The Limits of Network Transparency in a Distributed Programming Language*, Université Catholique de Louvain, Belgium, 2007.

[69] G. COULSON, G. S. BLAIR, P. GRACE, F. TAÏANI, A. JOOLIA, K. LEE, J. UEYAMA, T. SIVAHARAN. *A generic component model for building systems software*, in "ACM Trans. Computer Systems", 2008, vol. 26, n^o 1.

[70] S. DAL-ZILIO. *Mobile Processes: A Commented Bibliography*, in "Modeling and Verification of Parallel Processes, 4th Summer School, MOVEP 2000", LNCS, Springer, 2001, vol. 2067.

[71] A. DEARLE. *Software Deployment, Past, Present and Future*, in "Future of Software Engineering (FOSE '07)", IEEE, 2007.

[72] M. FAHNDRICH, M. AIKEN, C. HAWBLITZEL, O. HODSON, G. C. HUNT, J. R. LARUS, S. LEVI. *Language Support for Fast and Reliable Message-based Communication in Singularity OS*, in "1st EuroSys Conference", ACM, 2006.

[73] C. FOURNET, F. LE FESSANT, L. MARANGET, A. SCHMITT. *JoCaml: a language for concurrent, distributed and mobile programming*, in "Summer Schol Adv. Functional Programming", LNCS, 2003, vol. 2638.

[74] M. GOLM, M. FELSER, C. WAWERSICH, J. KLEINÖDER. *The JX Operating System*, in "USENIX Annual Technical Conference, General Track", USENIX, 2002.

[75] G. HEISER, K. ELPHINSTONE, I. KUZ, G. KLEIN, S. PETTERS. *Towards Trusworthy Computing Systems: Taking Microkernels to the Next Level*, in "Operating Systems Review", 2007, vol. 41, n^{o} 4.

[76] J. HELLERSTEIN, Y. DAO, S. PAREKH, D. TILBURY. *Feedback Control of Computing Systems*, Wiley-Interscience, 2004.

[77] M. HENNESSY, J. RATHKE, N. YOSHIDA. *SafeDpi: a language for controlling mobile code*, in "Acta Informatica", 2005, vol. 42, n^{o} 4-5.

[78] M. HERLIHY, N. SHAVIT. *The Art of Multiprocessor Programming*, Morgan Kaufmann, 2008.

[79] M. C. HUEBSCHER, J. A. MCCANN. *A survey of autonomic computing - degrees, models, and applications*, in "ACM Comput. Surv.", 2008, vol. 40, n^{o} 3.

[80] G. C. HUNT, J. R. LARUS. *Singularity: rethinking the software stack*, in "Operating Systems Review", 2007, vol. 41, n^{o} 2.

[81] M. JELASITY, S. VOULGARIS, R. GUERRAOUI, A. KERMARREC, M. VAN STEEN. *Gossip-based peer sampling*, in "ACM Trans. Comput. Syst.", 2007, vol. 25, n^{o} 3.

[82] J. O. KEPHART, D. M. CHESS. *The Vision of Autonomic Computing*, in "Computer - IEEE Computer Magazine", 2003, vol. 36, n^{o} 1, p. 41–50.

[83] J. O. KEPHART. *Research Challenges of Autonomic Computing*, in "ICSE '05: Proceedings of the 27th international conference on Software engineering", New York, NY, USA, ACM Press, 2005, p. 15–22, http://doi.acm.org/10.1145/1062455.1062464.

[84] G. KICZALES, J. DES RIVIÈRES, D. BOBROW. *The Art of the Metaobject Protocol*, MIT Press, 1991.

[85] G. KICZALES. *Aspect-Oriented Programming*, in "ACM Computing Surveys", 1996, vol. 28, n^{o} 4, 154, http://doi.acm.org/10.1145/242224.242420.

[86] G. KLEIN, K. ELPHINSTONE, G. HEISER, J. ANDRONICK, D. COCK, P. DERRIN, D. ELKADUWE, K. ENGELHARDT, R. KOLANSKI, M. NORRISH, T. SEWELL, H. TUCH, S. WINWOOD. *seL4: formal verification of an OS kernel*, in "22nd ACM Symposium on Operating Systems Principles 2009", ACM, 2009.

[87] F. KON, F. COSTA, G. S. BLAIR, R. H. CAMPBELL. *The Case for Reflective Middleware*, in "Communications of the ACM", June 2002, vol. 45, n^{o} 6, p. 33–38.

[88] M. N. KROHN, E. KOHLER, M. F. KAASHOEK. *Events Can Make Sense*, in "USENIX Annual Technical Conference", 2007.

[89] K. LAU, Z. WANG. *Software Component Models*, in "IEEE Trans. Software Eng.", 2007, vol. 33, n^{o} 10.

[90] G. LEAVENS, M. SITARAMAN. *Foundations of Component-Based Systems*, Cambridge University Press, 2000.

[91] Y. LIU, S. SMITH. *Interaction-Based Programming with Classages*, in "Proceedings OOPSLA", 2005.

[92] N. LYNCH. *Distributed Algorithms*, Morgan Kaufmann, 1996.

[93] N. MEDVIDOVIC, R. N. TAYLOR. *A Classification and Comparison Framework for Software Architecture Description Languages*, in "IEEE Transactions on Software Engineering, vol. 26, no. 1", 2000.

[94] P. MELL, T. GRANCE. *The NIST Definition of Cloud Computing*, 2010, http://csrc.nist.gov/groups/SNS/cloud-computing/.

[95] M. MILLER. *Robust Composition: Towards a Unified Approach to Access Control and Concurrency Control*, Johns Hopkins University, baltimor, Maryland, USA, 2006.

[96] R. MILNER. *Communicating and mobile systems : the π-calculus*, Cambridge University Press, 1999.

[97] J. MITCHELL. *Foundations for Programming Languages*, MIT Press, 1996.

[98] M. ODERSKY, M. ZENGER. *Scalable component abstractions*, in "Proceedings OOPSLA", 2005.

[99] D. OPPENHEIMER, A. GANAPATHI, D. A. PATTERSON. *Why do Internet services fail, and what can be done about it?*, in "4th USENIX Symposium on Internet Technologies and Systems (USITS '03)", March 2003.

[100] J. RISSON, T. MOORS. *Survey of research towards robust peer-to-peer networks: Search methods*, in "Computer Networks", 2006, vol. 50, n^o 17.

[101] A. ROSSBERG, D. LE BOTLAN, G. TACK, T. BRUNKLAUS, G. SMOLKA. *Alice Through the Looking Glass*, in "Trends in Functional Programming", Intellect Books, 2006, vol. 5.

[102] A. ROSSBERG. *The Missing Link - Dynamic Components for ML*, in "11th International Conference on Functional Programming", ACM Press, 2006.

[103] D. SANGIORGI. *On the origins of bisimulation and coinduction*, in "ACM Trans. Program. Lang. Syst.", 2009, vol. 31, n^o 4.

[104] D. SANGIORGI. *Bisimulation for Higher-Order Process Calculi*, in "Information and Computation", 1996, vol. 131, n^o 2.

[105] D. SANGIORGI, D. WALKER. *The π-calculus: A Theory of Mobile Processes*, Cambridge University Press, 2001.

[106] P. SEWELL, J. LEIFER, K. WANSBROUGH, F. Z. NARDELLI, M. ALLEN-WILLIANS, P. HABOUZIT, V. VAFEIADIS. *Acute: High-level programming language design for distributed computation*, in "Int. Conf. Functional Programming", 2005.

[107] D. SHAH. *Gossip Algorithms*, in "Foundations and Trends in Networking", 2009, vol. 3, n^o 1.

[108] M. SHAW, D. GARLAN. *Software Architecture: Perspectives on an Emerging Discipline*, Prentice-Hall, 1996.

[109] A. SILBERSCHATZ, P. B. GALVIN, G. GAGNE. *Operating System Concepts*, 8th edition, Wiley, 2008.

[110] C. SZYPERSKI. *Component Software, 2nd edition*, Addison-Wesley, 2002.

[111] G. TEL. *Introduction to Distributed Algorithms*, 2nd edition, Cambdrige University Press, 2001.

[112] P. VAN ROY, S. HARIDI. *Concepts, Techniques and Models of Computer Programming*, MIT Press, 2004.

[113] N. WIRTH. *Project Oberon: The Design of an Operating System and Compiler*, ACM Press, 1992.

[114] N. ZELDOVICH, A. YIP, F. DABEK, R. MORRIS, D. MAZIÈRES, M. F. KAASHOEK. *Multiprocessor Support for Event-Driven Programs*, in "USENIX Annual Technical Conference, General Track", 2003.